Tell us about your PDF experience.

Power BI guidance documentation

Power BI guidance documentation provides best practice information from the team that builds Power BI and the folks that work with our enterprise customers. Here you'll find learnings to improve performance and success with Power BI. We'll update and add to them as new information is available.

Power BI guidance

OVERVIEW

Guidance for Power BI

Optimization guide for Power BI

Whitepapers overview

Transform and shape data

CONCEPT

Query folding guidance

Disable Power Query background refresh

Referencing Power Query queries

Introduction to dataflows

Data modeling

CONCEPT

Understand star schema

Data reduction techniques

One-to-one relationships

Many-to-many relationships

Active vs inactive relationships

Bi-directional relationships

Relationship troubleshooting

DirectQuery model guidance

Composite model guidance

Row-level security (RLS) guidance

DAX

CONCEPT

Appropriate use of error functions

Avoid converting BLANKs to values

Avoid using FILTER as a filter argument

Column and measure references

DIVIDE function vs divide operator

Use COUNTROWS instead of COUNT

Use SELECTEDVALUE instead of VALUES

Use variables to improve formulas

Power BI reports

CONCEPT

Separate reports from models

Extend visuals with report page tooltips

Use report page drillthrough

Power BI paginated reports

CONCEPT

When to use paginated reports

Data retrieval guidance

Image use guidance

Use cascading parameters

Avoid blank pages when printing

Plan to migrate .rdl reports to Power BI

Admin and deployment

CONCEPT

On-premises data gateway sizing

Monitor report performance

Troubleshoot report performance

Lifecycle management best practices

Access the Power BI activity log

Migration to Power BI

CONCEPT

Power BI migration overview

Prepare to migrate to Power BI

Gather requirements

Plan deployment

Conduct proof of concept

Create content

Deploy to Power BI

Learn from customer Power BI migrations

Migrate AAS models to Power BI Premium

Fabric adoption roadmap

CONCEPT

Introduction

Data culture

Executive sponsorship

Content ownership

Content delivery scope

Center of Excellence

Governance

Mentoring and user enablement

Community of practice

User support

Implementation planning

CONCEPT

Introduction

BI strategy

Tenant setup

User tools and devices

Workspaces

Content lifecycle management

Security

Info protection and DLP

Auditing and monitoring

Usage scenarios

Center of Excellence (COE)

CONCEPT

Microsoft's BI transformation

Establish a COE

BI solution architecture in the COE

Fabric adoption roadmap

Guidance for Power BI

Article • 09/22/2021

Here you will find the guidance and recommended practices for Power BI. Guidance will continue to be updated and added to.

Data modeling

Guidance	Description
Understand star schema and the importance for Power BI	Describes star schema design and its relevance to developing Power BI data models optimized for performance and usability.
Data reduction techniques for Import modeling	Describes different techniques to help reduce the data loaded into Import models.

DAX

Guidance	Description
DAX: DIVIDE function vs divide operator (/)	Describes proper use of the DIVIDE function within DAX.

Dataflows

Guidance	Description
Dataflows best practice	Describes best practices for designing dataflows in Power BI.

More questions? Try asking the Power BI Community

Optimization guide for Power BI

Article • 12/30/2024

This article provides guidance that enables developers and administrators to produce and maintain optimized Power BI solutions. You can optimize your solution at different architectural layers. Layers include:

- The data source(s)
- The data model
- Visualizations, including dashboards, Power BI reports, and Power BI paginated reports
- The environment, including capacities, data gateways, and the network

Optimizing the data model

The data model supports the entire visualization experience. Data models are either hosted in the Power BI ecosystem or externally (by using DirectQuery or Live Connection), and in Power BI they are referred to as *semantic models*. It's important to understand your options, and to choose the appropriate semantic model type for your solution. There are three semantic model table storage modes: Import, DirectQuery, and Composite. For more information, see Semantic models in the Power BI service, and Semantic model modes in the Power BI service.

For specific semantic model table storage mode guidance, see:

- Data reduction techniques for Import modeling
- DirectQuery model guidance in Power BI Desktop
- Composite model guidance in Power BI Desktop

Optimizing for report authors and model consumers

The semantic model is the foundation of all reporting in Power BI. Consumers of the semantic model can create Power BI reports in Power BI Desktop by connecting to a published semantic model or connecting to data and creating a local semantic model. The semantic model can also be used to create Power BI reports in the browser, create Power BI explorations, create paginated reports, create DAX queries, and create reports in Excel with Analyze in Excel, connecting to Power BI in Excel, or exporting data from a report visual, as well as many other reporting tools. A semantic model author can help semantic model consumers understand and utilize the semantic model with how they build the model.

- **Names**: Tables, columns, and measures in the semantic model with descriptive names. For example, 'Store Sales' as a table name is more intuitive than 'Table1'.
- **Descriptions**: Tables, columns, and measures in the model can have descriptions added to them to provide more detail than can fit in the name. Explain not only what they include but how they should be used.
- **Hide**: You can hide tables, columns, and measures in the model to show only what you expect them to use in a report. For example, relationship columns may be an ID that is not necessary for reporting and can be hidden as it's not expected to be used in a report, or data columns that have a measure to aggregate the column could be hidden to encourage use of the measure instead. Hidden objects can always be unhidden later by the model consumer, so they will still be available, but hiding can provide focus.
- **Hierarchies**: You can create hierarchies to convey the hierarchy across multiple columns. For example, a Calendar hierarchy may contain Year, Month, Day columns, and a Product hierarchy may contain Category, Sub-Category, Product columns. Right-click a column to create a hierarchy.
- **Measures**: You can use measures to aggregate data columns in the semantic model to provide consistency across reports. Measures can range from the SUM of a column, to a health index combining multiple aggregations in a specific way or comparing aggregations across time periods, such as daily average this month compared to the daily average of the same month last year. Measures can also be surfaced in Power BI search and other features, such as Metrics and Scorecards.
- **Formats**: You can specify how a column or measure is displayed in a visual, by default. Values in visuals can be customized further in the visual. Format options include if it has a thousands comma, how many decimal places, how a date is shown, etc. You can also apply custom or dynamic formats.
- **Data category**: You can specify a column data category, such as if it's a Country or Web URL.

These are common features of Power BI semantic model that can be leveraged to help your report authors and model consumers. There are many others, such as calculation groups, field parameters, what if parameters, and grouping and binning columns, which should be evaluated to see if they apply your specific reporting needs.

Optimizing visualizations

Power BI visualizations can be dashboards, Power BI reports, or Power BI paginated reports. Each has different architectures, and so each has their own guidance.

Dashboards

It's important to understand that Power BI maintains a cache for your dashboard tiles—except live report tiles, and streaming tiles. If your semantic model enforces dynamic row-level security (RLS), be sure to understand performance implications as tiles will cache on a per-user basis.

When you pin live report tiles to a dashboard, they're not served from the query cache. Instead, they behave like reports, and make queries to v-cores on the fly.

As the name suggests, retrieving the data from the cache provides better and more consistent performance than relying on the data source. One way to take advantage of this functionality is to have dashboards be the first landing page for your users. Pin often-used and highly requested visuals to the dashboards. In this way, dashboards become a valuable "first line of defense", which delivers consistent performance with less load on the capacity. Users can still click through to a report to analyze details.

For DirectQuery and live connection semantic models, the cache is updated on a periodic basis by querying the data source. By default, it happens every hour, though you can configure a different frequency in the semantic model settings. Each cache update will send queries to the underlying data source to update the cache. The number of queries that generate depends on the number of visuals pinned to dashboards that rely on the data source. Notice that if row-level security is enabled, queries are generated for each different security context. For example, consider there are two different roles that categorize your users, and they have two different views of the data. During query cache refresh, Power BI generates two sets of queries.

Power BI reports

There are several recommendations for optimizing Power BI report designs.

> **Note**
>
> When reports are based on a DirectQuery semantic model, for additional report design optimizations, see DirectQuery model guidance in Power BI Desktop (Optimize report designs).

Apply the most restrictive filters

The more data that a visual needs to display, the slower that visual is to load. While this principle seems obvious, it's easy to forget. For example: suppose you have a large semantic model. Based on that semantic model, you build a report with a table. End

users use slicers on the page to get to the rows they want—typically, they're only interested in a few dozen rows.

A common mistake is to leave the default view of the table unfiltered—that is, all 100M+ rows. The data for these rows loads into memory and is uncompressed at every refresh. This processing creates huge demands for memory. The solution: use the "Top N" filter to reduce the max number of items that the table displays. You can set the max item to larger than what users would need, for example, 10,000. The result is the end-user experience doesn't change, but memory use drops greatly. And most importantly, performance improves.

A similar design approach to the above is suggested for every visual in your report. Ask yourself, is all the data in this visual needed? Are there ways to filter the amount of data shown in the visual with minimal impact to the end-user experience? Remember, tables in particular can be expensive.

Limit visuals on report pages

The above principle applies equally to the number of visuals added to a report page. It's highly recommended you limit the number of visuals on a particular report page to only what is necessary. Drillthrough pages and report page tooltips are great ways to provide additional details without jamming more visuals onto the page.

Evaluate custom visual performance

Be sure to put each custom visual through its paces to ensure high performance. Poorly optimized Power BI visuals can negatively affect the performance of the entire report.

Power BI paginated reports

Power BI paginated report designs can be optimized by applying best practice design to the report's data retrieval. For more information, see Data retrieval guidance for paginated reports.

Also, ensure your capacity has sufficient memory allocated to the paginated reports workload.

Optimizing the environment

You can optimize the Power BI environment by configuring capacity settings, sizing data gateways, and reducing network latency.

Capacity settings

When using capacities—available with Power BI Premium (P SKUs), Premium Per User (PPU) licenses, or Power BI Embedded (A SKUs, A4-A6)—you can manage capacity settings. For more information, see Microsoft Fabric capacity licenses and Managing Premium capacities.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Gateway sizing

A gateway is required whenever Power BI must access data that isn't accessible directly over the Internet. You can install the on-premises data gateway on a server on-premises, or VM-hosted Infrastructure-as-a-Service (IaaS).

To understand gateway workloads and sizing recommendations, see On-premises data gateway sizing.

Network latency

Network latency can impact report performance by increasing the time required for requests to reach the Power BI service, and for responses to be delivered. Tenants in Power BI are assigned to a specific region.

> **Tip**
>
> To determine where your tenant is located, see **Where is my Power BI tenant located?**

When users from a tenant access the Power BI service, their requests always route to this region. As requests reach the Power BI service, the service may then send additional

requests—for example, to the underlying data source, or a data gateway—which are also subject to network latency.

Tools such as Azure Speed Test provide an indication of network latency between the client and the Azure region. In general, to minimize the impact of network latency, strive to keep data sources, gateways, and your Power BI capacity as close as possible. Preferably, they reside within the same region. If network latency is an issue, try locating gateways and data sources closer to your Power BI capacity by placing them inside cloud-hosted virtual machines.

Monitoring performance

You can monitor performance to identify bottlenecks. Slow queries—or report visuals—should be a focal point of continued optimization. Monitoring can be done at design time in Power BI Desktop, or on production workloads in Power BI Premium capacities. For more information, see Monitoring report performance in Power BI.

Related content

For more information about this article, check out the following resources:

- Power BI guidance
- Monitoring report performance
- Fabric adoption roadmap
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Query folding guidance in Power BI Desktop

Article • 12/30/2024

This article targets data modelers developing models in Power BI Desktop. It provides best practice guidance on when—and how—you can achieve Power Query query folding.

Query folding is the ability for a Power Query query to generate a single query statement that retrieves and transforms source data. For more information, see Power Query query folding.

Guidance

Query folding guidance differs based on the model mode.

For a **DirectQuery** or **Dual** storage mode table, the Power Query query must achieve query folding.

For an **Import** table, it may be possible to achieve query folding. When the query is based on a relational source—and if a single `SELECT` statement can be constructed—you achieve *best data refresh performance* by ensuring that query folding occurs. If the Power Query mashup engine is still required to process transformations, you should strive to minimize the work it needs to do, especially for large semantic models.

The following bulleted-list provides specific guidance.

- **Delegate as much processing to the data source as possible**: When all steps of a Power Query query can't be folded, discover the step that prevents query folding. When possible, move later steps earlier in sequence so they may be factored into the query folding. Note the Power Query mashup engine may be smart enough to reorder your query steps when it generates the source query.

  For a relational data source, if the step that prevents query folding could be achieved in a single `SELECT` statement—or within the procedural logic of a stored procedure—consider using a native SQL query, as described next.

- **Use a native SQL query**: When a Power Query query retrieves data from a relational source, it's possible for some sources to use a native SQL query. The query can in fact be any valid statement, including a stored procedure execution. If the statement produces multiple result sets, only the first will be returned.

Parameters can be declared in the statement, and we recommend that you use the Value.NativeQuery M function. This function was designed to safely and conveniently pass parameter values. It's important to understand that the Power Query mashup engine can't fold later query steps, and so you should include all—or as much—transformation logic in the native query statement.

There are two important considerations you need to bear in mind when using native SQL queries:

- For a DirectQuery model table, the query must be a `SELECT` statement, and it can't use Common Table Expressions (CTEs) or a stored procedure.
- Incremental refresh can't use a native SQL query. So, it would force the Power Query mashup engine to retrieve all source rows, and then apply filters to determine incremental changes.

> **Important**
>
> A native SQL query can potentially do more than retrieve data. Any valid statement can be executed (and possibly multiple times), including one that modifies or deletes data. It's important that you apply the principle of least privilege to ensure that the account used to access the database has only read permission on required data.

- **Prepare and transform data in the source**: When you identify that certain Power Query query steps can't be folded, it may be possible to apply the transformations in the data source. The transformations could be achieved by writing a database view that logically transforms source data. Or, by physically preparing and materializing data, in advance of Power BI querying it. A relational data warehouse is an excellent example of prepared data, usually consisting of pre-integrated sources of organizational data.

Related content

For more information about this article, check out the following resources:

- Power Query Query folding concept article
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Referencing Power Query queries

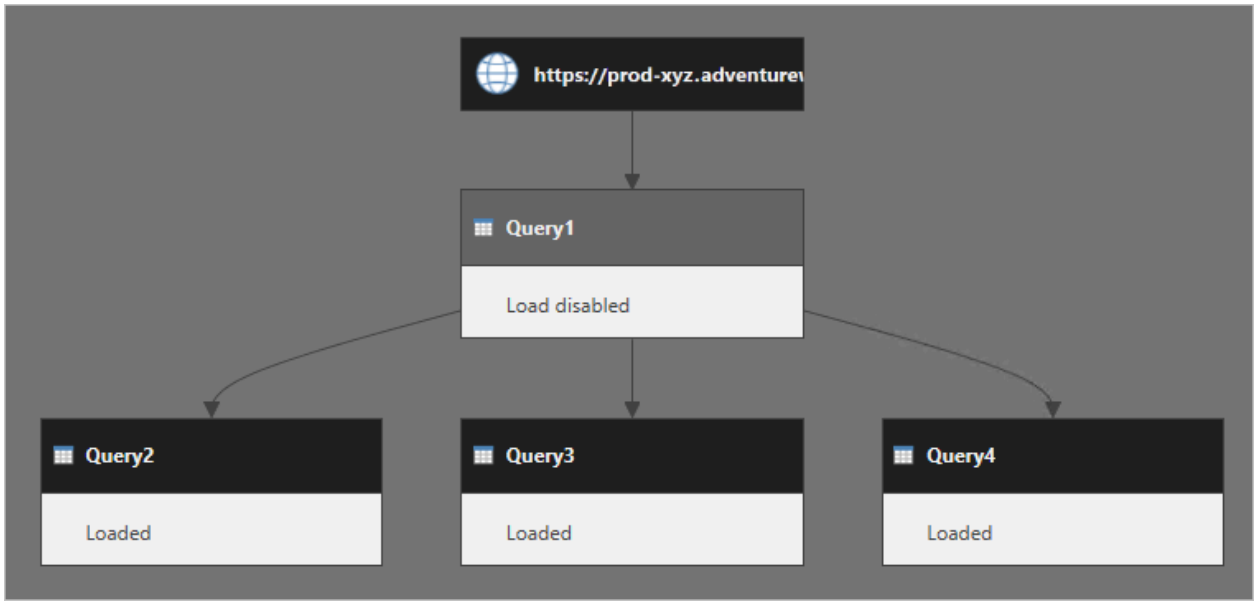
Article • 12/30/2024

This article targets you as a data modeler working with Power BI Desktop. It provides you with guidance when defining Power Query queries that reference other queries.

Let's be clear about what this means: *When a query references a second query, it's as though the steps in the second query are combined with, and run before, the steps in the first query.*

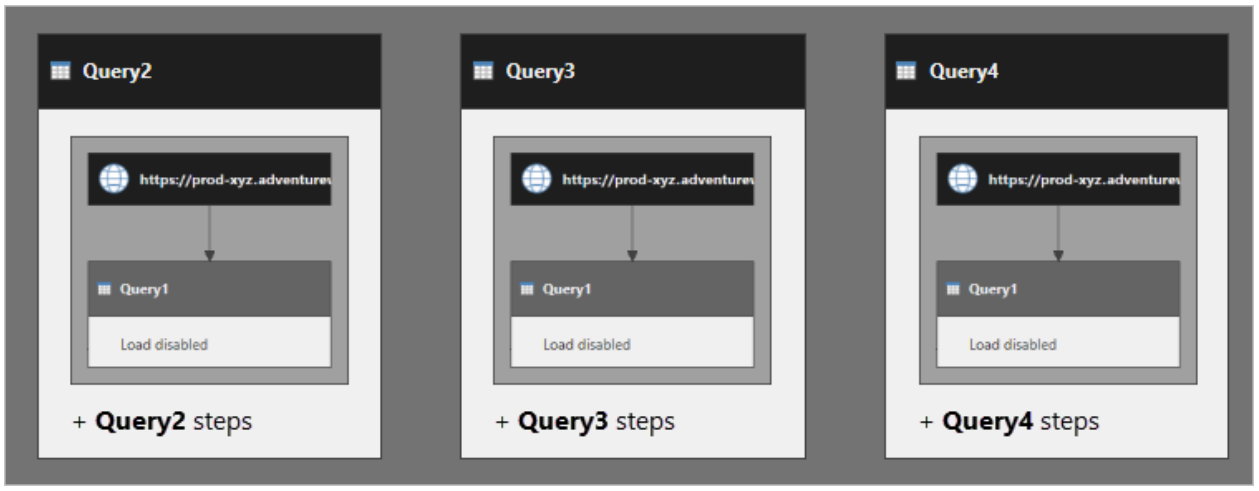
Scenario

Consider several queries: **Query1** sources data from a web service, and its load is disabled. **Query2**, **Query3**, and **Query4** all reference **Query1**, and their outputs are loaded to the data model.



When the data model is refreshed, it's often assumed that Power Query retrieves the **Query1** result, and that it's reused by referenced queries. This thinking is incorrect. In fact, Power Query executes **Query2**, **Query3**, and **Query4** separately.

You can think that **Query2** has the **Query1** steps embedded into it. It's the case for **Query3** and **Query4**, too. The following diagram presents a clearer picture of how the queries are executed.



Query1 is executed three times. The multiple executions can result in slow data refresh and negatively impact the data source.

The use of the Table.Buffer function in **Query1** won't eliminate the additional data retrieval. This function buffers a table to memory, and the buffered table can only be used within the same query execution. So, in the example, if **Query1** is buffered when **Query2** is executed, the buffered data couldn't be used when **Query3** and **Query4** are executed. They themselves will buffer the data twice more. (This result could in fact compound the negative performance, because the table will be buffered by each referencing query.)

> ⓘ **Note**
>
> Power Query caching architecture is complex, and it's not the focus of this article. Power Query can cache data retrieved from a data source. However, when it executes a query, it may retrieve the data from the data source more than once.

Recommendations

Generally, we recommend you reference queries to avoid the duplication of logic across your queries. However, as described in this article, this design approach can contribute to slow data refreshes and overburden data sources.

We recommend you create a dataflow instead. Using a dataflow can improve data refresh time and reduce impact on your data sources.

You can design the dataflow to encapsulate the source data and transformations. As the dataflow is a persisted store of data in the Power BI service, its data retrieval is fast. So, even when referencing queries result in multiple requests for the dataflow, data refresh times can be improved.

In the example, if **Query1** is redesigned as a dataflow entity, **Query2**, **Query3**, and **Query4** can use it as a data source. With this design, the entity sourced by **Query1** will be evaluated only once.

Related content

For more information related to this article, check out the following resources:

- Self-service data prep in Power BI
- Creating and using dataflows in Power BI
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Disable Power Query background refresh

Article • 12/30/2024

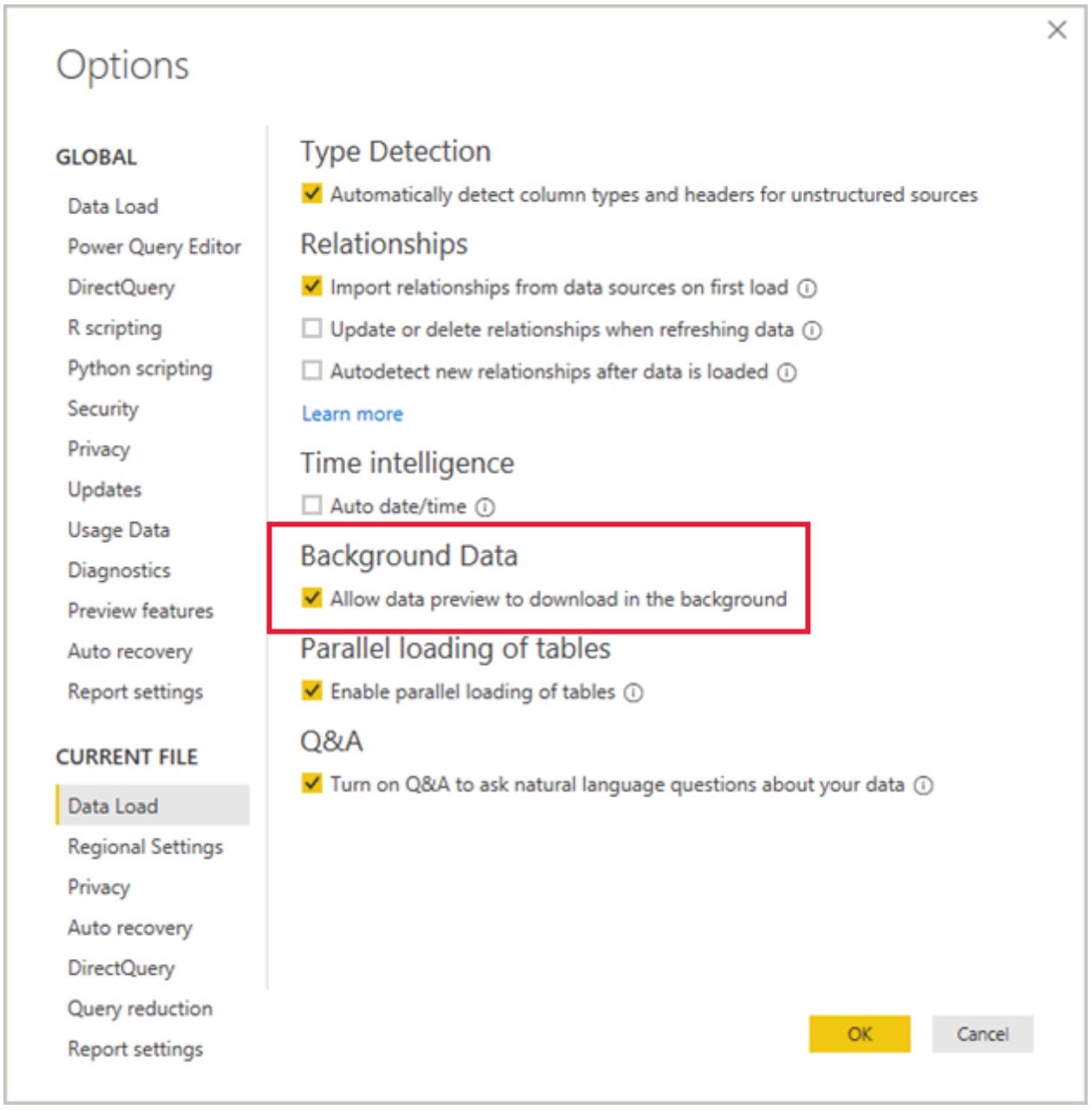
This article targets Import data modelers working with Power BI Desktop.

By default, when Power Query imports data, it also caches up to 1000 rows of preview data for each query. Preview data helps to present you with a quick preview of source data, and of transformation results for each step of your queries. It's stored separately on-disk and not inside the Power BI Desktop file.

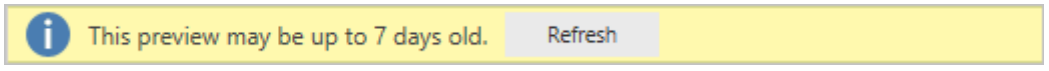
However, when your Power BI Desktop file contains many queries, retrieving and storing preview data can extend the time it takes to complete a refresh.

Recommendation

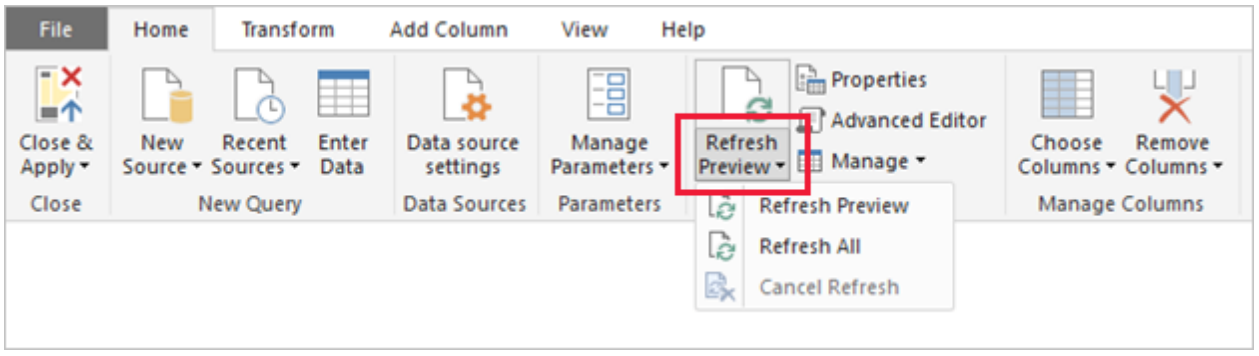
You'll achieve a faster refresh by setting the Power BI Desktop file to update the preview cache *in the background*. In Power BI Desktop, you enable it by selecting *File* > *Options and settings* > *Options*, and then selecting the *Data Load* page. You can then turn on the **Allow data preview to download in the background** option. Note this option can only be set for the current file.



Enabling background refresh can result in preview data becoming out of date. If it occurs, the Power Query Editor will notify you with the following warning:



It's always possible to update the preview cache. You can update it for a single query, or for all queries by using the **Refresh Preview** command. You'll find it on the **Home** ribbon of the Power Query Editor window.



Related content

For more information related to this article, check out the following resources:

- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

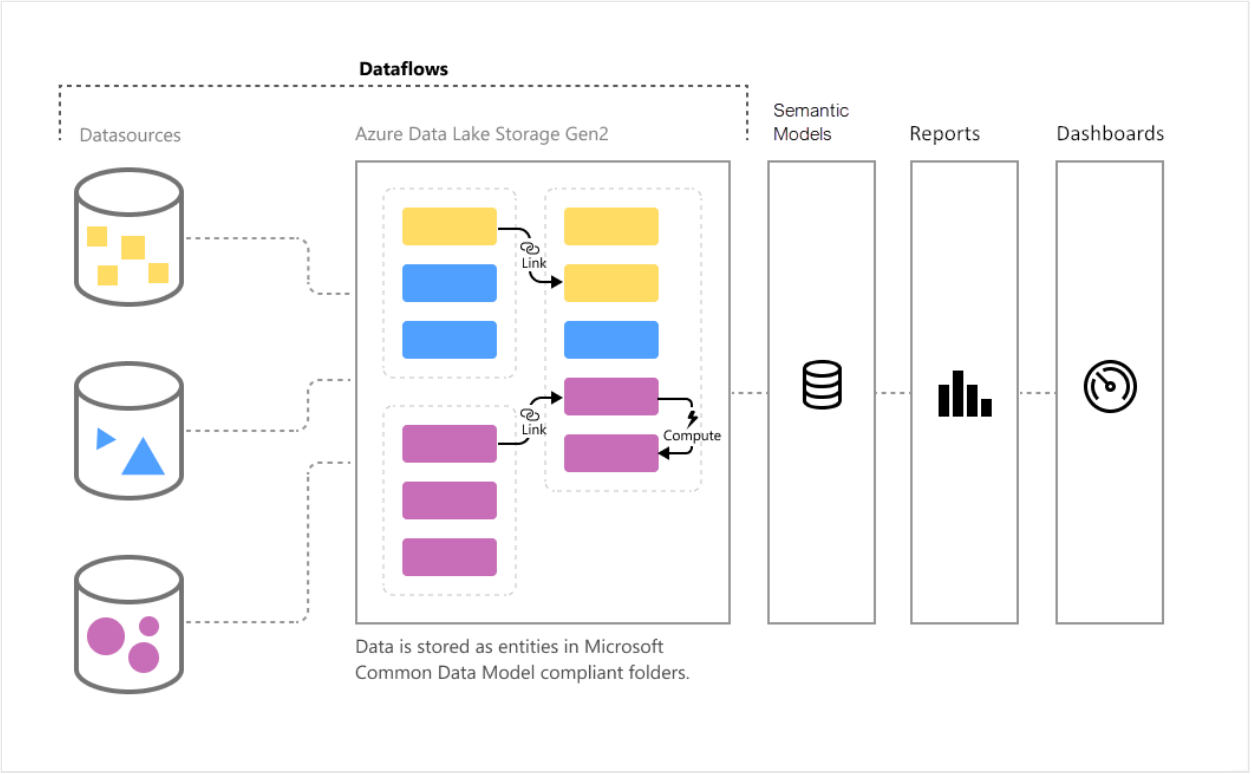
Was this page helpful? Yes No

Provide product feedback | Ask the community

Introduction to dataflows and self-service data prep

Article • 09/25/2024

As data volume continues to grow, so does the challenge of wrangling that data into well-formed, actionable information. We want data that's ready for analytics, to populate visuals, reports, and dashboards, so we can quickly turn our volumes of data into actionable insights. With self-service data prep for big data in Power BI, you can go from data to Power BI insights with just a few actions.



> **Tip**
>
> You can also try Dataflow Gen2 in Data Factory in Microsoft Fabric, an all-in-one analytics solution for enterprises. Microsoft Fabric covers everything from data movement to data science, real-time analytics, business intelligence, and reporting. Learn how to start a new trial for free.

Dataflows are designed to support the following scenarios:

- Create reusable transformation logic that can be shared by many semantic models and reports inside Power BI. Dataflows promote reusability of underlying data elements, preventing the need to create separate connections with your cloud or on-premises data sources.

- Persist data in your own Azure Data Lake Gen 2 storage, enabling you to expose it to other Azure services outside Power BI.
- Create a single source of truth, curated from raw data using industry standard definitions, which can work with other services and products in the Power Platform. Encourage uptake by removing analysts' access to underlying data sources.
- Strengthen security around underlying data sources by exposing data to report creators in dataflows. This approach allows you to limit access to underlying data sources, reducing the load on source systems, and gives administrators finer control over data refresh operations.
- If you want to work with large data volumes and perform ETL at scale, dataflows with Power BI Premium scales more efficiently and gives you more flexibility. Dataflows supports a wide range of cloud and on-premises sources.

You can use Power BI Desktop and the Power BI service with dataflows to create semantic models, reports, dashboards, and apps that use the Common Data Model. From these resources, you can gain deep insights into your business activities. Dataflow refresh scheduling is managed directly from the workspace in which your dataflow was created, just like your semantic models.

> ⓘ **Note**
>
> Dataflows may not be available in the Power BI service for all U.S. Government DoD customers. For more information about which features are available, and which are not, see Power BI feature availability for U.S. Government customers.

Related content

This article provided an overview of self-service data prep for big data in Power BI, and the many ways you can use it.

The following articles provide more information about dataflows and Power BI:

- Creating a dataflow
- Configure and consume a dataflow
- Configuring Dataflow storage to use Azure Data Lake Gen 2
- Premium features of dataflows
- AI with dataflows
- Dataflows considerations and limitations
- Dataflows best practices

- Power BI usage scenarios: Self-service data preparation

For more information about the Common Data Model, you can read its overview article:

- Common Data Model - overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Understand star schema and the importance for Power BI

Article • 12/30/2024

This article targets Power BI Desktop data modelers. It describes star schema design and its relevance to developing Power BI semantic models optimized for performance and usability.

Important

Power BI semantic models depend on Power Query to import or connect to data. That means you must use Power Query to transform and prepare the source data, which might be challenging when you have large data volumes or you need to implement advanced concepts like slowly changing dimensions (described later in this article).

When you're presented with these challenges, we recommend you first develop a data warehouse and Extract, Transform, and Load (ETL) processes to periodically load the data warehouse. Your semantic model can then connect to the data warehouse. For more information, see Dimensional modeling in Microsoft Fabric Warehouse.

Tip

This article isn't intended to provide a complete discussion on star schema design. For more information, refer directly to widely adopted published content, like *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd edition, 2013) by Ralph Kimball, and others.

Star schema overview

Star schema is a mature modeling approach widely adopted by relational data warehouses. It requires modelers to classify their model tables as either *dimension* or *fact*.

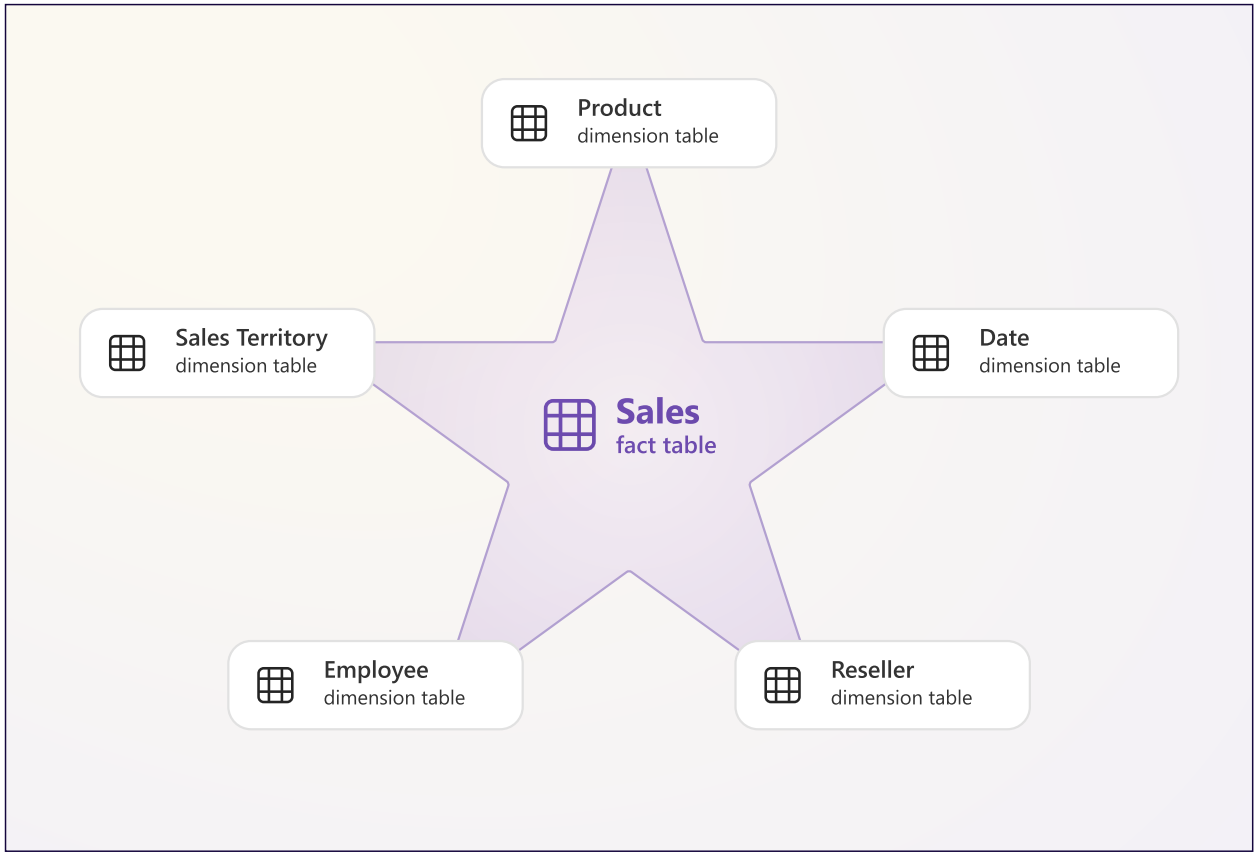
Dimension tables

Dimension tables describe business entities—the *things* you model. Entities can include products, people, places, and concepts including time itself. The most consistent table you'll find in a star schema is a date dimension table. A dimension table contains a key column (or columns) that acts as a unique identifier, and other columns. Other columns support filtering and grouping your data.

Fact tables

Fact tables store observations or events, and can be sales orders, stock balances, exchange rates, temperatures, and more. A fact table contains dimension key columns that relate to dimension tables, and numeric measure columns. The dimension key columns determine the *dimensionality* of a fact table, while the dimension key values determine the *granularity* of a fact table. For example, consider a fact table designed to store sale targets that has two dimension key columns `Date` and `ProductKey`. It's easy to understand that the table has two dimensions. The granularity, however, can't be determined without considering the dimension key values. In this example, consider that the values stored in the `Date` column are the first day of each month. In this case, the granularity is at month-product level.

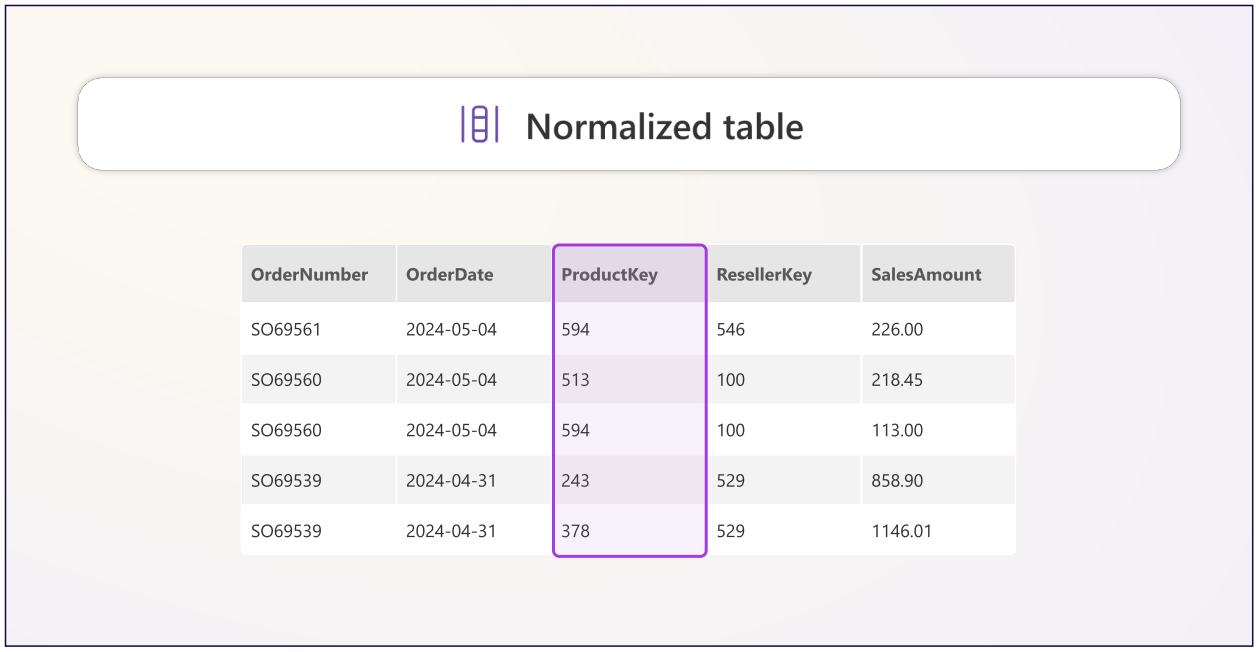
Generally, dimension tables contain a relatively small number of rows. Fact tables, on the other hand, can contain a large number of rows and continue to grow over time.



Normalization vs. denormalization

To understand some star schema concepts described in this article, it's important to know two terms: normalization and denormalization.

Normalization is the term used to describe data that's stored in a way that reduces repetitious data. Consider a table of products that has a unique key value column, like the product key, and other columns that describe product characteristics, like product name, category, color, and size. A sales table is considered normalized when it stores only keys, like the product key. In the following image, notice that only the `ProductKey` column records the product.



If, however, the sales table stores product details beyond the key, it's considered *denormalized*. In the following image, notice that the `ProductKey` and other product-related columns record the product.

When you source data from an export file or data extract, it's likely that it represents a denormalized set of data. In this case, use Power Query to transform and shape the source data into multiple normalized tables.

As described in this article, you should strive to develop optimized Power BI semantic models with tables that represent normalized fact and dimension data. However, there's one exception where a snowflake dimension might be denormalized in order to produce a single model table.

Star schema relevance to Power BI semantic models

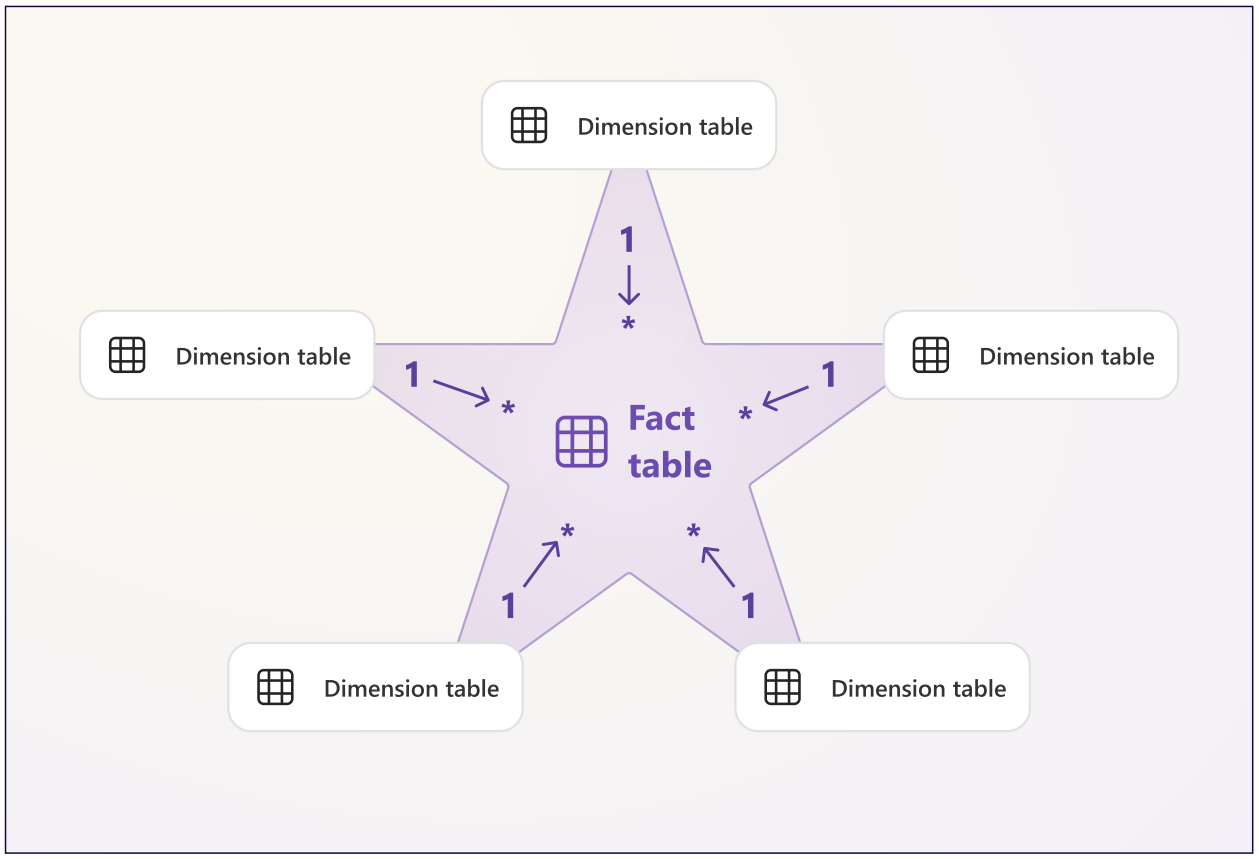
Star schema design and many related concepts introduced in this article are highly relevant to developing Power BI models that are optimized for performance and usability.

Consider that each Power BI report visual generates a query that's sent to the Power BI semantic model. Generally, queries filter, group, and summarize model data. A well-designed model, then, is one that provides tables for filtering and grouping, and tables for summarizing. This design fits well with star schema principles:

- Dimension tables enable *filtering* and *grouping*.
- Fact tables enable *summarization*.

There's no table property that modelers set to set the table type as dimension or fact. It's in fact determined by the model relationships. A model relationship establishes a

filter propagation path between two tables, and it's the cardinality property of the relationship that determines the table type. A common relationship cardinality is *one-to-many* or its inverse *many-to-one*. The "one" side is always a dimension table while the "many" side is always a fact table.



A well-structured model design includes tables that are either dimension tables or fact tables. Avoid mixing the two types together for a single table. We also recommend that you strive to deliver the right number of tables with the right relationships in place. It's also important that fact tables always load data at a consistent grain.

Lastly, it's important to understand that optimal model design is part science and part art. Sometimes you can break with good guidance when it makes sense to do so.

There are many concepts related to star schema design that can be applied to a Power BI semantic model. These concepts include:

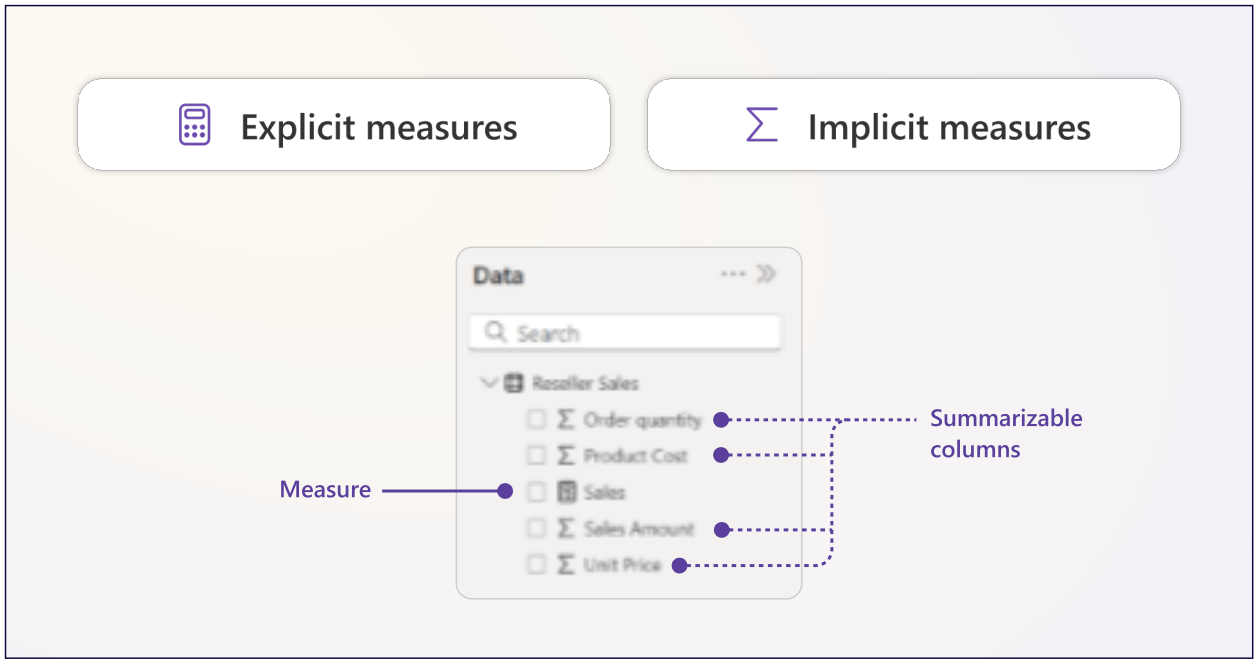
- Measures
- Surrogate keys
- Snowflake dimensions
- Role-playing dimensions
- Slowly changing dimensions
- Junk dimensions
- Degenerate dimensions
- Factless fact tables

Measures

In star schema design, a *measure* is a fact table column that stores values to be summarized. In a Power BI semantic model, a measure has a different—but similar—definition. A model supports both explicit and implicit measures.

- **Explicit measures** are expressly created and they're based on a formula written in Data Analysis Expressions (DAX) that achieves summarization. Measure expressions often use DAX aggregation functions like `SUM`, `MIN`, `MAX`, `AVERAGE`, and others to produce a scalar value result at query time (values are never stored in the model). Measure expression can range from simple column aggregations to more sophisticated formulas that override filter context and/or relationship propagation. For more information, read about DAX Basics in Power BI Desktop.
- **Implicit measures** are columns that can be summarized by a report visual or Q&A. They offer a convenience for you as a model developer, as in many instances you don't need to create (explicit) measures. For example, the Adventure Works reseller sales `Sales Amount` column can be summarized in numerous ways (sum, count, average, median, min, max, and others), without the need to create a measure for each possible aggregation type.

In the **Data** pane, explicit measures are represented by the calculator icon while implicit measures are represented by the sigma symbol (∑).



However, there are three compelling reasons why you might create measures, even for simple column-level summarizations:

- When you know your report authors will query the semantic model by using Multidimensional Expressions (MDX), the model must include explicit measures.

That's because MDX can't achieve summarization of column values. Notably, MDX is used when performing Analyze in Excel because PivotTables issue MDX queries.

- When you know your report authors will create Power BI paginated reports by using the MDX query designer, the semantic model must include explicit measures. Only the MDX query designer supports server aggregates. So, if report authors need to have measures evaluated by Power BI (instead of by the paginated report engine), they must use the MDX query designer.

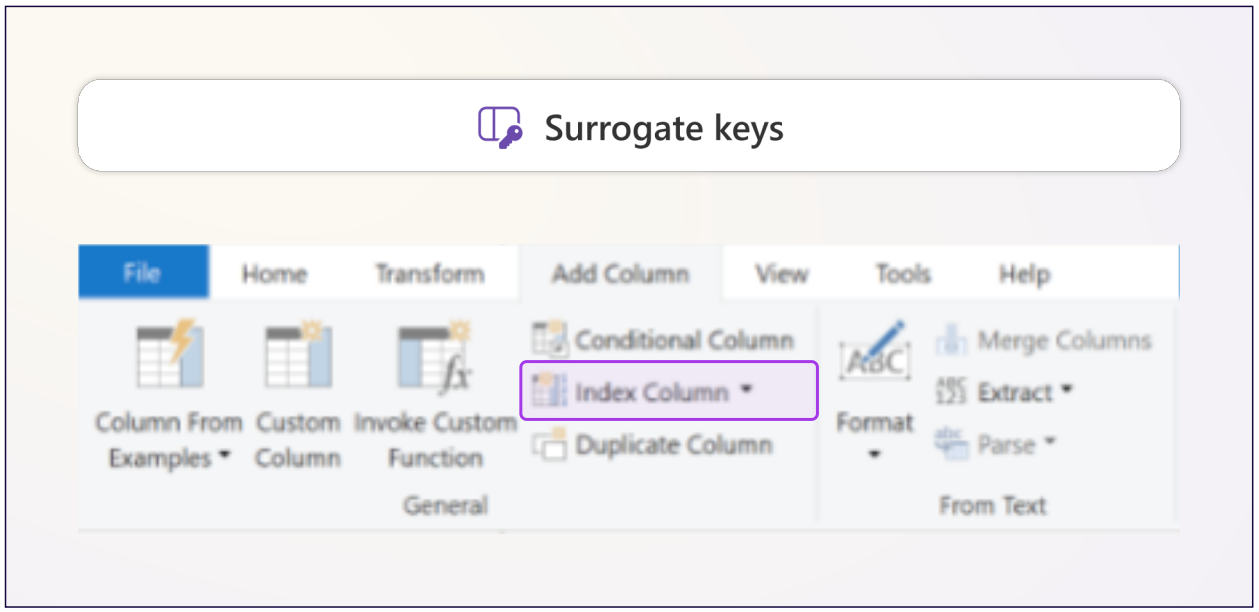
- When you want to control how your report authors summarize columns in specific ways. For example, the reseller sales `Unit Price` column (which represents a per unit rate) can be summarized, but only by using specific aggregation functions. It should never be summed, but it's appropriate to summarize by using other aggregation functions like min, max, or average. In this instance, the modeler can hide the `Unit Price` column, and create measures for all appropriate aggregation functions.

  This design approach works well for reports authored in the Power BI service and for Q&A. However, Power BI Desktop live connections allow report authors to show hidden fields in the **Data** pane, which can result in circumventing this design approach.

Surrogate keys

A *surrogate key* is a unique identifier that you add to a table to support star schema modeling. By definition, it's not defined or stored in the source data. Commonly, surrogate keys are added to relational data warehouse dimension tables to provide a unique identifier for each dimension table row.

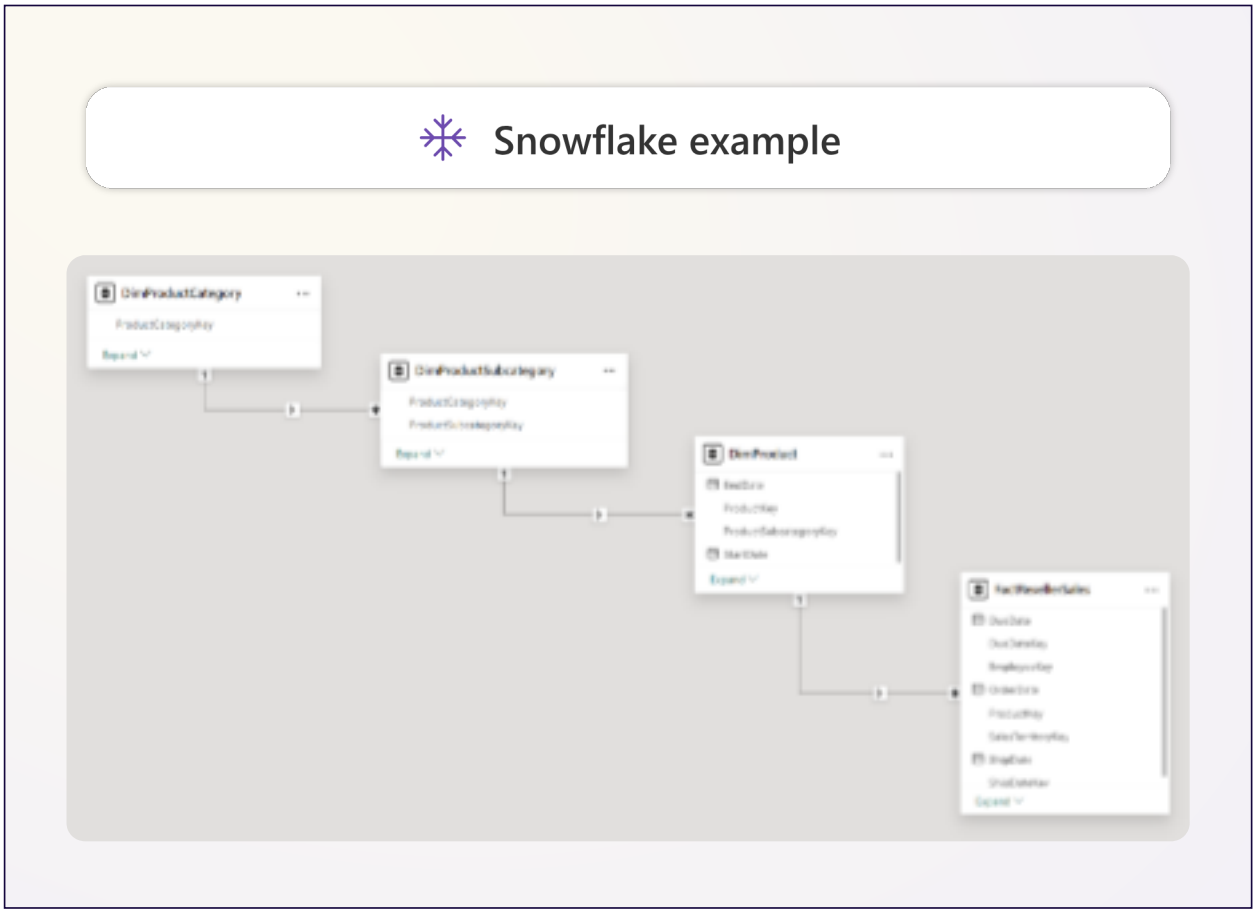
Power BI semantic model relationships are based on a single unique column in one table, which propagates filters to a single column in a different table. When a dimension table in your semantic model doesn't include a single unique column, you must add a unique identifier to become the "one" side of a relationship. In Power BI Desktop, you can achieve this requirement by adding a Power Query index column.



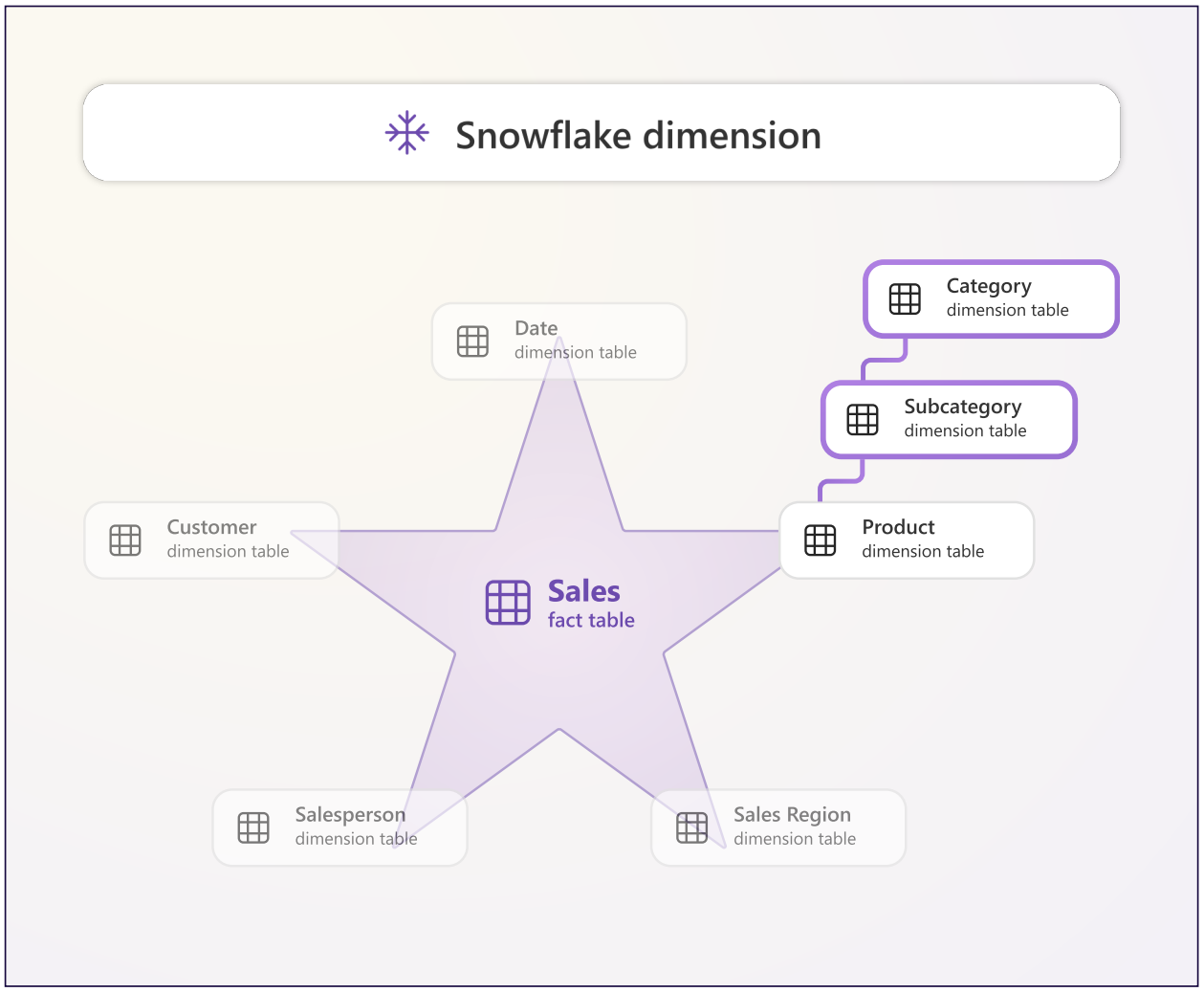
You must merge this query with the "many"-side query so that you can add the index column to it also. When you load these queries to the semantic model, you can then create a one-to-many relationship between the model tables.

Snowflake dimensions

A *snowflake dimension* is a set of normalized tables for a single business entity. For example, Adventure Works classifies products by category and subcategory. Products are assigned to subcategories, and subcategories are in turn assigned to categories. In the Adventure Works relational data warehouse, the product dimension is normalized and stored in three related tables: `DimProductCategory`, `DimProductSubcategory`, and `DimProduct`.



If you use your imagination, you can picture the normalized tables positioned outwards from the fact table, forming a snowflake design.



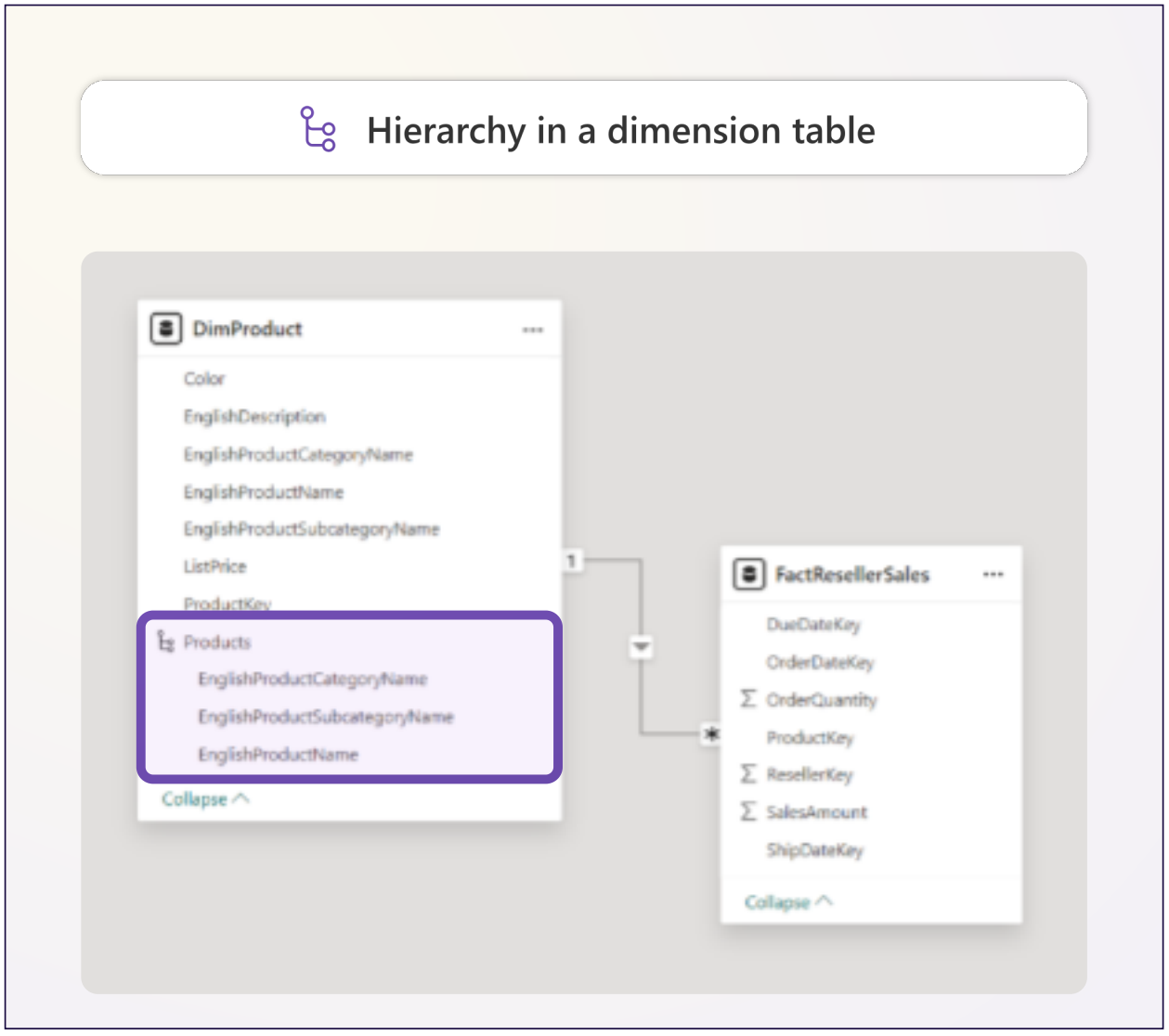
In Power BI Desktop, you can choose to mimic a snowflake dimension design (perhaps because your source data does) or combine the source tables to form a single, denormalized model table. Generally, the benefits of a single model table outweigh the benefits of multiple model tables. The most optimal decision can depend on the volumes of data and the usability requirements for the model.

When you choose to mimic a snowflake dimension design:

- Power BI loads more tables, which is less efficient from storage and performance perspectives. These tables must include columns to support model relationships, and it can result in a larger model size.
- Longer relationship filter propagation chains need to be traversed, which might be less efficient than filters applied to a single table.
- The **Data** pane presents more model tables to report authors, which can result in a less intuitive experience, especially when snowflake dimension tables contain only one or two columns.
- It's not possible to create a hierarchy that comprises columns from more than one table.

When you choose to integrate into a single model table, you can also define a hierarchy that encompasses the highest and lowest grain of the dimension. Possibly, the storage

of redundant denormalized data can result in increased model storage size, particularly for large dimension tables.



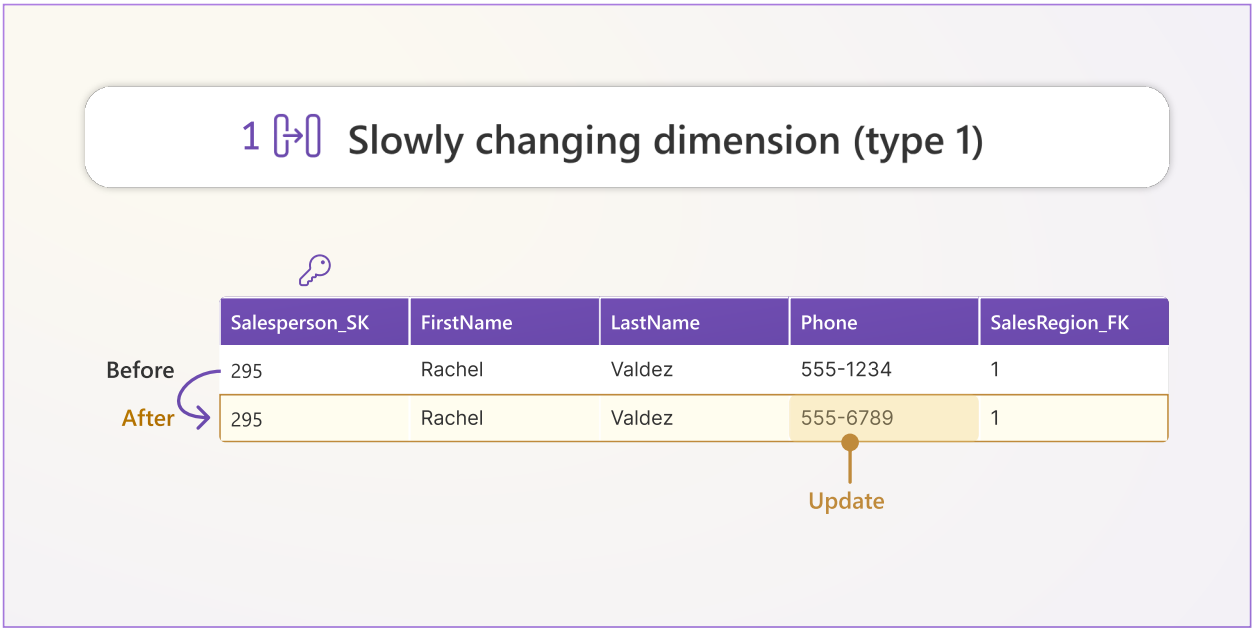
Slowly changing dimensions

A *slowly changing dimension* (or SCD) is one that appropriately manages change of dimension members over time. It applies when business entity values change *slowly* over time in an unplanned manner. A good example of an SCD is a customer dimension, because its contact detail columns like email address and phone number change infrequently. In contrast, some dimensions are considered to be *rapidly* changing when a dimension attribute changes often, like the market price of a stock. The common design approach in these instances is to store rapidly changing attribute values in a fact table measure.

Star schema design theory refers to two common SCD types: Type 1 and Type 2. A dimension table could be Type 1 or Type 2, or support both types simultaneously for different columns.

Type 1 SCD

A *Type 1* SCD always reflects the latest values, and when changes in source data are detected, the dimension table data is overwritten. This design approach is common for columns that store supplementary values, like the email address or phone number of a customer. When a customer email address or phone number changes, the dimension table updates the customer row with the new values. It's as if the customer always had this contact information.



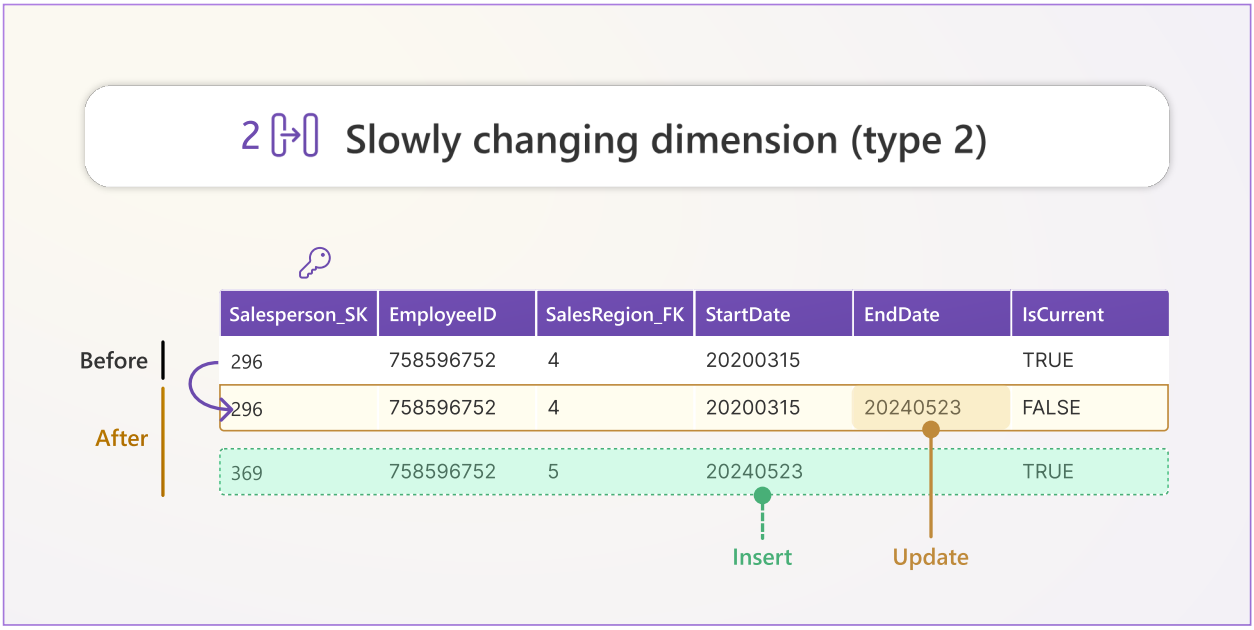
A non-incremental refresh of a Power BI model dimension table achieves the result of a Type 1 SCD. It refreshes the table data to ensure the latest values are loaded.

Type 2 SCD

A *Type 2* SCD supports versioning of dimension members. If the source system doesn't store versions, it's usually the data warehouse load process that detects changes and appropriately manages the change in a dimension table. In this case, the dimension table must use a surrogate key to provide a unique reference to a *version* of the dimension member. It also includes columns that define the date range validity of the version (for example, `StartDate` and `EndDate`) and possibly a flag column (for example, `IsCurrent`) to easily filter by current dimension members.

For example, Adventure Works assigns every salesperson to a sales region. When a salesperson relocates region, a new version of the salesperson must be created to ensure that historical facts remain associated with the former region. To support accurate historic analysis of sales by salesperson, the dimension table must store versions of salespeople and their associated region(s). The table should also include start and end date values to define the time validity. Current versions might define an

empty end date (or 12/31/9999), which indicates that the row is the current version. The table must also have a surrogate key because the business key (in this instance, employee ID) won't be unique.



It's important to understand that when the source data doesn't store versions, you must use an intermediate system (like a data warehouse) to detect and store changes. The table load process must preserve existing data and detect changes. When a change is detected, the table load process must expire the current version. It records these changes by updating the `EndDate` value and inserting a new version with the `StartDate` value commencing from the previous `EndDate` value. Also, related facts must use a time-based lookup to retrieve the dimension key value relevant to the fact date. A Power BI semantic model uses Power Query, and so it can't produce this result. It can, however, load data from a pre-loaded SCD Type 2 dimension table.

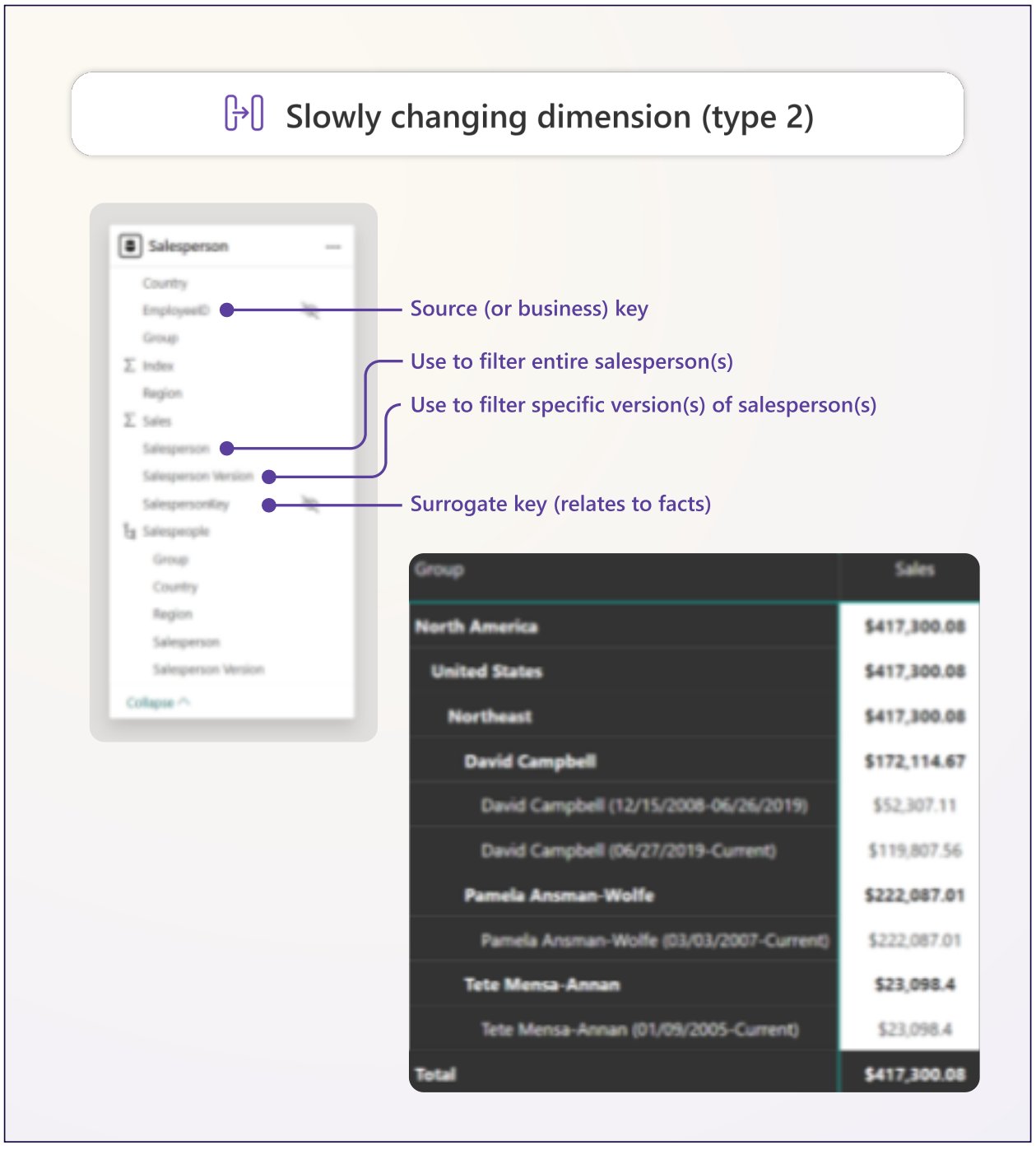
> **Tip**
>
> To learn how to implement a Type 2 SCD dimension table in a Fabric warehouse, see **Manage historical change**.

The Power BI semantic model should support querying historical data for a member, regardless of change, and for a version of the member, which represents a particular state of the member in time. In the context of Adventure Works, this design enables you to query the salesperson regardless of assigned sales region, or for a particular version of the salesperson.

To achieve this requirement, the Power BI semantic model dimension table must include a column for filtering the salesperson, and a different column for filtering a specific version of the salesperson. It's important that the version column provides a non-

ambiguous description, like `David Campbell (12/15/2008-06/26/2019)` or `David Campbell (06/27/2019-Current)`. It's also important to educate report authors and consumers about the basics of SCD Type 2, and how to achieve appropriate report designs by applying correct filters.

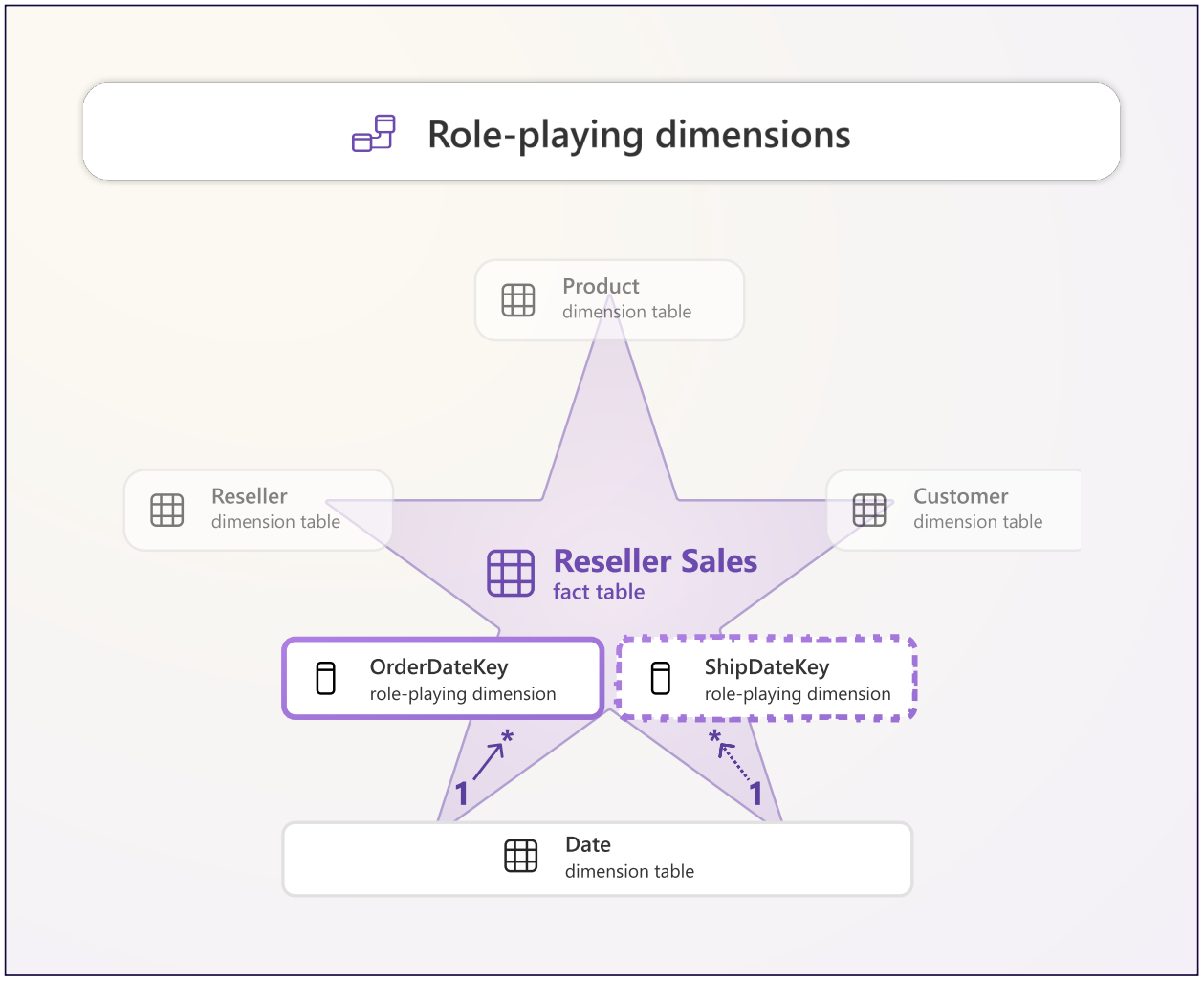
It's a good design practice to include a hierarchy that allows visuals to drill down to the version level.



Role-playing dimensions

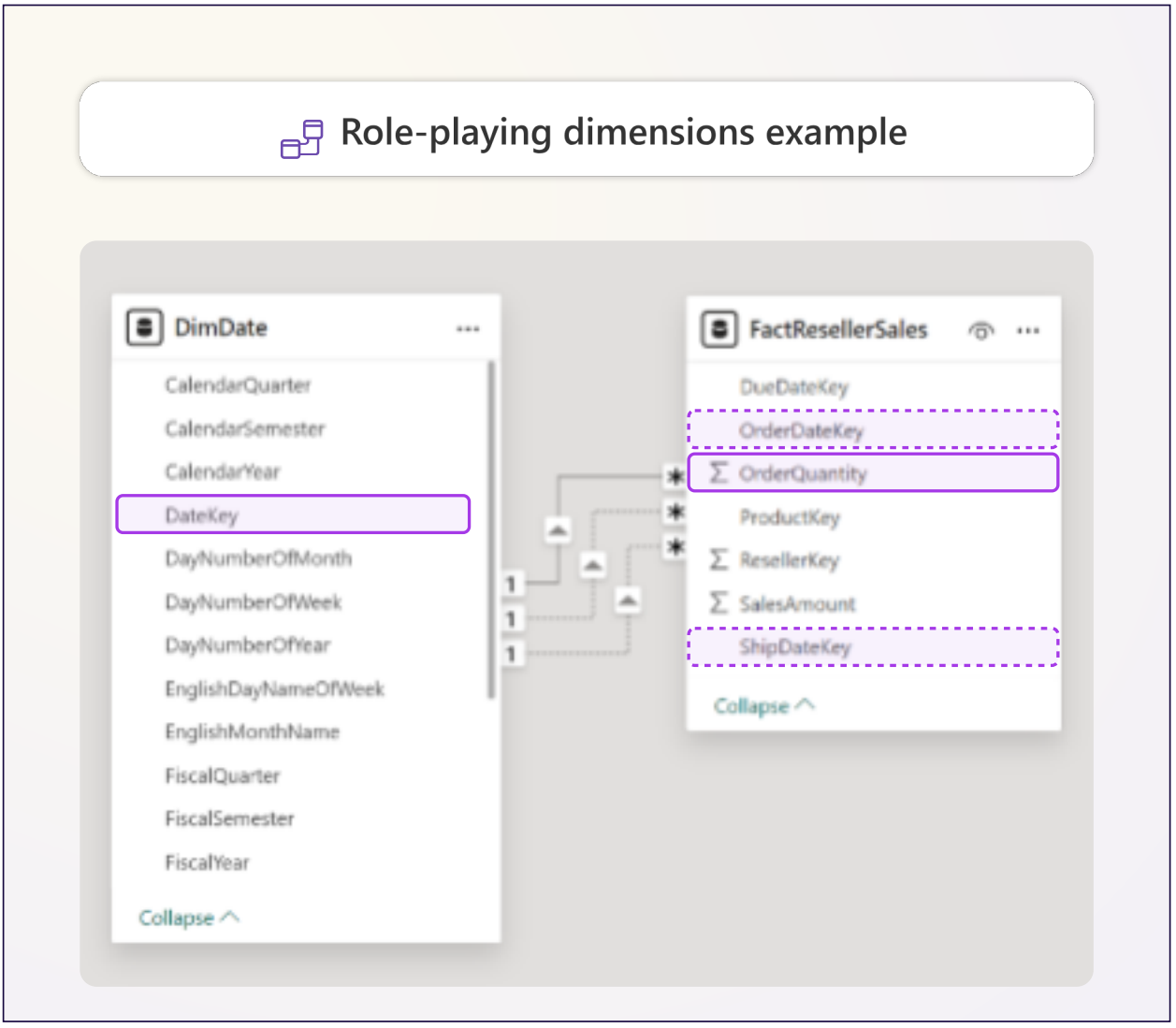
A *role-playing dimension* is a dimension that can filter related facts differently. For example, at Adventure Works the date dimension table has three relationships to the

reseller sales facts. The same dimension table can be used to filter the facts by order date, ship date, or delivery date.



In a data warehouse, the accepted design approach is to define a single date dimension table. At query time, the "role" of the date dimension is established by which fact column you use to join the tables. For example, when you analyze sales by order date, the table join relates to the reseller sales order date column.

In a Power BI semantic model, this design can be imitated by creating multiple relationships between two tables. In the Adventure Works example, the date and reseller sales tables would have three relationships.



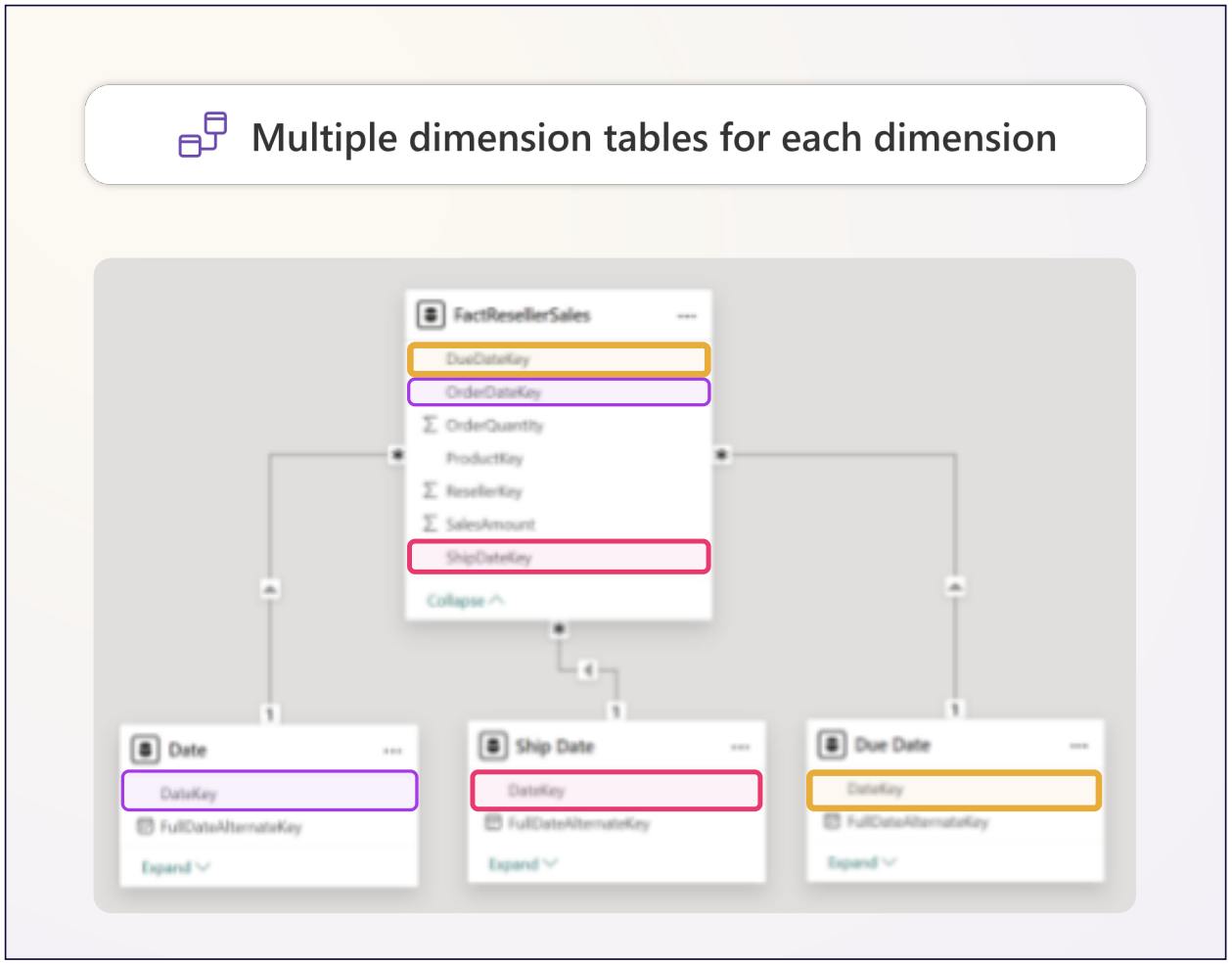
While this design is possible, there can only be one *active* relationship between two Power BI semantic model tables. All remaining relationships must be set to inactive. Having a single active relationship means there's a default filter propagation from date to reseller sales. In this instance, the active relationship is set to the most common filter that's used by reports, which at Adventure Works is the order date relationship.

The only way to use an inactive relationship is to define a DAX expression that uses the USERELATIONSHIP function. In our example, the model developer must create measures to enable analysis of reseller sales by ship date and delivery date. This work can be tedious, especially when the reseller table defines many measures. It also creates a cluttered **Data** pane that has an overabundance of measures. There are other limitations, too:

- When report authors rely on summarizing columns rather than defining measures, they can't achieve summarization for the inactive relationships without writing a report-level measure. Report-level measures can only be defined when authoring reports in Power BI Desktop.
- With only one active relationship path between date and reseller sales, it's not possible to simultaneously filter reseller sales by different types of dates. For

example, you can't produce a visual that plots order date sales by shipped sales.

To overcome these limitations, a common Power BI modeling technique is to create a dimension table for each role-playing instance. You can create each dimension table as a referencing query using Power Query, or a calculated table using DAX. The model can contain a `Date` table, a `Ship Date` table and a `Delivery Date` table, each with a single and active relationship to their respective reseller sales table columns.



This design approach doesn't require you to define multiple measures for different date roles, and it allows simultaneous filtering by different date roles. A minor price to pay with this design approach, however, is that there will be duplication of the date dimension table resulting in an increased model storage size. Because dimension tables typically store fewer rows relative to fact tables, it's rarely a concern.

We recommend you follow good design practices when creating model dimension tables for each role:

- Ensure that the column names are self-describing. While it's possible to have a `Year` column in all date tables (column names are unique within their table), it's not self-describing by default visual titles. Consider renaming columns in each dimension role table so that the `Ship Date` table has a year column named `Ship Year`, and so on.

- When relevant, ensure that table descriptions provide feedback to report authors (through **Data** pane tooltips) about how filter propagation is set up. This clarity is important when the model contains a generically named table, like `Date`, which is used to filter many fact tables. In the case that this table has, for example, an active relationship to the reseller sales order date column, consider providing a table description like `Filters reseller sales by order date`.

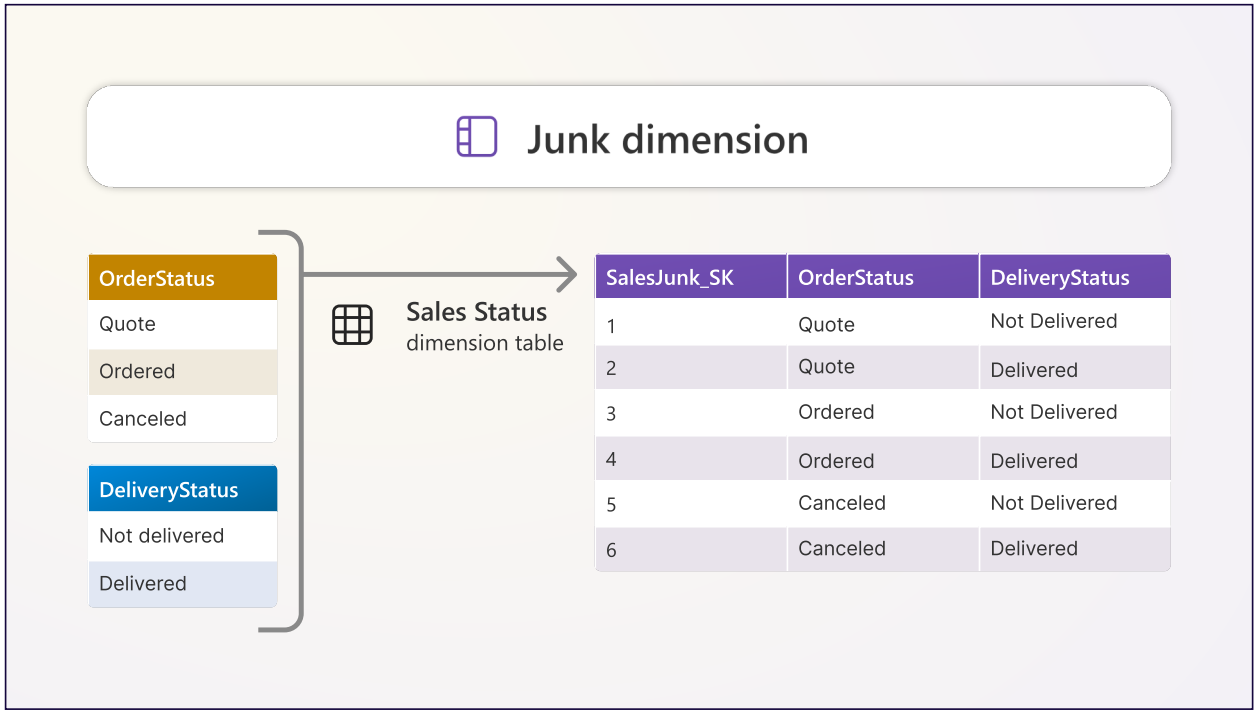
For more information, see Active vs inactive relationship guidance.

Junk dimensions

A *junk dimension* is useful when there are many dimensions, especially consisting of few attributes (perhaps one), and when these attributes have few values. Good candidates include order status columns, or customer demographic columns like gender or age group.

The design objective of a junk dimension is to consolidate many *small* dimensions into a single dimension to reduce the model storage size and also reduce **Data** pane clutter by surfacing fewer model tables.

A junk dimension table is typically the Cartesian product of all dimension attribute members, with a surrogate key column to uniquely identify each row. You can build the dimension in a data warehouse, or by using Power Query to create a query that performs full outer query joins, then adds a surrogate key (index column).



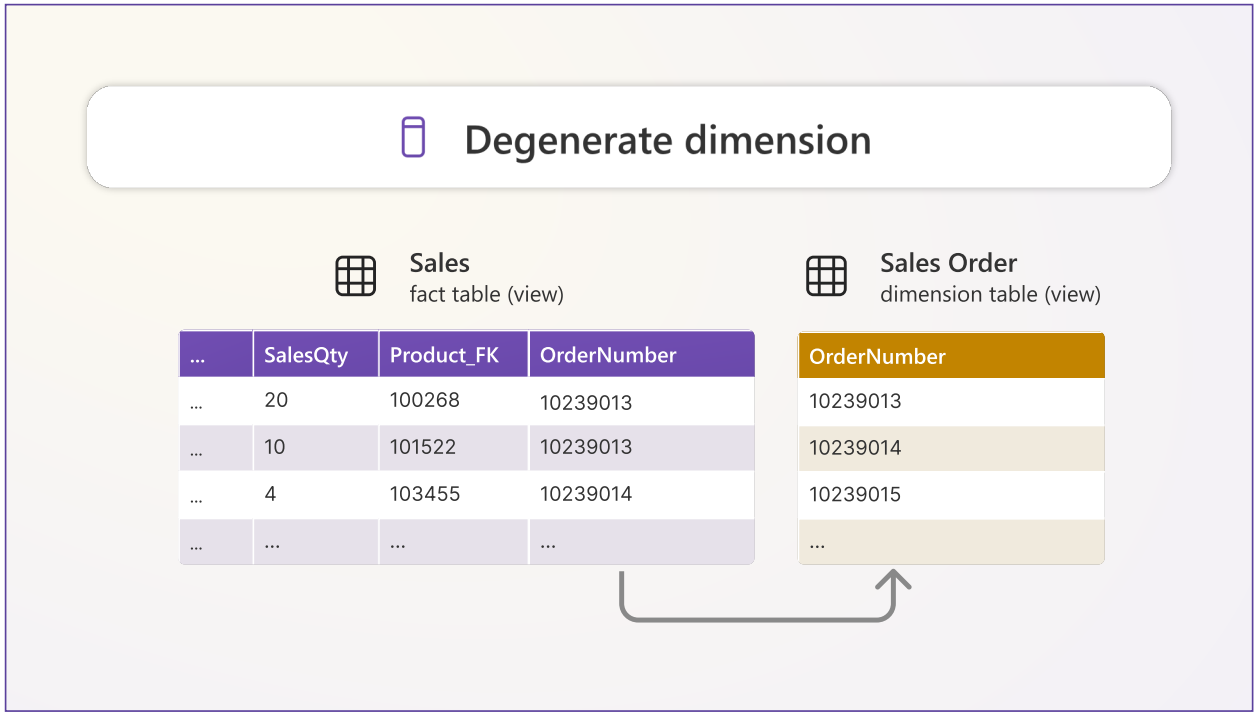
You load this query to the model as a dimension table. You also need to merge this query with the fact query so the index column is loaded to the model to support the

creation of a "one-to-many" model relationship.

Degenerate dimensions

A *degenerate dimension* refers to an attribute of the fact table that's required for filtering. At Adventure Works, the reseller sales order number is a good example. In this instance, it doesn't make sense to create an independent table consisting of just this one column because it would increase the model storage size and result in **Data** pane clutter.

In the Power BI semantic model, it can be appropriate to add the sales order number column to the fact table to allow filtering or grouping by sales order number. It's an exception to the formerly introduced rule that you shouldn't mix table types (generally, model tables should be either dimension or fact).



However, if the Adventure Works resellers sales table has order number *and* order line number columns, and they're required for filtering, creating a degenerate dimension table would be a good design. For more information, see One-to-one relationship guidance (Degenerate dimensions).

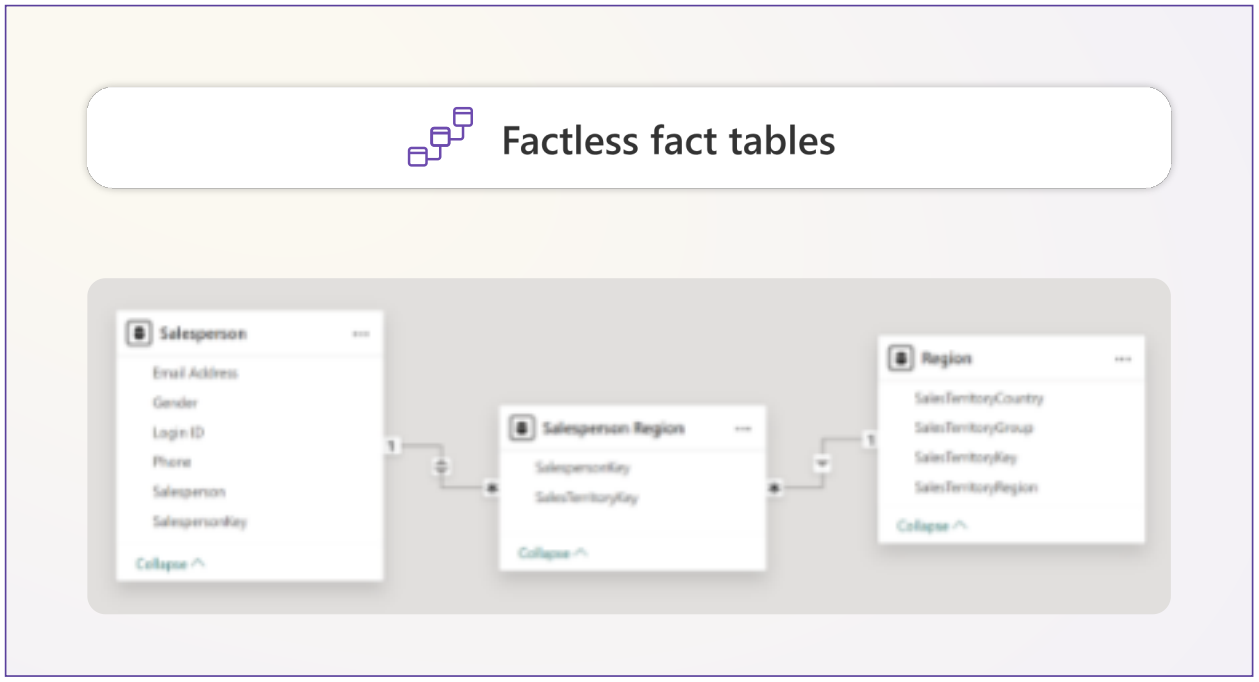
Factless fact tables

A *factless fact* table doesn't include any measure columns. It contains only dimension keys.

A factless fact table could store observations defined by dimension keys. For example, at a particular date and time, a particular customer signed in to your web site. You could define a measure to count the rows of the factless fact table to perform analysis of when and how many customers signed in.

A more compelling use of a factless fact table is to store relationships between dimensions, and it's a Power BI semantic model design approach we recommend for defining many-to-many dimension relationships. In a many-to-many dimension relationship design, the factless fact table is referred to as a *bridging table*.

For example, consider that salespeople can be assigned to one *or more* sales regions. The bridging table would be designed as a factless fact table consisting of two columns: salesperson key and region key. Duplicate values can be stored in both columns.



This many-to-many design approach is well documented, and it can be achieved without a bridging table. However, the bridging table approach is considered the best practice when relating two dimensions. For more information, see Many-to-many relationship guidance (Relate two dimension tables).

Related content

For more information about star schema design or Power BI semantic model design, see the following articles:

- Dimensional modeling Wikipedia article
- Model relationships in Power BI Desktop
- One-to-one relationship guidance
- Many-to-many relationship guidance

- Bi-directional relationship guidance
- Active vs inactive relationship guidance
- Dimensional modeling in Microsoft Fabric Warehouse
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Data reduction techniques for Import modeling

Article • 12/30/2024

This article targets Power BI Desktop data modelers who develop and publish Power BI semantic models. Specifically, it describes various techniques to help reduce the data loaded into Import models.

Import models are loaded with data that's compressed and optimized, and then stored to disk by the VertiPaq storage engine. When source data is loaded into memory, it's possible to achieve 10x compression, and so it's reasonable to expect that 10 GB of source data can compress to about 1 GB in size. Further, when persisted to disk an extra 20% reduction can be achieved.

Despite the efficiencies achieved by the VertiPaq storage engine, you should strive to minimize the data that's loaded into your models. That's especially true for large models, or models that you anticipate will grow to become large over time. Four compelling reasons include:

- Larger model sizes might not be supported by your capacity. Shared capacity can host models up to 1 GB in size, while Premium capacities can host larger models depending on the SKU. For more information, see Large semantic models in Power BI Premium.
- Smaller model sizes reduce contention for capacity resources, in particular memory. Many smaller models in a capacity can be concurrently loaded for longer periods of time, resulting in lower eviction rates.
- Smaller model sizes achieve faster data refresh, resulting in lower latency reporting, higher semantic model refresh throughput, and less pressure on source system and capacity resources.
- Smaller table row counts can lead to faster calculation evaluations, which result in better overall query performance.

> **ⓘ Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.

For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Remove unnecessary columns

Model table columns serve two main purposes:

- **Reporting**, to achieve report designs that appropriately filter, group, and summarize model data.
- **Model structure**, by supporting model relationships, model calculations, security roles, and even data color formatting.

You can probably remove any column that doesn't serve either of these purposes. Removing a column from a table is sometimes referred to as *vertical filtering*.

We recommend that you design models with exactly the right number of columns based on your known reporting requirements. Your requirements might change over time, but bear in mind that it's easier to add columns later than it is to remove them later. Removing columns can break reports or the model structure.

Remove unnecessary rows

You should load model tables with as few rows as possible. You can achieve that by loading filtered rowsets into model tables for two different reasons: to filter by time or by entity. Removing rows is sometimes referred to as *horizontal filtering*.

- **Filtering by time** involves limiting the amount of *data history* loaded into fact tables (and limiting the date rows loaded into the model date tables). We recommend that you don't load all available history by default, unless it's a known reporting requirement. You can implement time-based Power Query filters with parameters, and even set them to use relative time periods (relative to the refresh date—for example, the past five years). Also, bear in mind that a retrospective change to time filters won't break reports; it will just result in less (or more) data history available in reports.
- **Filtering by entity** involves loading a subset of source data into the model. For example, instead of loading sales facts for all sales regions, only load facts for a single region. This design approach results in many smaller models, and it can also eliminate the need to define row-level security (RLS)—but it requires granting specific semantic model permissions in the Power BI service, and creating duplicate reports that connect to each semantic model. You can use Power Query

parameters and Power BI Template files to simplify management and publication. For more information, see Create and use report templates in Power BI Desktop.

Group by and summarize

Perhaps the most effective technique to reduce a model size is to load pre-summarized data. You can use this technique to raise the grain of fact tables. There's a distinct trade-off, however, resulting in loss of detail.

Consider an example where a source sales fact table stores one row per order line. Significant data reduction can be achieved by summarizing all sales metrics, grouping by date, customer, and product. An even more significant data reduction can be achieved by grouping by date *at month level*. While it could achieve a possible 99% reduction in model size, reporting at day level or individual order line level is no longer possible. Deciding to summarize fact data always involves tradeoffs. Tradeoff could be mitigated by a model design that includes some tables in DirectQuery storage mode, which is described later in this article.

Optimize column data types

The VertiPaq storage engine uses separate internal data structures for each column. By design, these data structures achieve the highest optimizations for numeric column data, which uses *value encoding*. Text and other non-numeric data, however, uses *hash encoding*. Hash encoding requires the storage engine to assign a numeric identifier to each unique value contained in the column. It's the numeric identifier then, that's stored in the data structure, requiring a hash lookup during storage and querying.

In some specific instances, you can convert source text data into numeric values. For example, a sales order number might be consistently prefixed by a text value (for example `SO123456`). In this case, the prefix `SO` could be removed, and the order number value converted to a whole number. For large tables, this modification can result in significant data reduction, especially when the column contains unique or high-cardinality values.

In this example, we recommend that you set the column default summarization property to `Do Not Summarize`. It helps to avoid inappropriate summarization of the order number values.

Preference for custom columns

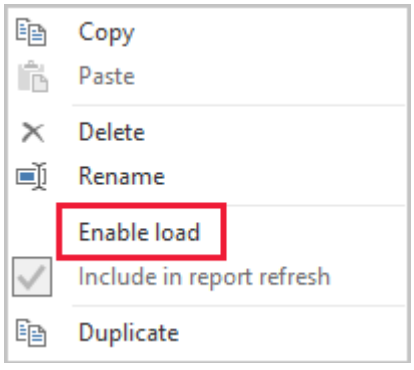
The VertiPaq storage engine stores model calculated columns (defined in DAX) just like regular Power Query-sourced columns. However, the internal data structures are stored slightly differently, and typically achieve less efficient compression. Also, the data structures are built once all Power Query tables are loaded, which can result in extended data refresh times. It's therefore less efficient to add table columns as *calculated* columns than Power Query *computed* columns (defined in M).

Whenever possible, preference creating custom columns in Power Query. When the source is a database, you can achieve greater load efficiency in two ways: The calculation can be defined in the SQL statement (using the native query language of the provider), or it can be materialized as a column in the data source.

However, in some instances, model calculated columns might be the better choice. That's true when the formula involves evaluating measures, or it requires specific modeling functionality only supported in DAX functions. For information on one such example, see Understanding functions for parent-child hierarchies in DAX.

Disable Power Query query load

Power Query queries that are intended to support data integration with other queries shouldn't be loaded to the model. To avoid loading these queries to the model, ensure that you disable the query load in these instances.



Disable auto date/time

Power BI Desktop includes an option called *Auto date/time*. When enabled, it creates hidden auto date/time tables for each date column in the model. This option supports report authors when configuring filters, grouping, and drill-down actions for calendar time periods. The hidden tables are in fact calculated tables that increase the size of the model.

For more information, see Auto date/time guidance in Power BI Desktop.

Use DirectQuery storage mode

DirectQuery storage mode is an alternative to Import storage mode. DirectQuery model tables don't import data. Instead, they consist only of metadata defining the table structure. When the table is queried, native queries are used to retrieve data from the underlying data source. When you combine Import and DirectQuery storage mode tables in a single model, it's known as a *Composite model*.

An effective technique to reduce the model size is to set the storage mode for larger fact tables to DirectQuery. Consider that this design approach often works well with the Group by and summarize technique introduced earlier. For example, summarized sales data could be used to achieve high performance summary reporting. A drillthrough page could display granular sales for specific (and narrow) filter context, displaying all in-context sales orders. In this example, the drillthrough page could include visuals based on a DirectQuery model table to retrieve the sales order data.

There are, however, many security and performance implications related to DirectQuery storage mode and Composite models. For more information, see Use composite models in Power BI Desktop.

Related content

For more information related to this article, check out the following articles:

- Semantic model modes in the Power BI service
- Storage mode in Power BI Desktop
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Design guidance for date tables in Power BI Desktop

Article • 12/30/2024

This article describes good design practices for creating date tables in your data models when using Power BI Desktop.

A date table is the most common table used for analysis. It stores one row per date, and it supports the common requirement to filter or group by specific periods of dates, like years, quarters, or months.

What's more, in order to work with Data Analysis Expressions (DAX) time intelligence functions, your data model must have at least one *date table*. A date table is a model table that meets the following requirements:

- ✓ It must have a column of data type **date** (or **date/time**)—known as the *date column*.
- ✓ The date column must contain unique values.
- ✓ The date column must not contain BLANKs.
- ✓ The date column must not have any missing dates.
- ✓ The date column must span full years. A year isn't necessarily a calendar year (January-December).
- ✓ The date table must be marked as a date table.

You can use any of several techniques to add a date table to your model:

- Use Auto date/time
- Connect with Power Query (to an existing date dimension table)
- Generate with Power Query
- Generate with DAX
- Clone with DAX

> **Tip**
>
> A date table is perhaps the most consistent feature you'll add to any of your data models. Within an organization a date table should be consistently defined. So, whatever technique you decide to use, we recommend you create a **Power BI Desktop template** that includes a fully configured date table. Share the template with all data modelers in your organization. So, whenever someone develops a new data model, they can begin with a consistently defined date table.

Use Auto date/time

The Auto date/time option delivers convenient, fast, and easy-to-use time intelligence. Reports authors can work with time intelligence when filtering, grouping, and drilling down through calendar time periods.

We recommended that you keep the Auto date/time option enabled only when you work with calendar time periods, and when you have simplistic model requirements in relation to time. You might also use this option when creating ad hoc models or performing data exploration or profiling. This approach, however, doesn't support a single date table design that can propagate filters to multiple tables. For more information, see Auto date/time guidance in Power BI Desktop.

Connect with Power Query

When your data source already has a date table, we recommend you use it as the source of your model date table. It's typically the case when you're connecting to a data warehouse, as it should already have a date dimension table. This way, your model uses a single source of truth for time in your organization.

If you're developing a DirectQuery model and your data source doesn't include a date table, we strongly recommend you add a date table to the data source. It should meet all the modeling requirements of a date table. You can then use Power Query to connect to the date table. This way, your model calculations can use the DAX time intelligence capabilities.

Generate with Power Query

You can generate a date table by using Power Query. For more information, see Chris Webb's blog entry Generating A Date Dimension Table In Power Query.

> **Tip**
>
> If you don't have a data warehouse or other consistent definition for time in your organization, consider using Power Query to publish a **dataflow**. Then, have all data modelers connect to the dataflow to add date tables to their models. The dataflow becomes the single source of truth for time in your organization.

If you need to generate a date table, consider doing it with DAX. You might find it's easier. What's more, it's likely to be more convenient, because DAX includes some built-

in intelligence to simplify creating and managing date tables.

Generate with DAX

You can generate a date table in your model by creating a calculated table using either the CALENDAR or CALENDARAUTO DAX function. Each function returns a single-column table of dates. You can then extend the calculated table with calculated columns to support your date interval filtering and grouping requirements.

- Use the `CALENDAR` function when you want to define a date range. You pass in two values: the start date and end date. These values can be defined by other DAX functions, like `MIN(Sales[OrderDate])` or `MAX(Sales[OrderDate])`.
- Use the `CALENDARAUTO` function when you want the date range to automatically encompass all dates stored in the model. You can pass in a single optional parameter that's the end month of the year (if your year is a calendar year, which ends in December, you don't need to pass in a value). It's a helpful function because it ensures that full years of dates are returned and so meets the requirement for a marked date table. What's more, you don't need to extend the table with rows for future years. That's because When a data refresh completes, it triggers the recalculation of the table. A recalculation automatically extends the date range of the table when dates for a new year are loaded into the model.

> **Tip**
>
> For more information about creating calculated tables, including an example of how to create a date table, work through the **Add calculated tables and columns to Power BI Desktop models** learning module.
>
> Also, consider using a free open-source tool to help manage dates with DAX. For more information, see **Bravo for Power BI**.

Clone with DAX

When your data model already has a date table and you need another date table—when date is a role-playing dimension dimension, you can easily clone the existing date table. You can clone a table by creating a calculated table. The calculated table expression is simply the name of the existing date table.

For example, to create a `Ship Date` table that clones the `Date` table, use the following calculated table expression.

DAX

```
Ship Date = 'Date'
```

> **Note**
>
> When you clone a table with DAX, it only creates columns. It does not apply any model properties from the cloned table, like formats or descriptions. Also, hierarchies aren't cloned.

Related content

For more information related to this article, check out the following resources:

- Auto date/time in Power BI Desktop
- Auto date/time guidance in Power BI Desktop
- Set and use date tables in Power BI Desktop
- Self-service data prep in Power BI
- CALENDAR function (DAX)
- CALENDARAUTO function (DAX)
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Auto date/time guidance in Power BI Desktop

Article • 12/30/2024

This article targets data modelers developing Import or Composite models in Power BI Desktop. It provides guidance, recommendations, and considerations when using Power BI Desktop *Auto date/time* in specific situations.

> **Note**
>
> For an overview and general introduction to Auto date/time, see Auto date/time in Power BI Desktop.

The *Auto date/time* option delivers convenient, fast, and easy-to-use time intelligence. Reports authors can work with time intelligence when filtering, grouping, and drilling down through calendar time periods.

Considerations

The following bulleted list describes considerations—and possible limitations—related to the *Auto date/time* option.

- **Applies to all or none**: When the *Auto date/time* option is enabled, it applies to all date columns in Import tables that aren't the "many" side of a relationship. It can't be selectively enabled or disabled on a column-by-column basis.
- **Calendar periods only**: The year and quarter columns relate to calendar periods. It means that the year begins on January 1 and finishes on December 31. There's no ability to customize the year commencement (or completion) date.
- **Customization**: It's not possible to customize the values used to describe time periods. Further, it's not possible to add additional columns to describe other time periods, for example, weeks.
- **Year filtering**: The `Quarter`, `Month`, and `Day` column values don't include the year value. For example, the `Month` column contains the month names only (that is, January, February, and others). The values are not fully self-describing, and in some report designs might not communicate the year filter context.

That's why it's important that filters or grouping must take place on the `Year` column. When drilling down by using the hierarchy, the year will be filtered, unless the `Year` level is intentionally removed. If there's no filter or group by year, a grouping by month, for example, would summarize values across all years for that month.

- **Single table date filtering**: Because each date column produces its own (hidden) auto date/time table, it's not possible to apply a time filter to one table and have it propagate to multiple model tables. Filtering in this way is a common modeling requirement when reporting on multiple subjects (fact tables) like sales and sales budget. When using auto date/time, the report author will need to apply filters to each different date column.

- **Model size**: For each date column that generates a hidden auto date/time table, it will result in an increased model size and also extend the data refresh time.

- **Other reporting tools**: It isn't possible to work with auto date/time tables when:
  - Using Analyze in Excel.
  - Using Power BI paginated report Analysis Services query designers.
  - Connecting to the model using non-Power BI report designers.

Recommendations

We recommend that you keep the *Auto date/time* option enabled only when you work with calendar time periods, and when you have simplistic model requirements in relation to time. Using this option can also be convenient when creating ad hoc models or performing data exploration or profiling.

When your data source already defines a date dimension table, this table should be used to consistently define time within your organization. It will certainly be the case if your data source is a data warehouse. Otherwise, you can generate date tables in your model by using the DAX CALENDAR or CALENDARAUTO functions. You can then add calculated columns to support the known time filtering and grouping requirements. This design approach might allow you to create a single date table that propagates to all fact tables, possibly resulting in a single table to apply time filters. For further information on creating date tables, read the Set and use date tables in Power BI Desktop article.

Tip

For more information about creating calculated tables, including an example of how to create a date table, work through the **Add calculated tables and columns to Power BI Desktop models** learning module.

If the *Auto date/time* option isn't relevant to your projects, we recommend that you disable the global *Auto date/time* option. It ensures that all new Power BI Desktop files you create won't enable the *Auto date/time* option.

Related content

For more information related to this article, check out the following resources:

- Design guidance for date tables in Power BI Desktop
- Auto date/time in Power BI Desktop
- Set and use date tables in Power BI Desktop
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

One-to-one relationship guidance

Article • 12/27/2024

This article targets you as a data modeler who works with Power BI Desktop. It provides you with guidance on working with one-to-one model relationships. A one-to-one relationship can be created when both tables each contain a column of common and unique values.

> **Note**
>
> An introduction to model relationships is not covered in this article. If you're not completely familiar with relationships, their properties or how to configure them, we recommend that you first read the Model relationships in Power BI Desktop article.
>
> It's also important that you have an understanding of star schema design. For more information, see Understand star schema and the importance for Power BI.

There are two scenarios that involve one-to-one relationships:

- **Degenerate dimensions**: You can derive a degenerate dimension from a fact table.
- **Row data spans across tables**: A single business entity or subject is loaded as two (or more) model tables, possibly because their data is sourced from different data stores. This scenario can be common for dimension tables. For example, master product details are stored in an operational sales system, and supplementary product details are stored in a different source.

  It's unusual, however, that you'd relate two fact tables with a one-to-one relationship. That's because both fact tables would need to have the same dimensionality and granularity. Also, each fact table would need unique columns to allow the model relationship to be created.

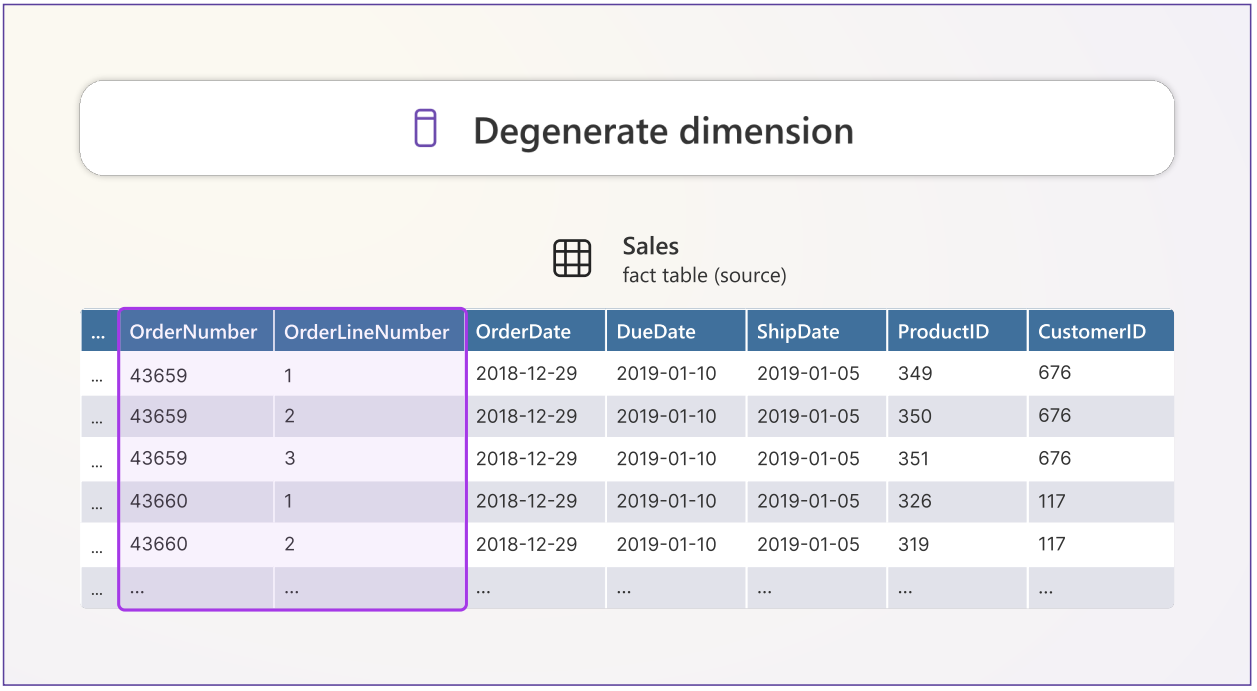
Degenerate dimensions

When columns from a fact table are used for filtering or grouping, you can consider making them available in a separate table. This way, you separate columns used for filtering or grouping from those columns used to summarize fact rows. This separation can:

- Reduce storage space.

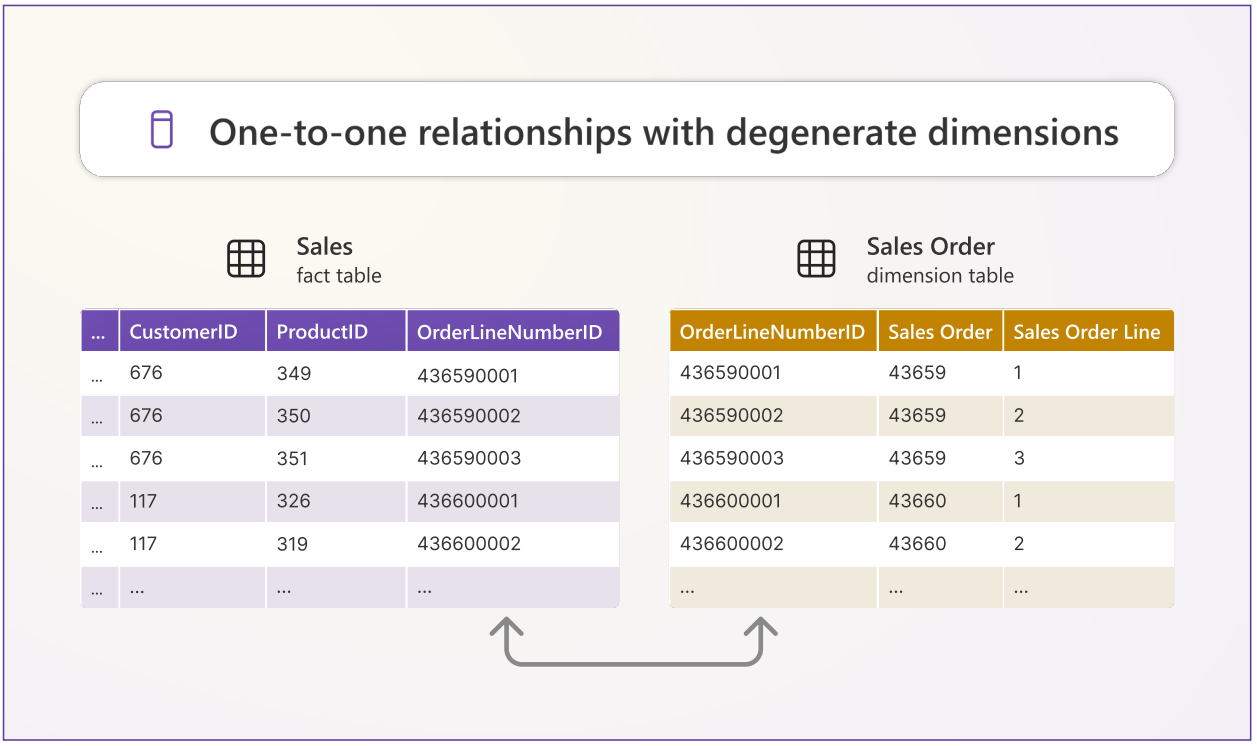
- Simplify model calculations.
- Contribute to improved query performance.
- Deliver a more intuitive **Data** pane experience to your report authors.

Consider a source table named `Sales` that stores sales order line reference details in two columns.



The `OrderNumber` column stores the order number, and the `OrderLineNumber` column stores a sequence of lines within the order.

In the following image, notice that the order number and order line number columns haven't been loaded to the `Sales` table. Instead, their values were used to create a surrogate key column named `OrderLineNumberID`. (The key value is calculated by multiplying the order number by 1000, and then adding the order line number.)



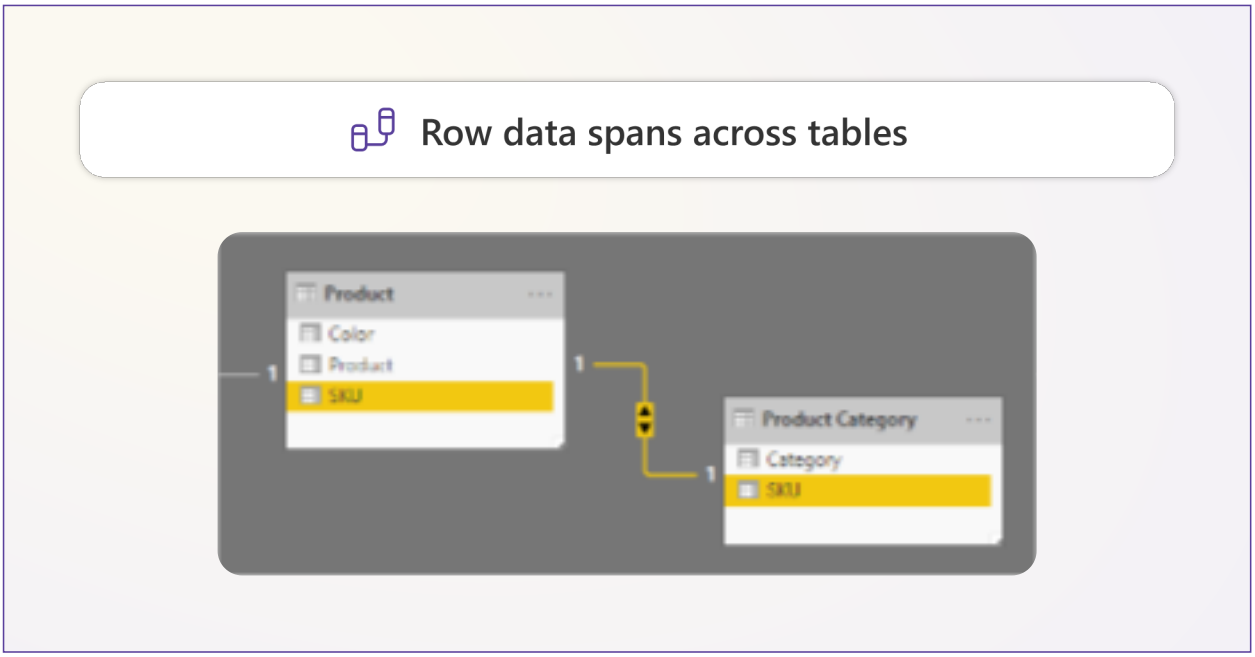
The `Sales Order` dimension table provides a rich experience for report authors with two columns: `Sales Order` and `Sales Order Line`. These particular columns support report designs that need to filter, group, or drill down through orders and order lines.

Because the `Sales Order` table is derived from the sales data, there should be exactly the same number of rows in each table. Further, there should be matching values between each `OrderLineNumberID` column.

Row data spans across tables

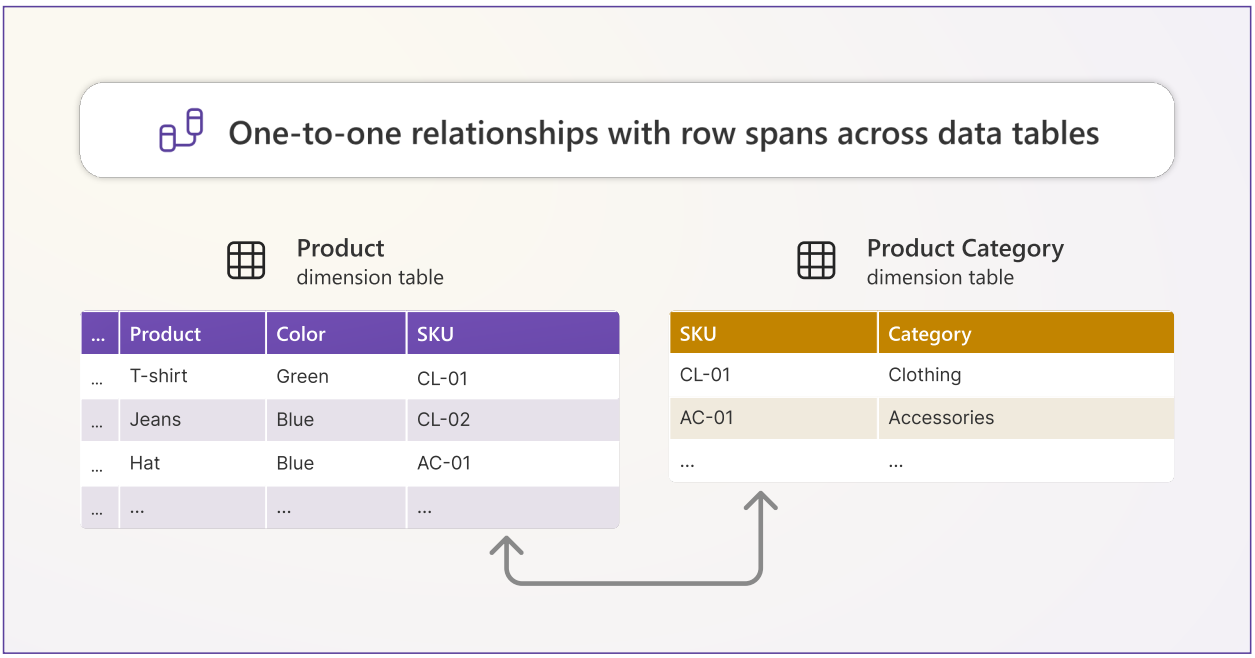
Consider an example that involves two one-to-one related dimension tables: `Product` and `Product Category`. Each table represents imported data and has a `SKU` (stock-keeping unit) column that contains unique values.

Here's a partial model diagram of the two tables.



The first table is named `Product`, and it contains three columns: `Color`, `Product`, and `SKU`. The second table is named `Product Category`, and it contains two columns: `Category` and `SKU`. A one-to-one relationship relates the two `SKU` columns. The relationship filters in both directions, which is always the case for one-to-one relationships.

To help describe how the relationship filter propagation works, the following image reveals some table rows. All examples in this article are based on this data.



...	Product	Color	SKU
...	T-shirt	Green	CL-01
...	Jeans	Blue	CL-02
...	Hat	Blue	AC-01
...	...	...	...

SKU	Category
CL-01	Clothing
AC-01	Accessories
...	...

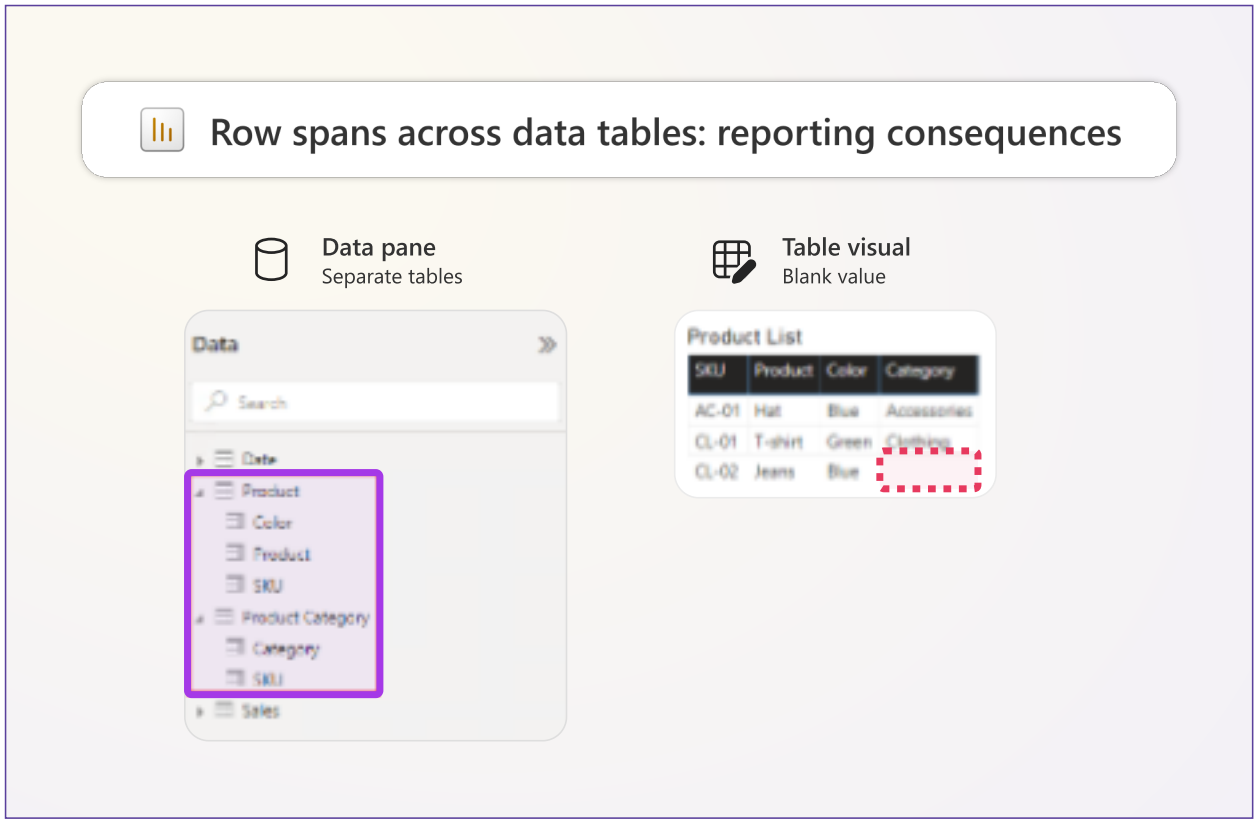
The row details for the two tables are described in the following bulleted list:

- The `Product` table has three rows:
  - `SKU` **CL-01**, `Product` **T-shirt**, `Color` **Green**
  - `SKU` **CL-02**, `Product` **Jeans**, `Color` **Blue**

    - `SKU` **AC-01**, `Product` **Hat**, `Color` **Blue**
- The `Product Category` table has two rows:
    - `SKU` **CL-01**, `Category` **Clothing**
    - `SKU` **AC-01**, `Category` **Accessories**

Notice that the `Product Category` table doesn't include a row for the product SKU **CL-02**. We'll discuss the consequences of this missing row later in this article.

In the **Data** pane, report authors find product-related fields in two tables: `Product` and `Product Category`. Let's see what happens when fields from both tables are added to a table visual. In this example, the `SKU` column is sourced from the `Product` table.



Notice that the `Category` value for product SKU **CL-02** is BLANK. That's because there's no corresponding row in the `Product Category` table for this product.

Recommendations

When possible, we recommend you avoid creating one-to-one model relationships when row data spans across model tables. That's because this design can:

- Contribute to **Data** pane clutter, listing more tables than necessary.
- Make it difficult for report authors to find related fields, because they're distributed across multiple tables.

- Limit the ability to create hierarchies, as their levels must be based on columns from the same table.
- Produce unexpected results when there isn't a complete match of rows between the tables.

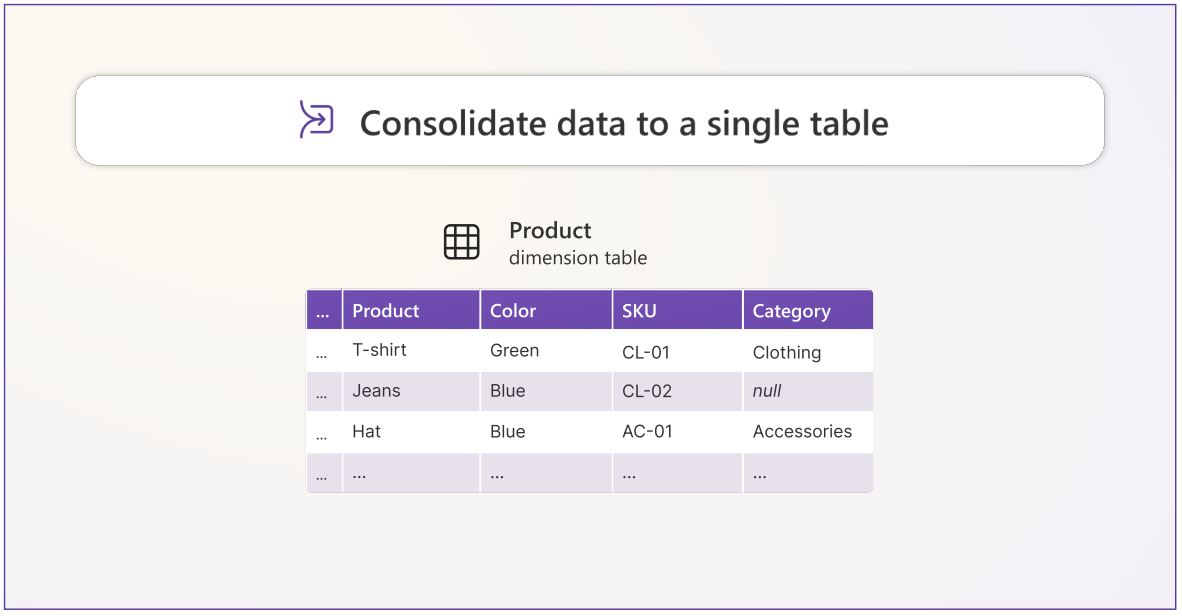
Specific recommendations differ depending on whether the one-to-one relationship is *intra source group* or *cross source group*. For more information about relationship evaluation, see Model relationships in Power BI Desktop.

Intra source group one-to-one relationship

When a one-to-one *intra source group* relationship exists between tables, we recommend consolidating the data into a single model table. You can do that by merging the Power Query queries.

The following steps present a methodology to consolidate and model the one-to-one related data.

1. **Merge queries**: When combining the two queries, give consideration to the completeness of data in each query. If one query contains a complete set of rows (like a master list), merge the other query with it. Set the merge transformation to use a *left outer join*, which is the default join type. This join type ensures you'll keep all rows of the first query, and supplement them with any matching rows of the second query. Expand all required columns of the second query into the first query.

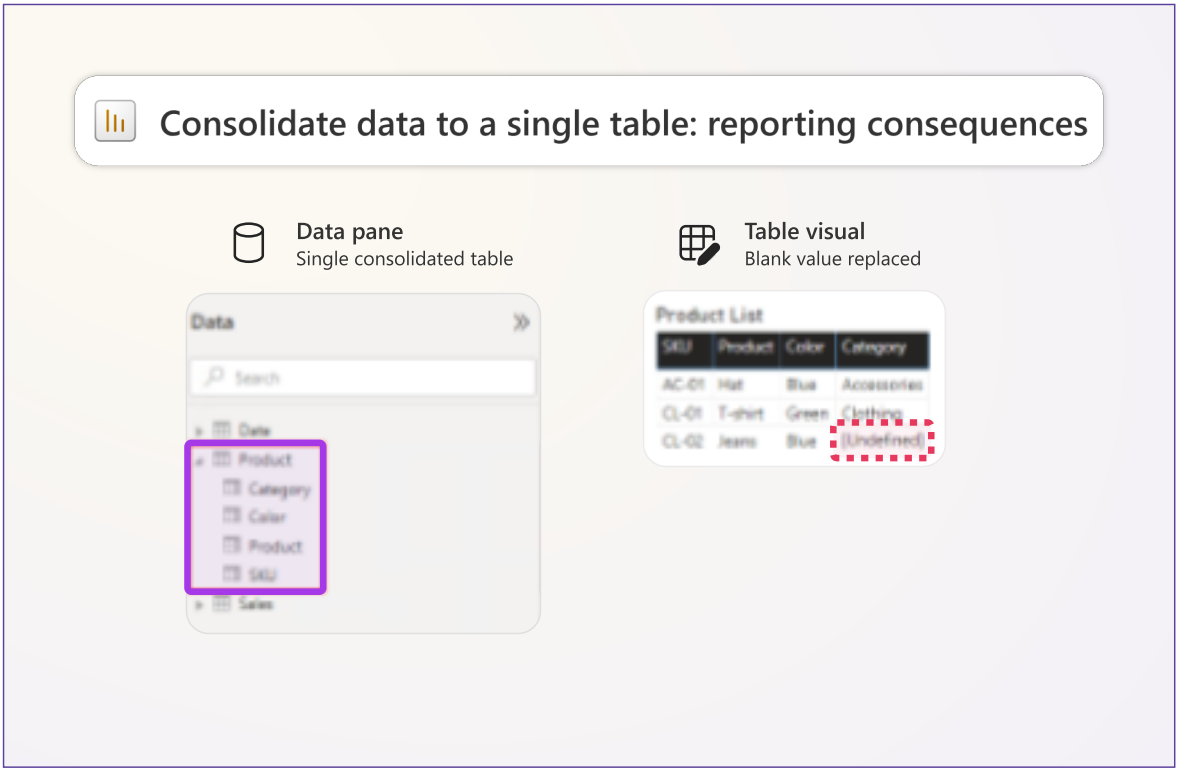


2. **Disable query load**: Be sure to disable the load of the second query. This way, it won't load its result as a model table. This configuration reduces the data model storage size, and helps to unclutter the **Data** pane.

In our example, report authors now find a single table named `Product` in the **Data** pane. It contains all product-related fields.

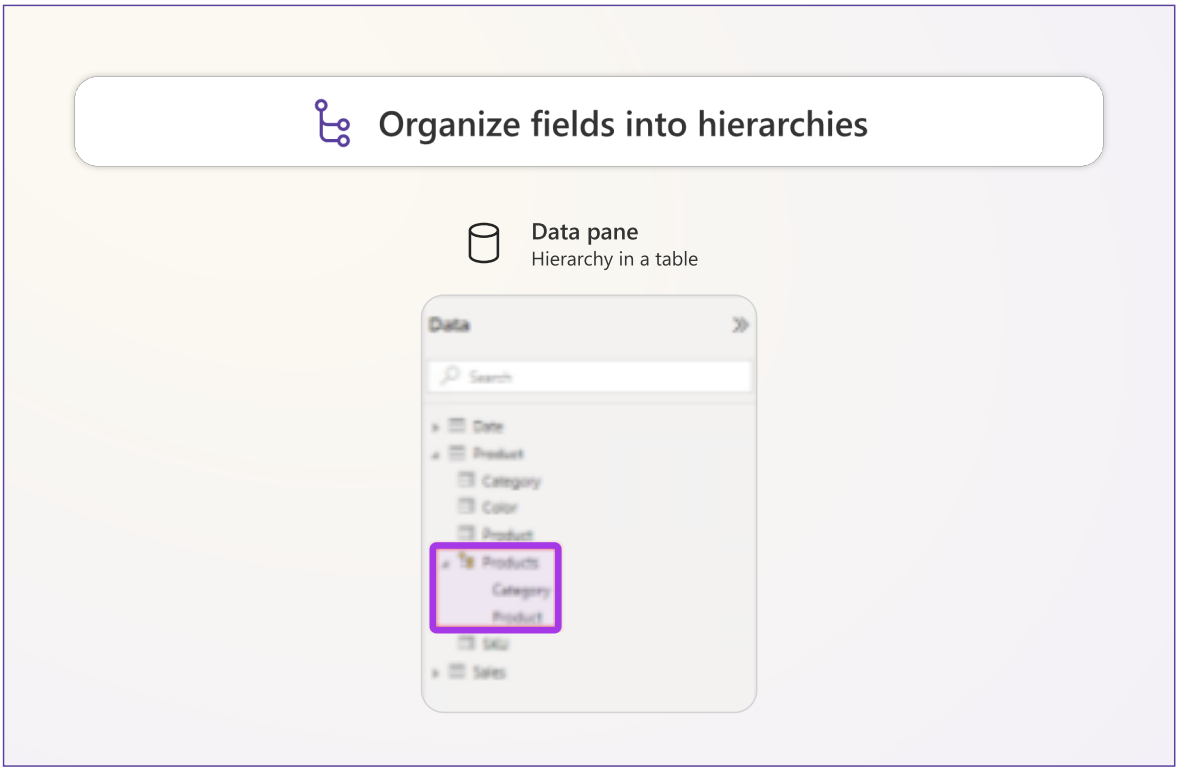
3. **Replace missing values**: If the second query has unmatched rows, null values appear in the columns introduced from it. When appropriate, consider replacing null values with a token value. Replacing missing values is especially important when report authors filter or group by the column values, as BLANKs could appear in report visuals.

   In the following image, notice that the category for product SKU **CL-02** now reads **[Undefined]**. In the query, null categories were replaced with this token text value.



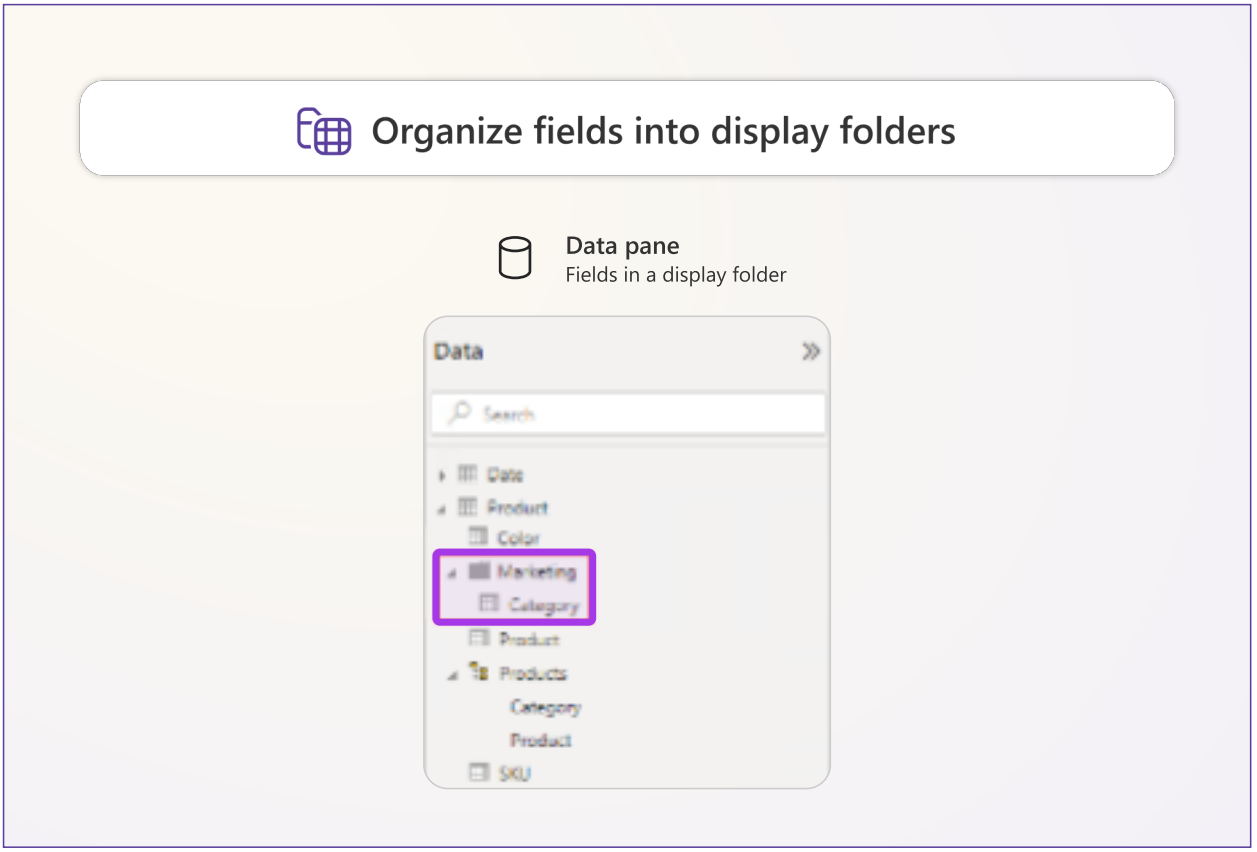
4. **Create hierarchies**: If relationships exist *between the columns* of the now-consolidated table, consider creating hierarchies. This way, report authors will quickly identify opportunities for report visual drilling.

   In our example, report authors now can use a hierarchy that has two levels: `Category` and `Product`.



If you like how separate tables help organize your fields, we still recommend consolidating into a single table. You can still organize your fields, but by using *display folders* instead.

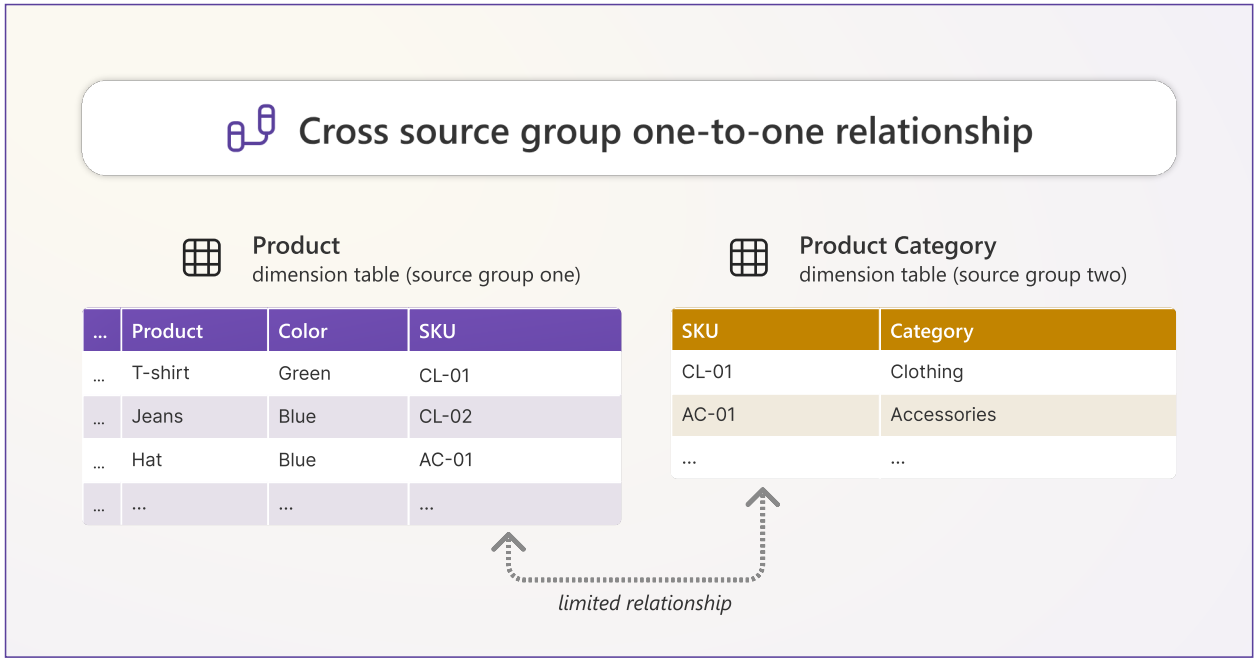
In our example, report authors can find the `Category` field within the `Marketing` display folder.



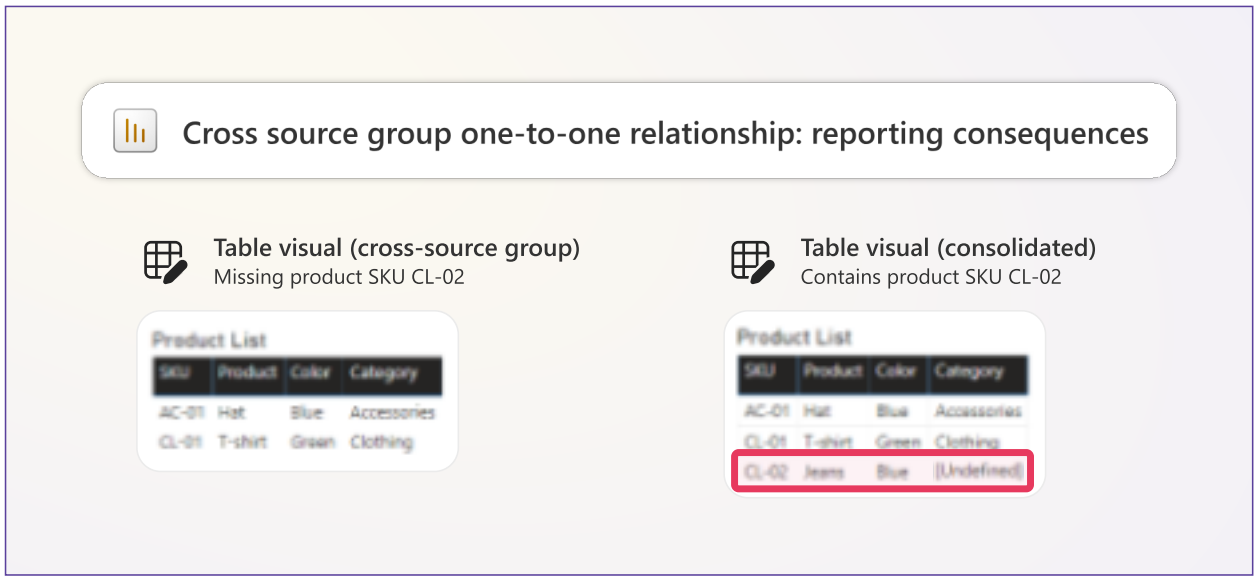
Should you still decide to define one-to-one intra source group relationships in your model, when possible, ensure there are matching rows in the related tables. As a one-to-one intra source group relationship is evaluated as a regular relationship, data integrity issues could surface in your report visuals as BLANKs. (You can see an example of a BLANK grouping in the first table visual presented in this article.)

Cross source group one-to-one relationship

When a one-to-one *cross source group* relationship exists between tables, there's no alternative model design—unless you pre-consolidate the data in your data source. Power BI will evaluate the one-to-one model relationship as a limited relationship. Therefore, take care to ensure there are matching rows in the related tables, as unmatched rows are eliminated from query results.



Let's see what happens when fields from both tables are added to a table visual, and a limited relationship exists between the tables.



The first table visual, which uses a cross-source group relationship, displays two rows only. Product SKU **CL-02** is missing because there's no matching row in the `Product Category` table. The second table visual, based on a single consolidated table in the model, displays three rows.

Related content

For more information related to this article, check out the following resources:

- Model relationships in Power BI Desktop
- Understand star schema and the importance for Power BI
- Relationship troubleshooting guidance
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Many-to-many relationship guidance

Article • 12/27/2024

This article targets you as a data modeler who works with Power BI Desktop. It describes three different many-to-many modeling scenarios. It also provides you with guidance on how to successfully design for them in your models.

> **Note**
>
> An introduction to model relationships is not covered in this article. If you're not completely familiar with relationships, their properties or how to configure them, we recommend that you first read the Model relationships in Power BI Desktop article.
>
> It's also important that you have an understanding of star schema design. For more information, see Understand star schema and the importance for Power BI.

There are three different many-to-many scenarios. They can occur when you're required to:

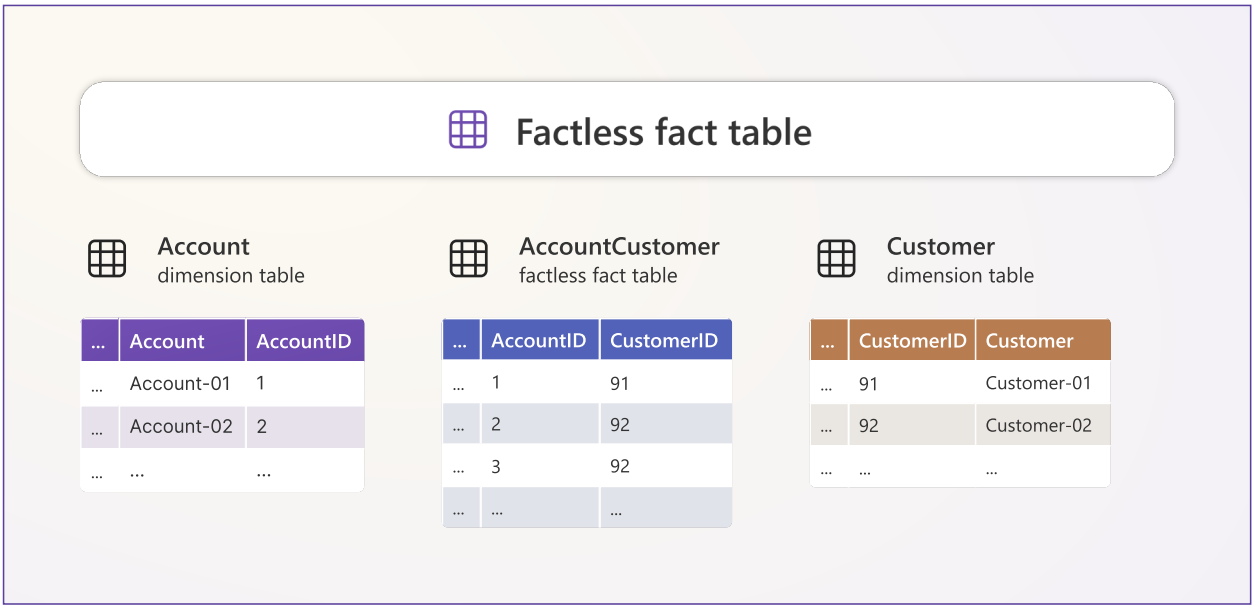
- Relate two dimension tables
- Relate two fact tables
- Relate higher grain fact tables, when the fact table stores rows at a higher grain than the dimension table rows

Relate many-to-many dimensions

The classic many-to-many scenario relates two entities, for example bank customers and bank accounts. Consider that customers can have multiple accounts, and accounts can have multiple customers. When an account has multiple customers, they're commonly called *joint account holders*.

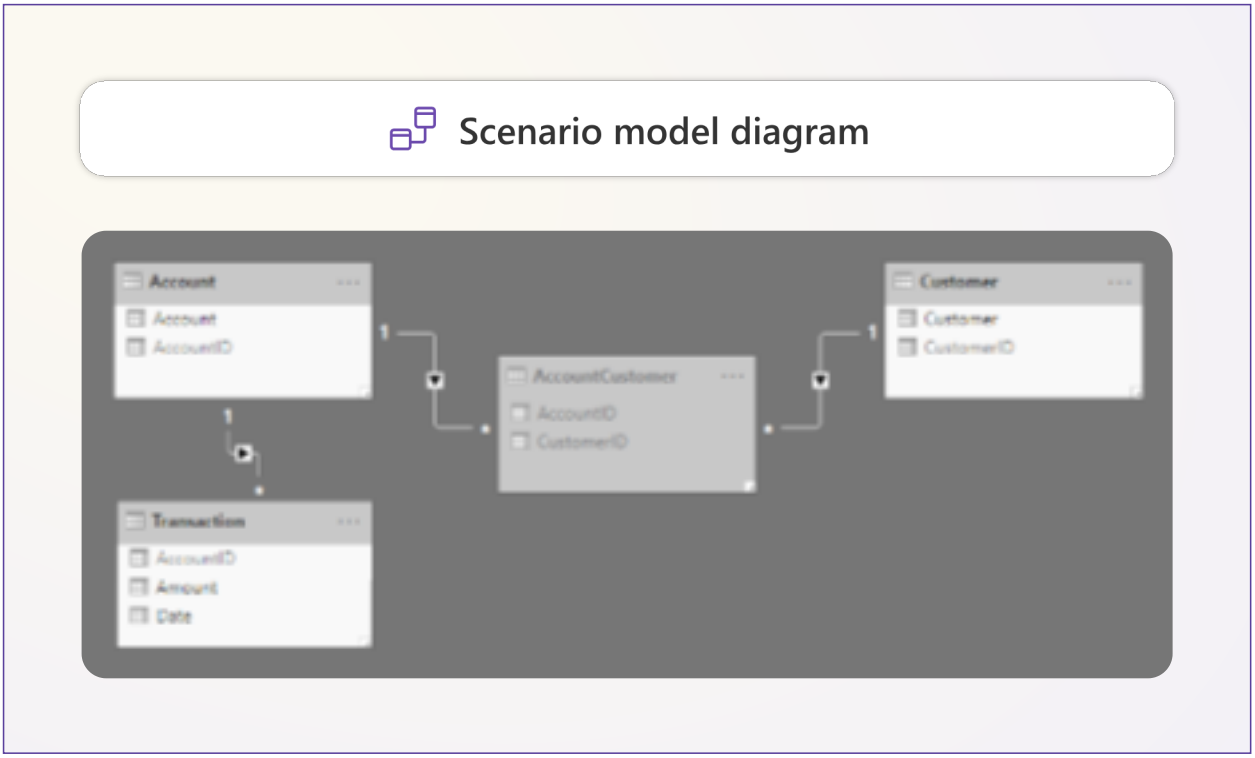
Modeling these entities is straight forward. One dimension table stores accounts, and another dimension table stores customers. As is characteristic of dimension tables, there's a unique identifier (ID) column in each table. To model the relationship between the two tables, a third table is required. This table is commonly referred to as a *bridging table*. In this example, it's purpose is to store one row for each customer-account association. Interestingly, when this table only contains identifier columns, it's called a factless fact table.

Here's a simplistic diagram of the three model tables.

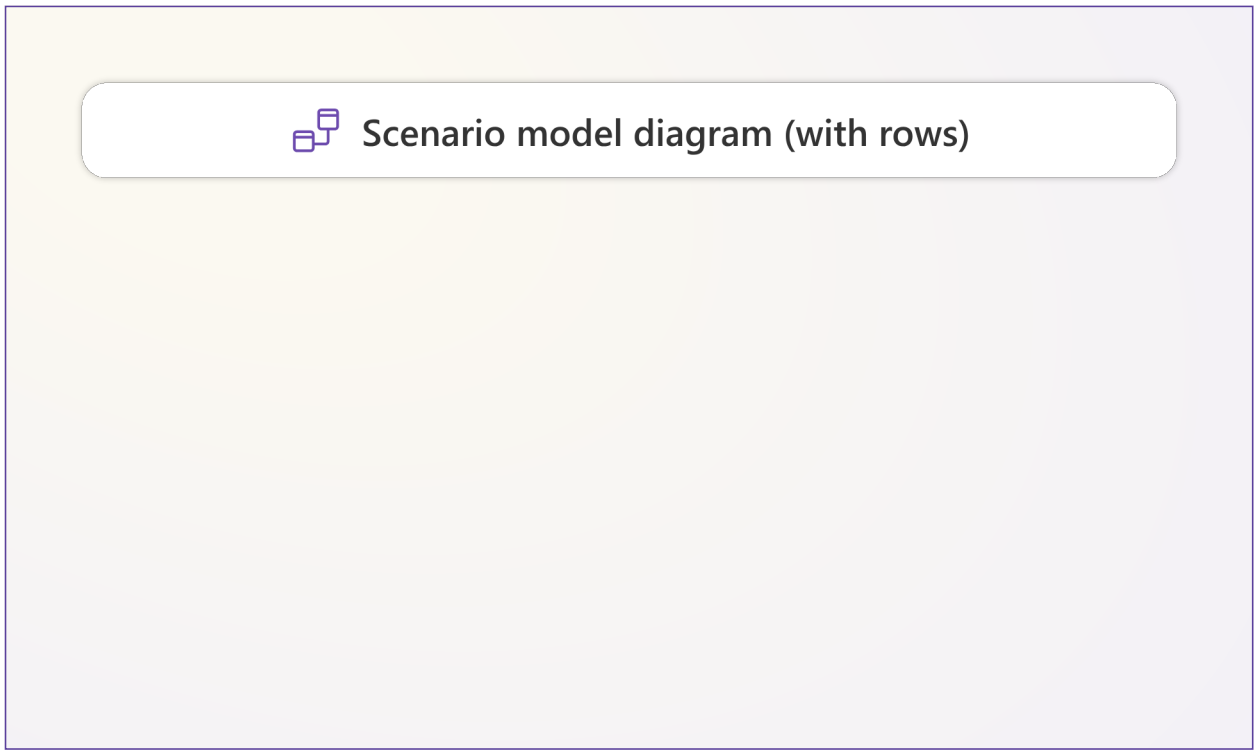


The first table is named `Account`, and it contains two columns: `AccountID` and `Account`. The second table is named `AccountCustomer`, and it contains two columns: `AccountID` and `CustomerID`. The third table is named `Customer`, and it contains two columns: `CustomerID` and `Customer`. Relationships don't exist between any of the tables.

Two one-to-many relationships are added to relate the tables. Here's an updated model diagram of the related tables. A fact table named `Transaction` has been added. It records account transactions. The bridging table and all identifier columns have been hidden.



To help describe how the relationship filter propagation works, the model diagram has been modified to reveal the table rows.

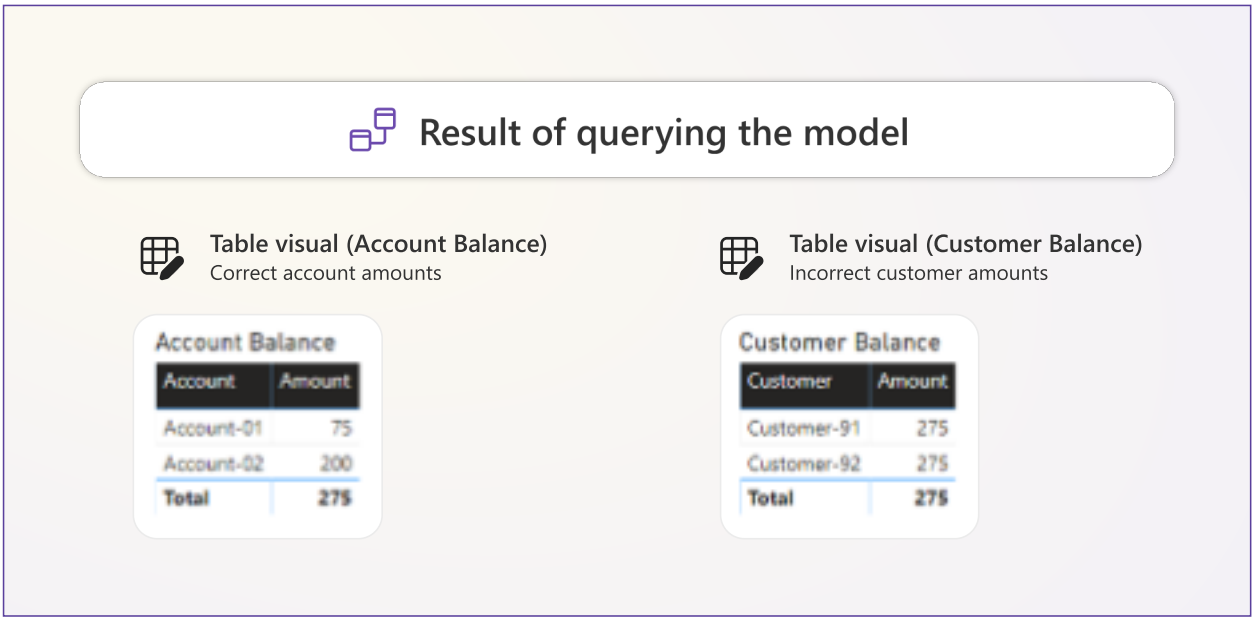


The row details for the four tables are presented in the following bulleted list:

- The `Account` table has two rows:
  - `AccountID` **1** is for **Account-01**
  - `AccountID` **2** is for **Account-02**
- The `Customer` table has two rows:
  - `CustomerID` **91** is for **Customer-91**
  - `CustomerID` **92** is for **Customer-92**
- The `AccountCustomer` table has three rows:
  - `AccountID` **1** is associated with `CustomerID` **91**
  - `AccountID` **1** is associated with `CustomerID` **92**
  - `AccountID` **2** is associated with `CustomerID` **92**
- The `Transaction` table has three rows:
  - `Date` **January 1 2019**, `AccountID` **1**, `Amount` **100**
  - `Date` **February 2 2019**, `AccountID` **2**, `Amount` **200**
  - `Date` **March 3 2019**, `AccountID` **1**, `Amount` **-25**

Let's see what happens when the model is queried.

In the following image, there are two table visuals that summarize the `Amount` column of the `Transaction` table. The first visual groups by account, and so the sum of the `Amount` columns represents the *account balance*. The second visual groups by customer, and so the sum of the `Amount` columns represents the *customer balance*.



The first table visual (Account Balance) has two columns: `Account` and `Amount`. It displays the following result:

- **Account-01** balance amount is **75**.
- **Account-02** balance amount is **200**.
- The total is **275**.

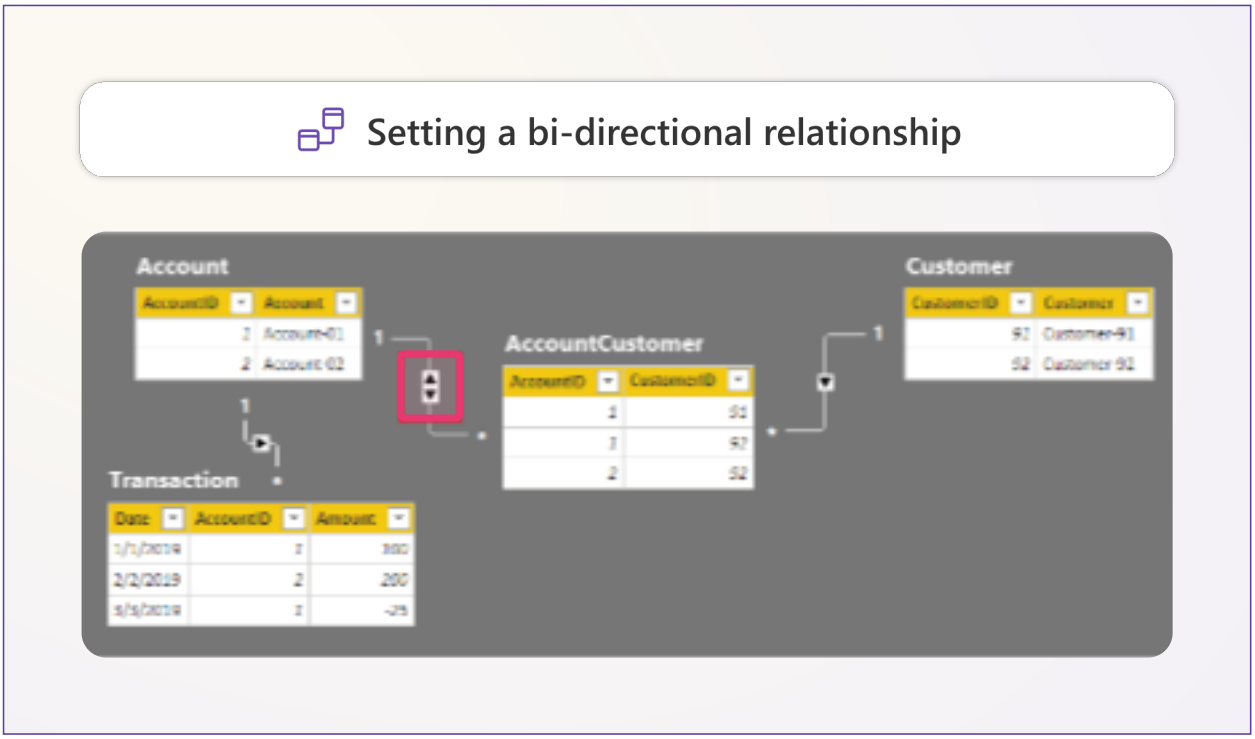
The second table visual (Customer Balance) has two columns: `Customer` and `Amount`. It displays the following result:

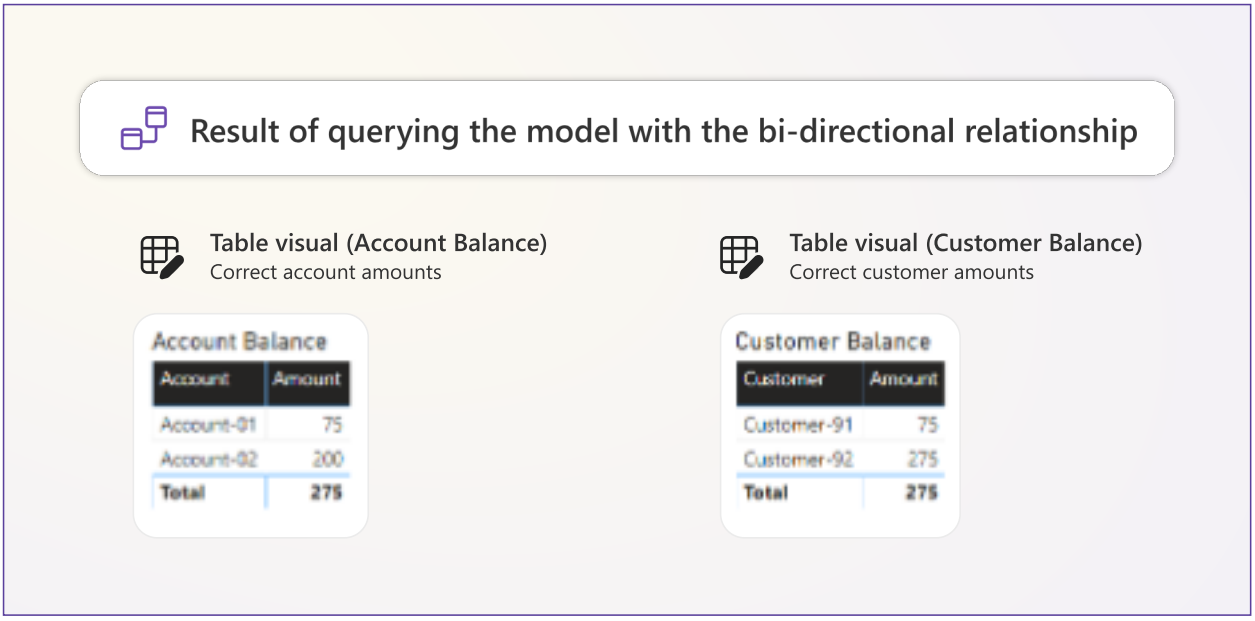
- **Customer-91** balance amount is **275**.
- **Customer-92** balance amount is **275**.
- The total is **275**.

A quick glance at the table rows and the Account Balance visual reveals that the result is correct, for each account and the total amount. That's because each account grouping results in a filter propagation to the `Transaction` table for that account.

However, something doesn't appear correct with the Customer Balance visual. Each customer in this visual has the same balance as the total balance. This result could only be correct if every customer was a joint account holder of every account. That's not the case in this example. There's an issue, and it's related to filter propagation. Filters aren't flowing all the way to the `Transaction` table.

If you follow the relationship filter directions from the `Customer` table to the `Transaction` table, you can determine that the relationship between the `Account` and `AccountCustomer` tables is propagating in the wrong direction. The filter direction for this relationship must be set to `Both`.





As expected, there has been no change to the Account Balance visual.

The Customer Balance visual, however, now displays the following result:

- **Customer-91** balance amount is **75**.
- **Customer-92** balance amount is **275**.
- The total is **275**.

The Customer Balance visual now displays a correct result. Follow the filter directions for yourself, and see how the customer balances were calculated. Also, understand that the visual total means *all customers*.

Someone unfamiliar with the model relationships could conclude that the result is incorrect. They might ask: *Why isn't the total balance for* `Customer-91` *and* `Customer-92`

equal to 350 (75 + 275)?

The answer to their question lies in understanding the many-to-many relationship. Each customer balance can represent the addition of multiple account balances, and so the customer balances are *non-additive*.

Relate many-to-many dimensions guidance

When you have a many-to-many relationship between dimension tables, follow this guidance:

- Add each many-to-many related entity as a model table, ensuring it has an ID column.
- Add a bridging table to store associated entities.
- Create one-to-many relationships between the three tables.
- Set *one* bi-directional relationship to allow filter propagation to continue to the fact table.
- When it isn't appropriate to have missing ID values, disable the `Is Nullable` property—data refresh will fail when missing values are sourced.
- Hide the bridging table (unless it contains other columns or measures required for reporting).
- Hide any ID columns that aren't suitable for reporting (for example, when the columns store surrogate key values).
- If it makes sense to leave an ID column visible, ensure that it's on the "one" side of the relationship—always hide the "many" side column. That's because filters applied to the "one" slide result in better filter performance.
- To avoid confusion or misinterpretation, communicate explanations to your report users—you can add descriptions with text boxes or visual header tooltips.

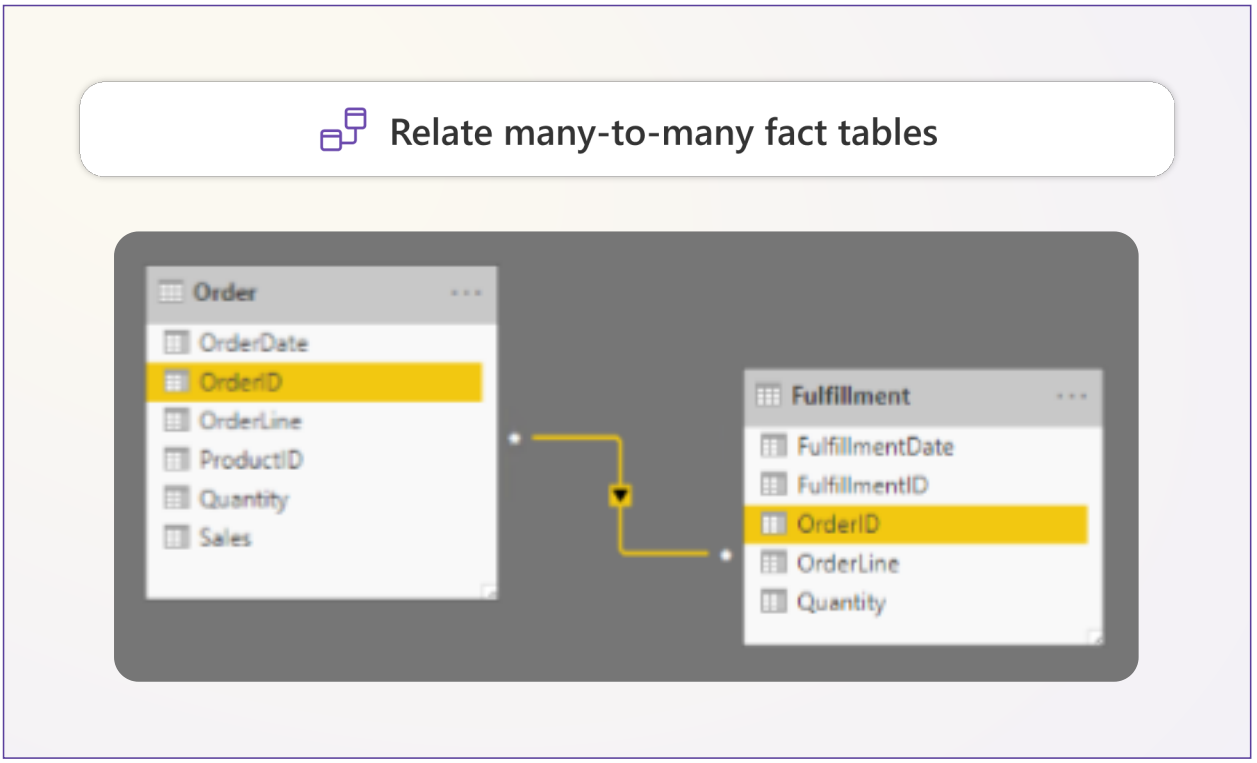
We don't recommend that you relate many-to-many dimension tables directly. This design approach requires setting up a relationship with a many-to-many cardinality. Conceptually it can be achieved, yet it implies that the related columns might contain duplicate values. It's a well-accepted design practice, however, that dimension tables have an ID column. Dimension tables should always use the ID column as the "one" side of a relationship.

Relate many-to-many facts

A different many-to-many scenario type involves relating two fact tables. Two fact tables can be related directly. This design technique can be useful for quick and simple data

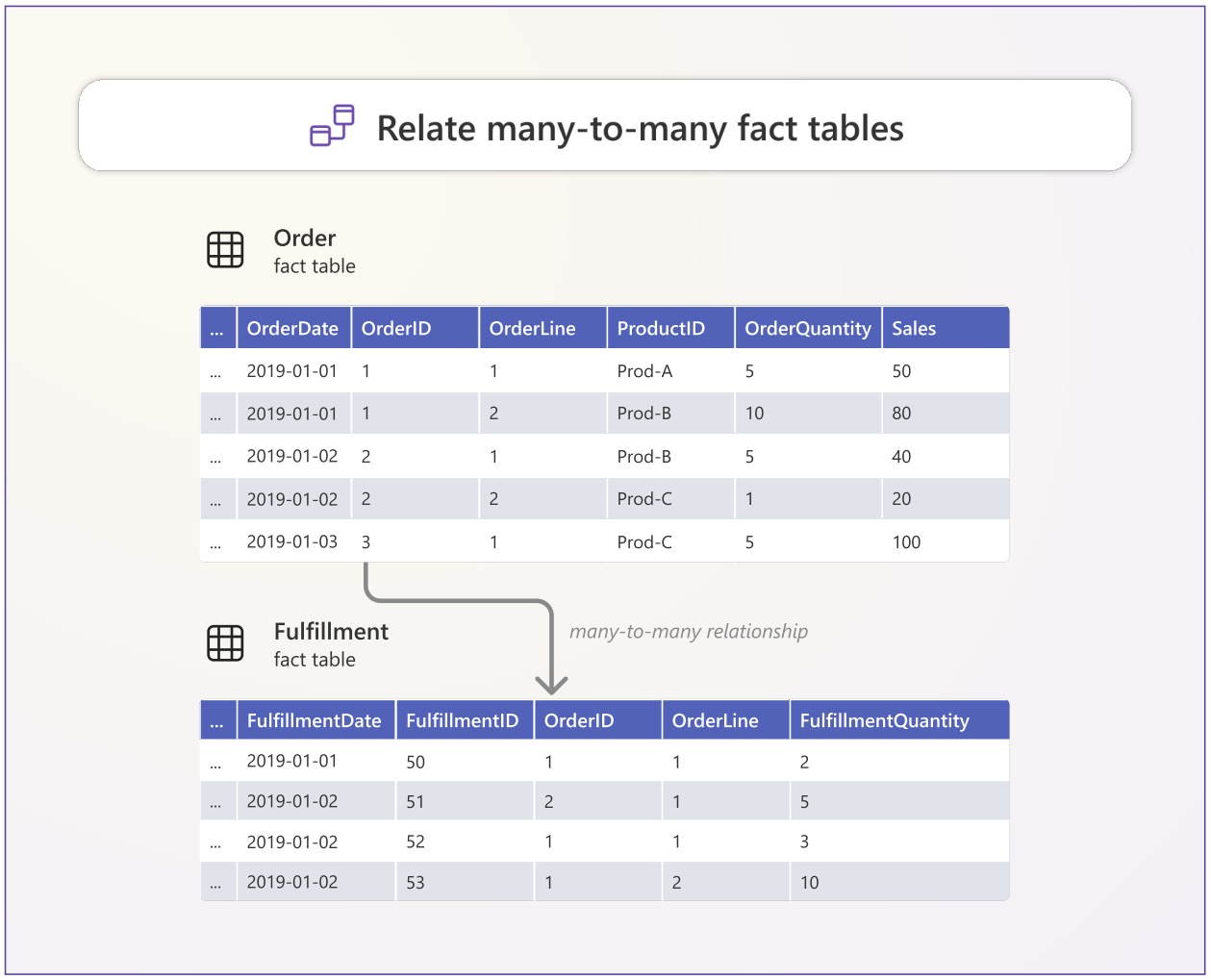
exploration. However, and to be clear, we generally don't recommend this design approach. We'll explain why later in this section.

Let's consider an example that involves two fact tables: `Order` and `Fulfillment`. The `Order` table contains one row per order line, and the `Fulfillment` table can contains zero or more rows per order line. Rows in the `Order` table represent sales orders. Rows in the `Fulfillment` table represent order items that have been shipped. A many-to-many relationship relates the `OrderID` columns in each table, with filter propagation only from the `Order` table (meaning the `Order` table filters the `Fulfillment` table).



The relationship cardinality is set to `Many-to-many` to support storing duplicate `OrderID` column values in both tables. In the `Order` table, duplicate ID values can exist because an order can have multiple lines. In the `Fulfillment` table, duplicate ID values can exist because orders can have multiple lines, and order lines can be fulfilled by many shipments.

Let's now take a look at the table rows. In the `Fulfillment` table, notice that order lines can be fulfilled by multiple shipments. (The absence of an order line means the order is yet to be fulfilled.)

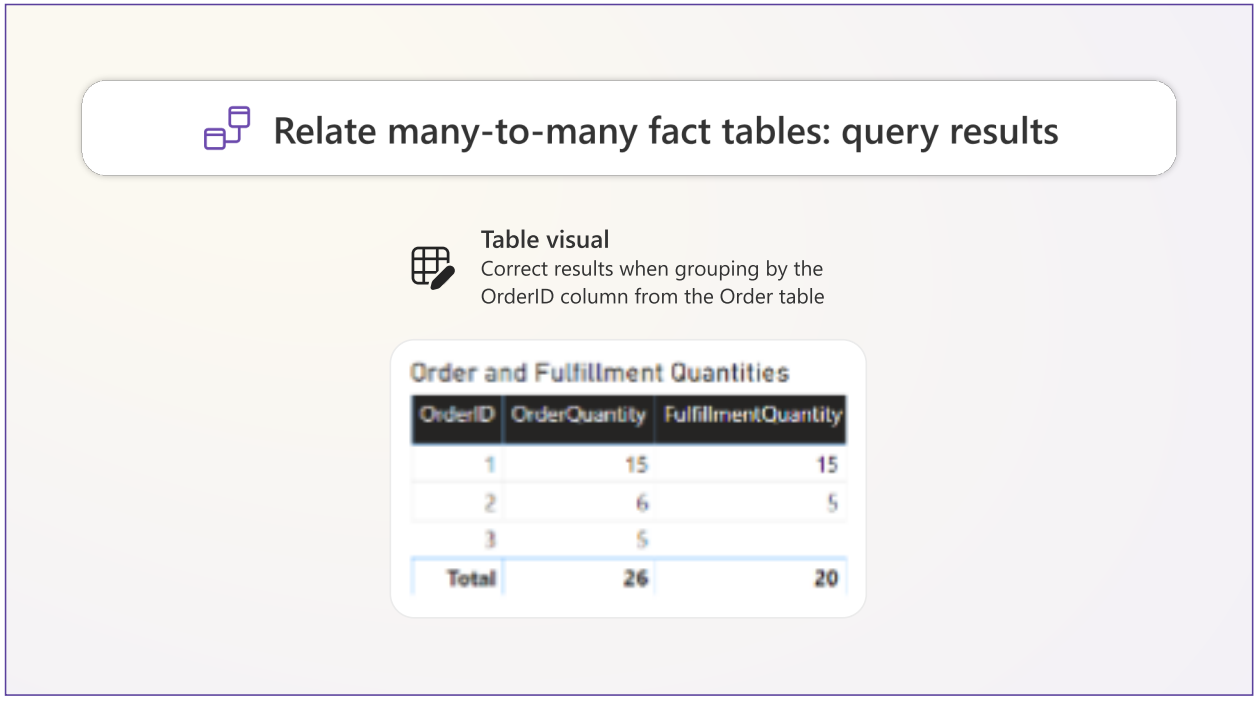


The row details for the two tables are described in the following bulleted list:

- The `Order` table has five rows:
  - `OrderDate` **January 1 2019**, `OrderID` **1**, `OrderLine` **1**, `ProductID` **Prod-A**, `OrderQuantity` **5**, `Sales` **50**
  - `OrderDate` **January 1 2019**, `OrderID` **1**, `OrderLine` **2**, `ProductID` **Prod-B**, `OrderQuantity` **10**, `Sales` **80**
  - `OrderDate` **February 2 2019**, `OrderID` **2**, `OrderLine` **1**, `ProductID` **Prod-B**, `OrderQuantity` **5**, `Sales` **40**
  - `OrderDate` **February 2 2019**, `OrderID` **2**, `OrderLine` **2**, `ProductID` **Prod-C**, `OrderQuantity` **1**, `Sales` **20**
  - `OrderDate` **March 3 2019**, `OrderID` **3**, `OrderLine` **1**, `ProductID` **Prod-C**, `OrderQuantity` **5**, `Sales` **100**
- The `Fulfillment` table has four rows:
  - `FulfillmentDate` **January 1 2019**, `FulfillmentID` **50**, `OrderID` **1**, `OrderLine` **1**, `FulfillmentQuantity` **2**
  - `FulfillmentDate` **February 2 2019**, `FulfillmentID` **51**, `OrderID` **2**, `OrderLine` **1**, `FulfillmentQuantity` **5**

- `FulfillmentDate` February 2 2019, `FulfillmentID` 52, `OrderID` 1, `OrderLine` 1, `FulfillmentQuantity` 3
- `FulfillmentDate` January 1 2019, `FulfillmentID` 53, `OrderID` 1, `OrderLine` 2, `FulfillmentQuantity` 10

Let's see what happens when the model is queried. Here's a table visual comparing order and fulfillment quantities by the `Order` table `OrderID` column.



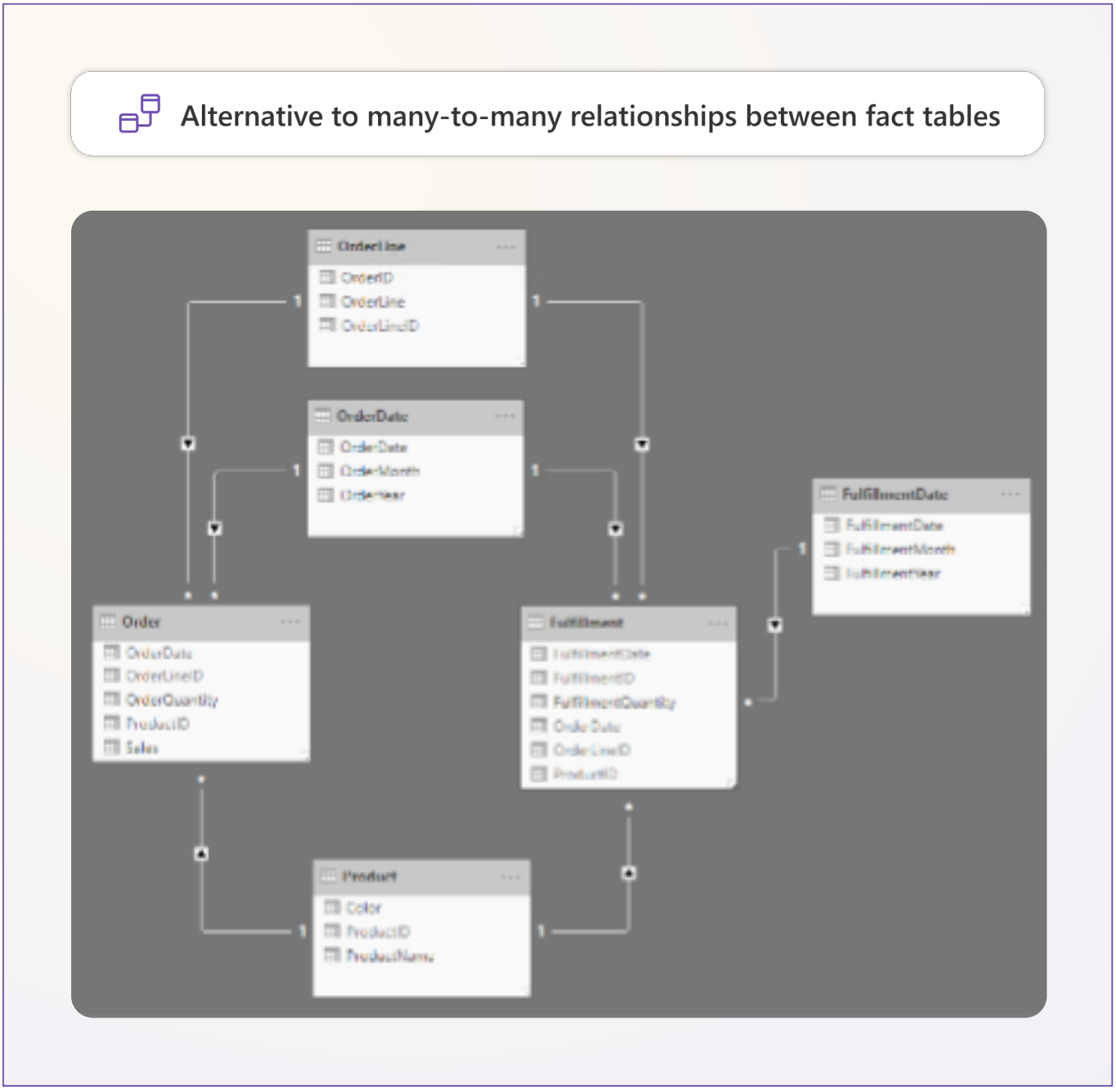
The visual presents an accurate result. However, the usefulness of the model is limited because you can only filter or group by the `Order` table `OrderID` column.

Relate many-to-many facts guidance

Generally, we don't recommend that you relate two fact tables directly by using many-to-many cardinality. The main reason is because the model won't provide flexibility in the ways your report visuals filter or group. In the example, it's only possible for visuals to filter or group by the `Order` table `OrderID` column. Another reason relates to the quality of your data. If your data has integrity issues, it's possible some rows might be omitted during querying due to the nature many-to-man cardinality and limited relationships.

Instead of relating fact tables directly, we recommend that you implement a star schema design. That means you add dimension tables. These dimension tables then relate to the fact tables by using one-to-many relationships. This design approach is robust as it efficiently delivers flexible reporting options. It lets you filter or group by using any of the dimension table columns, and summarize columns of any related fact table.

Let's consider a better solution.



Notice the following design changes:

- The model now has four extra tables: `OrderLine`, `OrderDate`, `Product`, and `FulfillmentDate`.
- The four extra tables are all dimension tables where one-to-many relationships relate them to the fact tables.
- The `OrderLine` table contains the `OrderLineID` column, which stores the `OrderID` value multiplied by 100, plus the `OrderLine` column value—an ID for each order line.
- The `Order` and `Fulfillment` tables now each contain an `OrderLineID` column, and they no longer contain the `OrderID` and `OrderLine` columns.
- The `Fulfillment` table now contains `OrderDate` and `ProductID` columns.
- The `FulfillmentDate` table has a relationship only to the `Fulfillment` table.
- All ID columns are hidden.

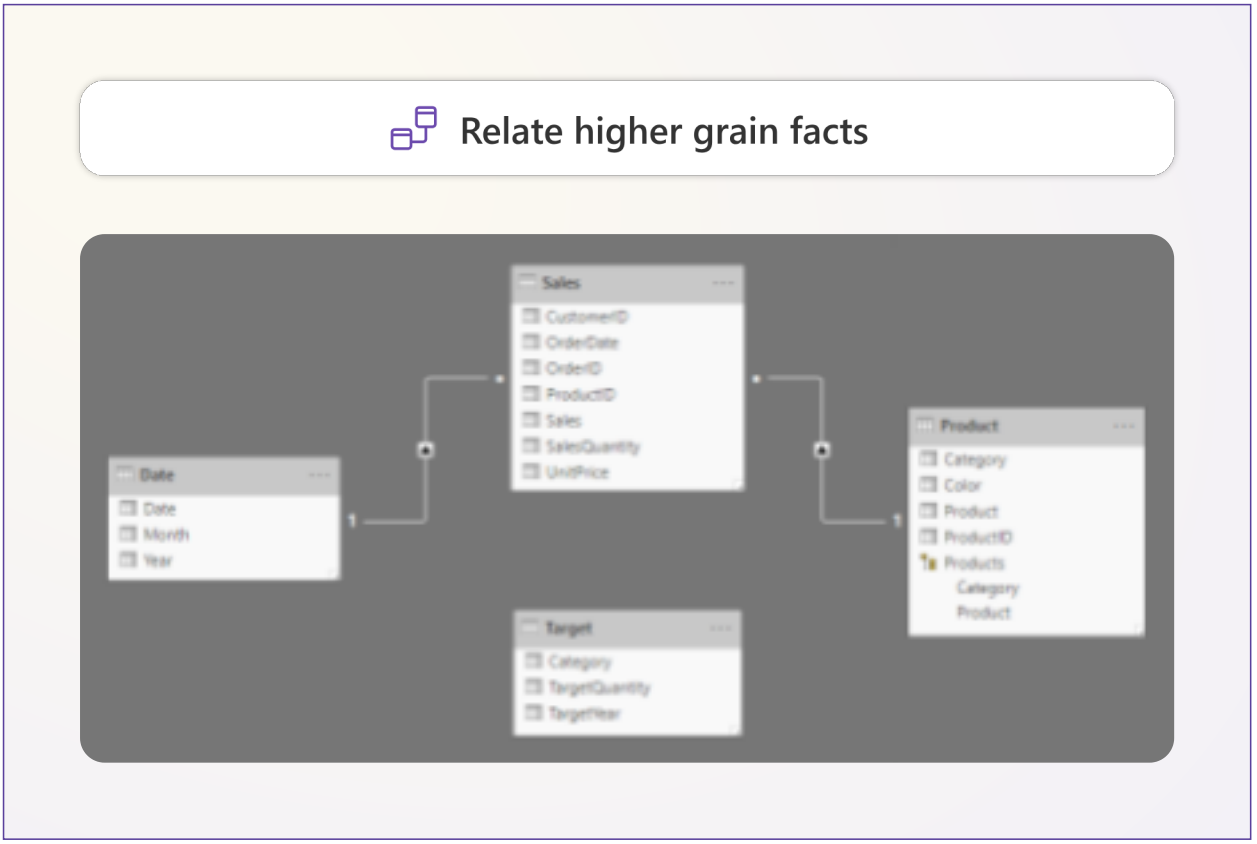
Taking the time to adopt a star schema design delivers the following benefits:

- Your report visuals can *filter or group* by any visible column from the dimension tables.
- Your report visuals can *summarize* any visible column from the fact tables.
- Filters applied to the `OrderLine`, `OrderDate`, or `Product` tables propagate to both fact tables.
- All relationships are one-to-many, and each relationship is a *regular relationship*. Data integrity issues won't be masked. For more information about relationship evaluation, see Model relationships in Power BI Desktop.

Relate higher grain facts

This many-to-many scenario is very different from the other two already described in this article.

Let's consider an example involving four tables: `Date`, `Sales`, `Product`, and `Target`. The `Date` and `Product` tables are dimension tables, and one-to-many relationships relate each to the `Sales` fact table. So far, it represents a good star schema design. The `Target` table, however, is yet to be related to the other tables.



The `Target` table contains three columns: `Category`, `TargetQuantity`, and `TargetYear`. The table rows reveal a granularity of year and product category. In other words, targets —used to measure sales performance—are set each year for each product category.

Because the `Target` table stores data at a higher level than the dimension tables, a one-to-many relationship can't be created. Well, it's true for just one of the relationships. Let's explore how the `Target` table can be related to the dimension tables.

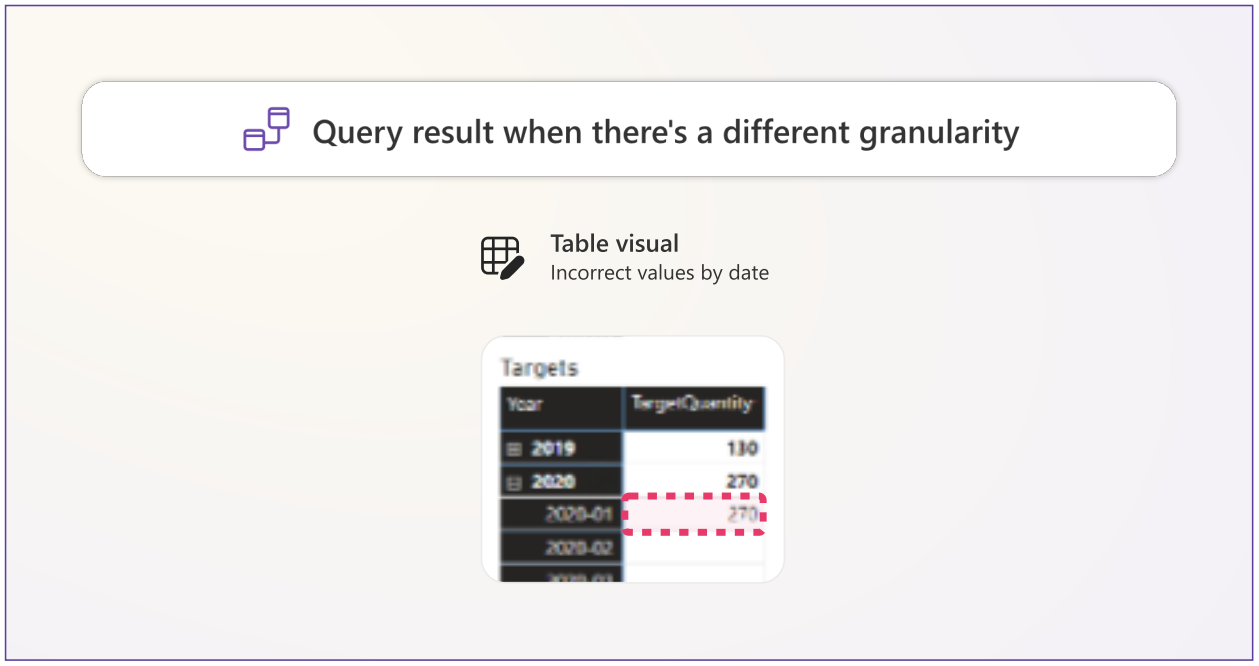
Relate higher grain time periods

A relationship between the `Date` and `Target` tables should be a one-to-many relationship. That's because the `TargetYear` column values are dates. In this example, each `TargetYear` column stores the first date of the target year.

> **Tip**
>
> When storing facts at a higher time granularity than day, set the column data type to **Date** (or **Whole number** if you're using date keys). In the column, store a value representing the first day of the time period. For example, a year period is recorded as January 1 of the year, and a month period is recorded as the first day of that month.

Care must be taken, however, to ensure that month or date level filters produce a meaningful result. Without any special calculation logic, report visuals might report that target dates are literally the first day of each year. All other days—and all months except January—will summarize the target quantity as BLANK.

The following matrix visual shows what happens when the report user drills down from a year into its months. The visual summarizes the `TargetQuantity` column. (The Show items with no data option has been enabled for the matrix rows.)



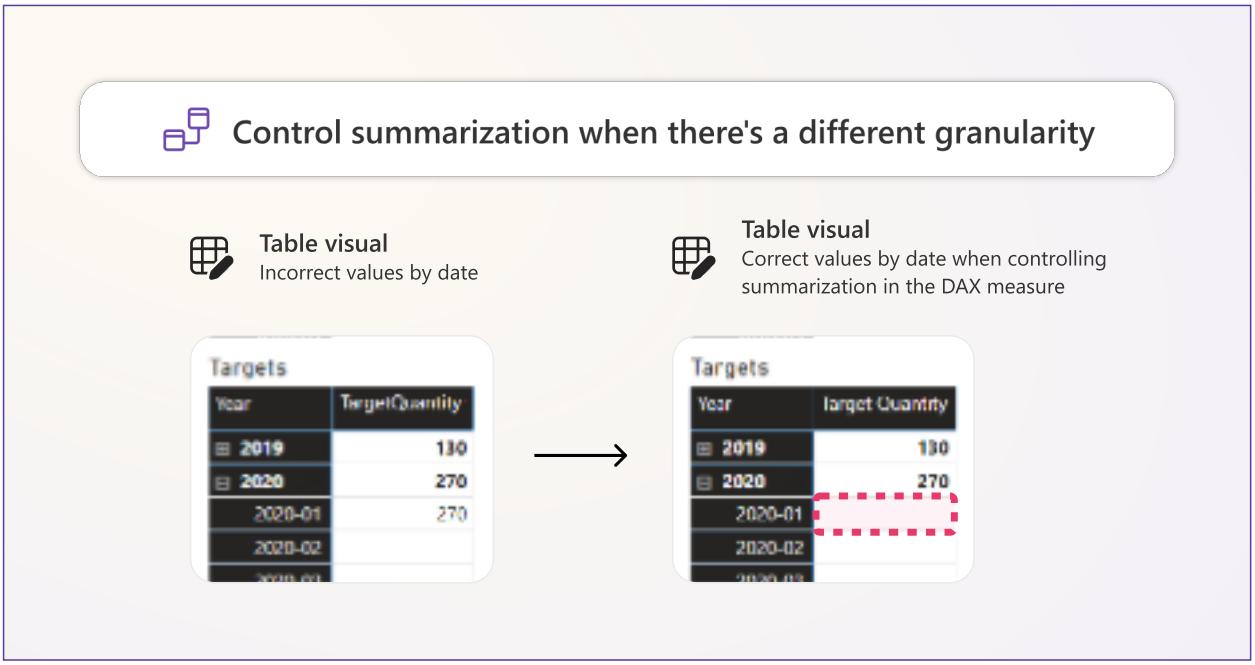
To avoid this behavior, we recommend that you control the summarization of your fact data by using measures. One way to control the summarization is to return BLANK when lower-level time periods are queried. Another way—defined with some sophisticated DAX—is to apportion values across lower-level time periods.

Consider the following measure definition that uses the ISFILTERED DAX function. It only returns a value when the `Date` and `Month` columns aren't filtered.

DAX

```
Target Quantity =
IF(
    NOT ISFILTERED('Date'[Date])
        && NOT ISFILTERED('Date'[Month]),
    SUM(Target[TargetQuantity])
)
```

The following matrix visual uses the `Target Quantity` measure. It shows that all monthly target quantities are BLANK.

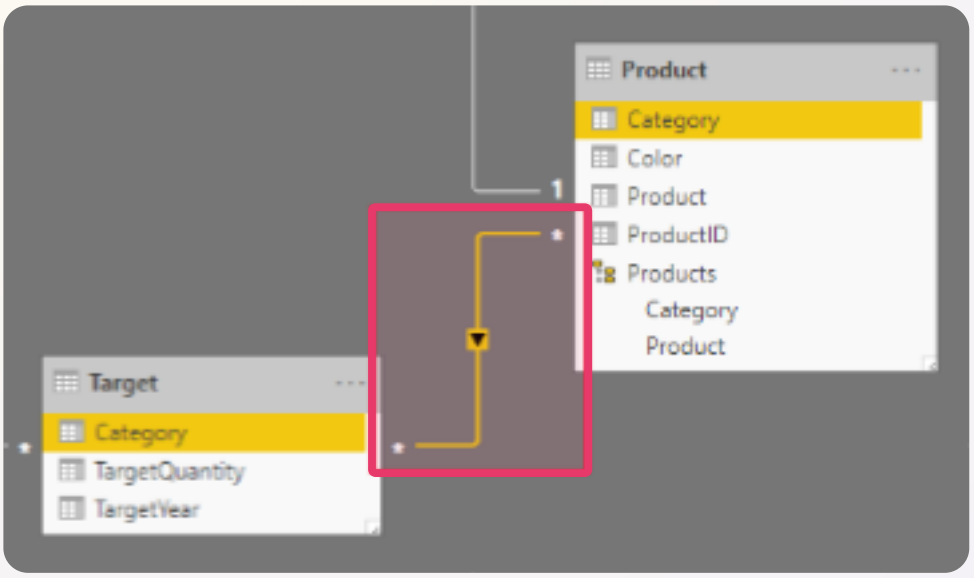


Relate higher grain (non-date)

A different design approach is required when relating a non-date column from a dimension table to a fact table (and it's at a higher grain than the dimension table).

The `Category` columns (from both the `Product` and `Target` tables) contains duplicate values. So, there's no "one" side for a one-to-many relationship. In this case, you'll need to create a many-to-many relationship. The relationship should propagate filters in a single direction, from the dimension table to the fact table.

Relate higher grain facts with many-to-many relationships



Let's now take a look at the table rows.

Relate higher grain facts with many-to-many relationships

Product
dimension table

...	ProductID	Product	Category	Color
...	1	T-shirt	Clothing	Green
...	2	Jeans	Clothing	Blue
...	3	Hat	Accessories	Blue

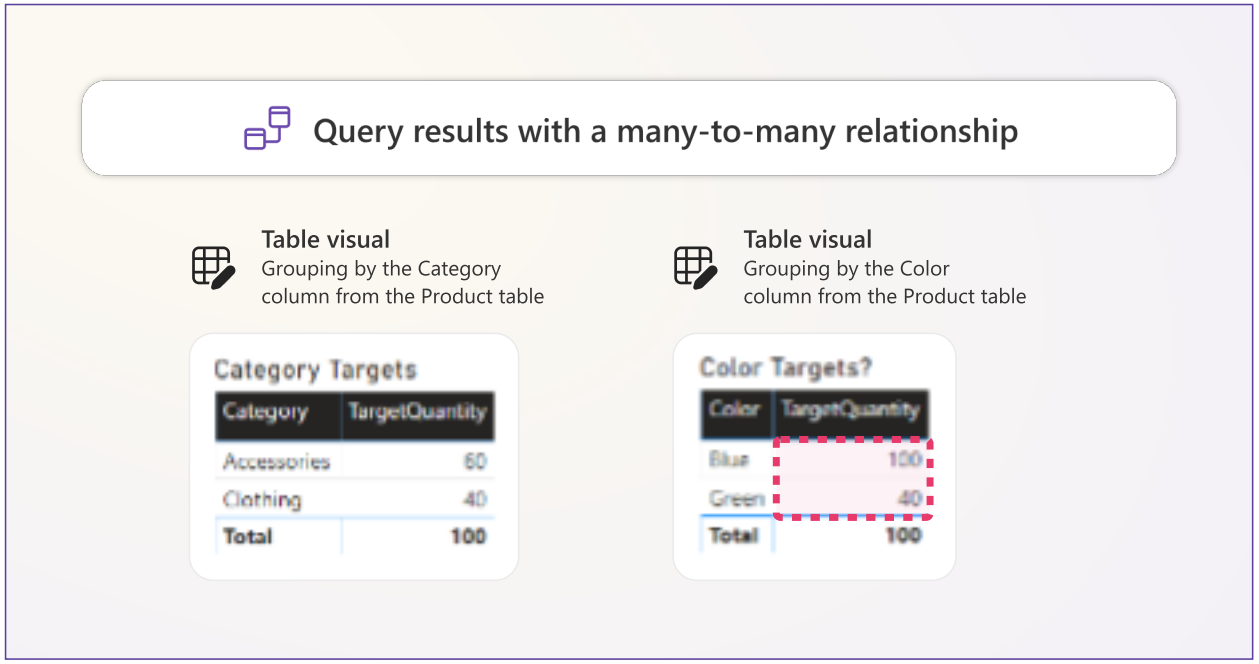
many-to-many relationship

Target
fact table

...	TargetYear	Category	FulfillmentQuantity
...	2019-01-01	Clothing	10
...	2019-01-01	Accessories	20
...	2019-01-01	Clothing	30
...	2019-01-02	Accessories	40
...	2019-01-02	Food	100
...	2019-01-02	Food	200

In the `Target` table, there are four rows: two rows for each target year (2019 and 2020), and two categories (Clothing and Accessories). In the `Product` table, there are three products. Two belong to the clothing category, and one belongs to the accessories category. One of the clothing colors is green, and the remaining two are blue.

A table visual grouping by the `Category` column from the `Product` table produces the following result. However, this visual produces the correct result. Let's now consider what happens when the `Color` column from the `Product` table is used to group target quantity.



The visual produces a misrepresentation of the data. What is happening here?

A filter on the `Color` column from the `Product` table results in two rows. One of the rows is for the Clothing category, and the other is for the Accessories category. These two category values are propagated as filters to the `Target` table. In other words, because the color blue is used by products from two categories, *those categories* are used to filter the targets.

To avoid this behavior, as described earlier, we recommend that you control the summarization of your fact data by using measures.

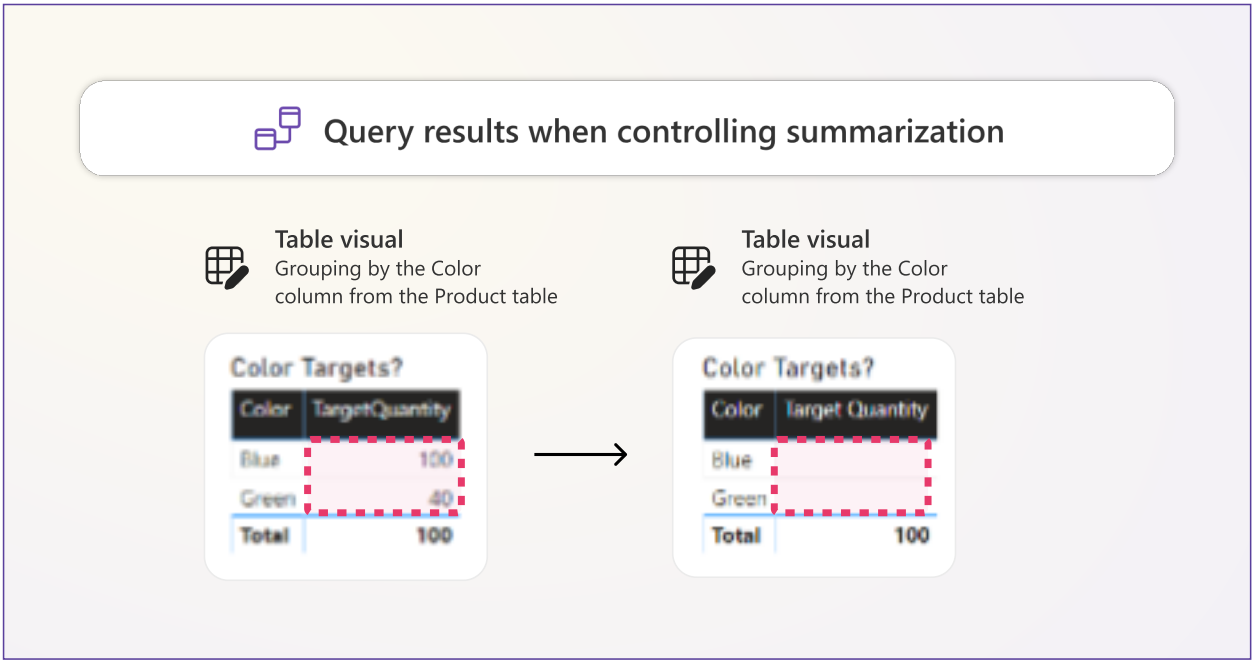
Consider the following measure definition. Notice that all `Product` table columns that are beneath the category level are tested for filters.

DAX

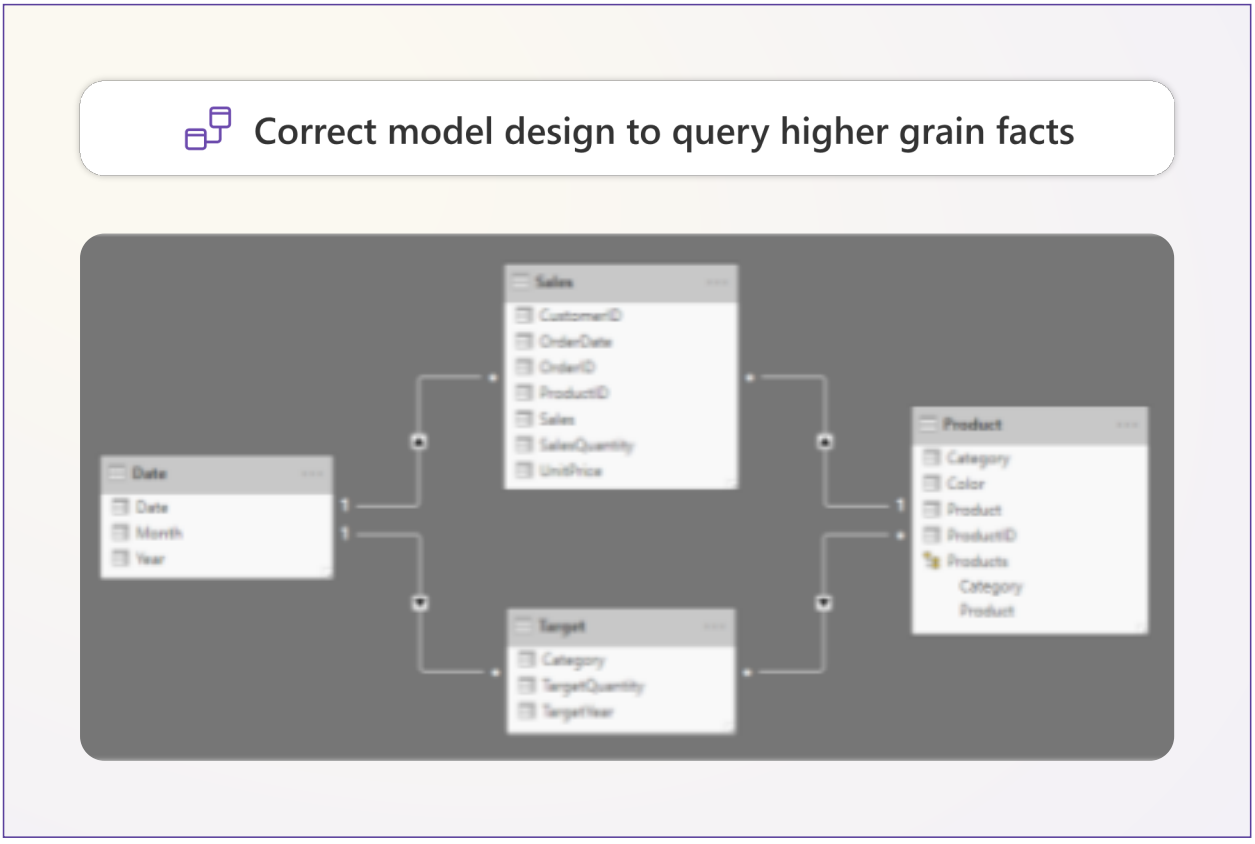
```
Target Quantity =
IF(
    NOT ISFILTERED('Product'[ProductID])
        && NOT ISFILTERED('Product'[Product])
```

```
        && NOT ISFILTERED('Product'[Color]),
    SUM(Target[TargetQuantity])
)
```

The following table visual uses the `Target Quantity` measure. It shows that all color target quantities are BLANK.



The final model design looks like the following.



Relate higher grain facts guidance

When you need to relate a dimension table to a fact table, and the fact table stores rows at a higher grain than the dimension table rows, follow this guidance:

- **For higher grain fact dates**
  - In the fact table, store the first date of the time period.
  - Create a one-to-many relationship between the date table and the fact table.
- **For other higher grain facts**
  - Create a many-to-many relationship between the dimension table and the fact table.
- **For both types**
  - Control summarization with measure logic—return BLANK when lower-level dimension columns are used to filter or group.
  - Hide summarizable fact table columns—that ensures only measures can be used to summarize the fact table.

Related content

For more information related to this article, check out the following resources:

- Model relationships in Power BI Desktop
- Understand star schema and the importance for Power BI
- Relationship troubleshooting guidance
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Active vs inactive relationship guidance

Article • 12/27/2024

This article targets you as a data modeler who works with Power BI Desktop. It provides you with guidance on when to create active or inactive model relationships. By default, active relationships propagate filters to other tables. Inactive relationship, however, only propagate filters when a DAX expression activates (uses) the relationship.

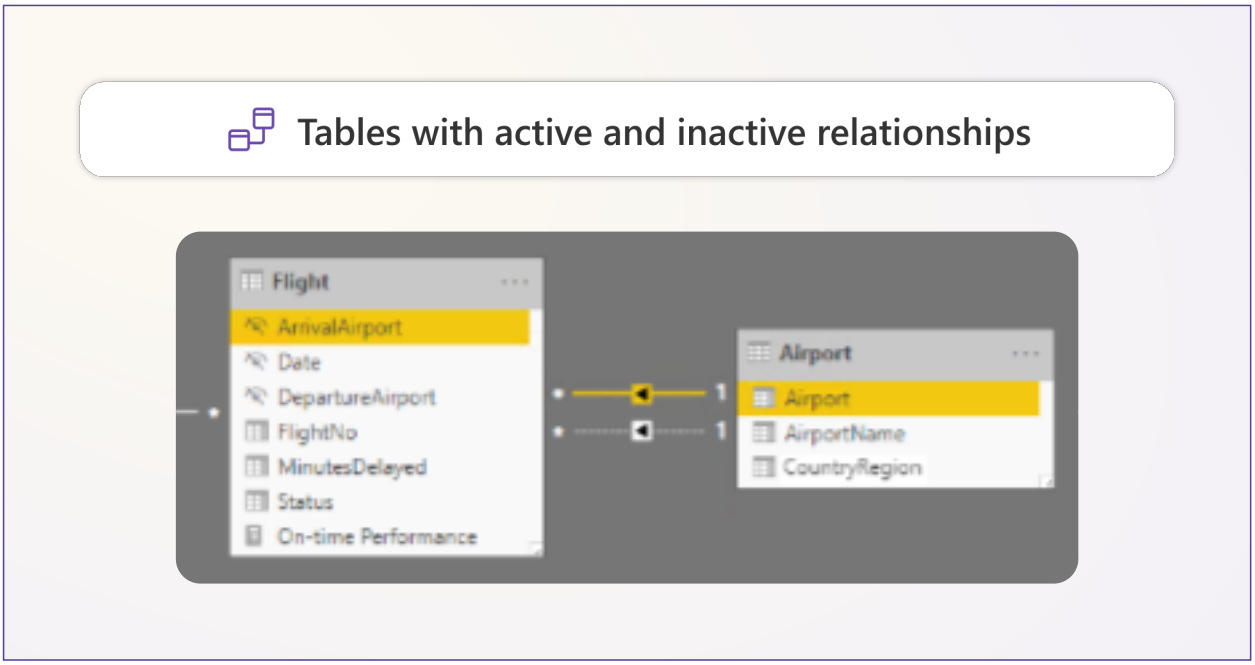
> **Note**
>
> An introduction to model relationships is not covered in this article. If you're not completely familiar with relationships, their properties or how to configure them, we recommend that you first read the Model relationships in Power BI Desktop article.
>
> It's also important that you have an understanding of star schema design. For more information, see Understand star schema and the importance for Power BI.

Active relationships

Generally, we recommend that you define active relationships whenever possible. They widen the scope and potential of how your model can be used by report authors, and users working with Q&A.

Consider an example of an Import model designed to analyze airline flight on-time performance (OTP). The model has a `Flight` table, which is a fact table that stores one row per flight. Each row records the flight date, flight number, departure and arrival airports, and any delay time (in minutes). There's also an `Airport` table, which is a dimension table that stores one row per airport. Each row describes the airport code, airport name, and the country or region.

Here's a partial model diagram of the two tables.

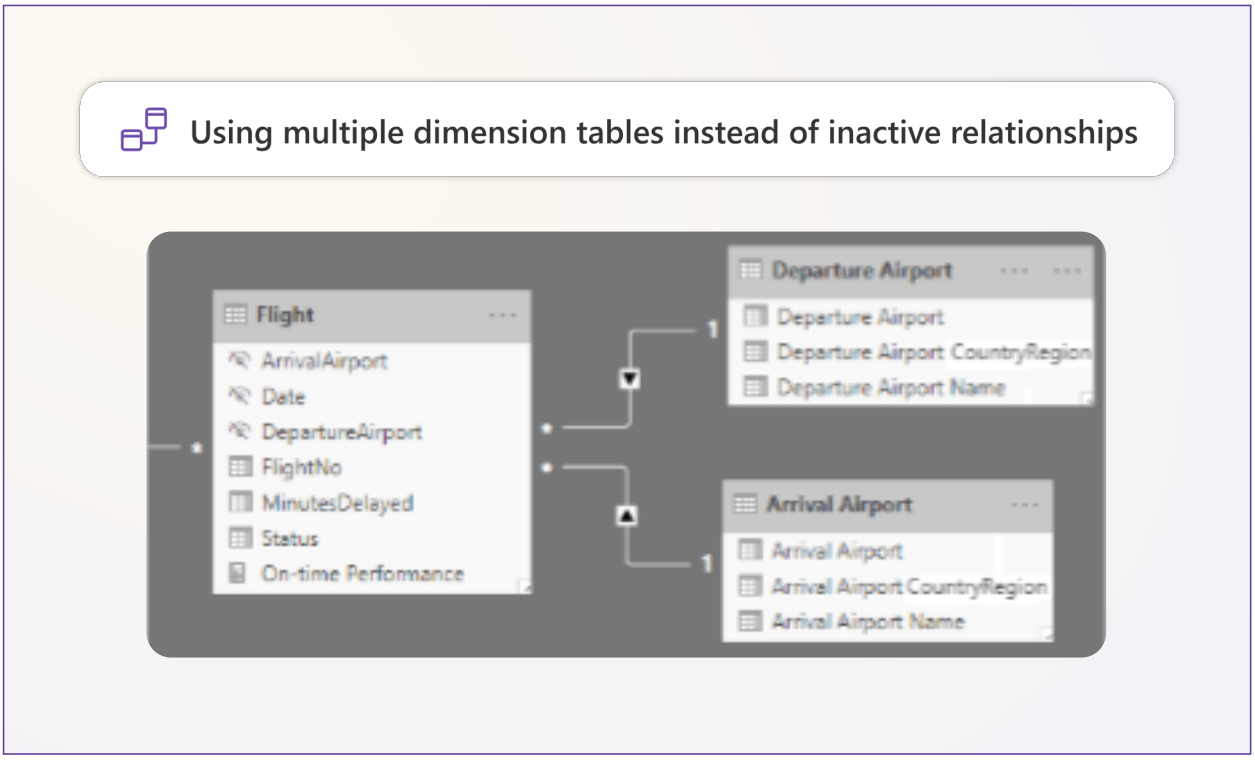


There are two model relationships between the `Flight` and `Airport` tables. In the `Flight` table, the `DepartureAirport` and `ArrivalAirport` columns relate to the `Airport` column of the `Airport` table. In star schema design, the `Airport` table is described as a role-playing dimension. In this model, the two roles are *departure airport* and *arrival airport*.

While this design works well for relational star schema designs, it doesn't work well for Power BI models. That's because model relationships are paths for filter propagation, and these paths must be deterministic. For more information on ensuring that filter propagation paths are deterministic see resolve relationship path ambiguity.Therefore—as presented in this example—one relationship is active while the other is inactive (represented by the dashed line). Specifically, it's the relationship to the `ArrivalAirport` column that's active, which means filters applied to the `Airport` table automatically propagate to the `ArrivalAirport` column of the `Flight` table.

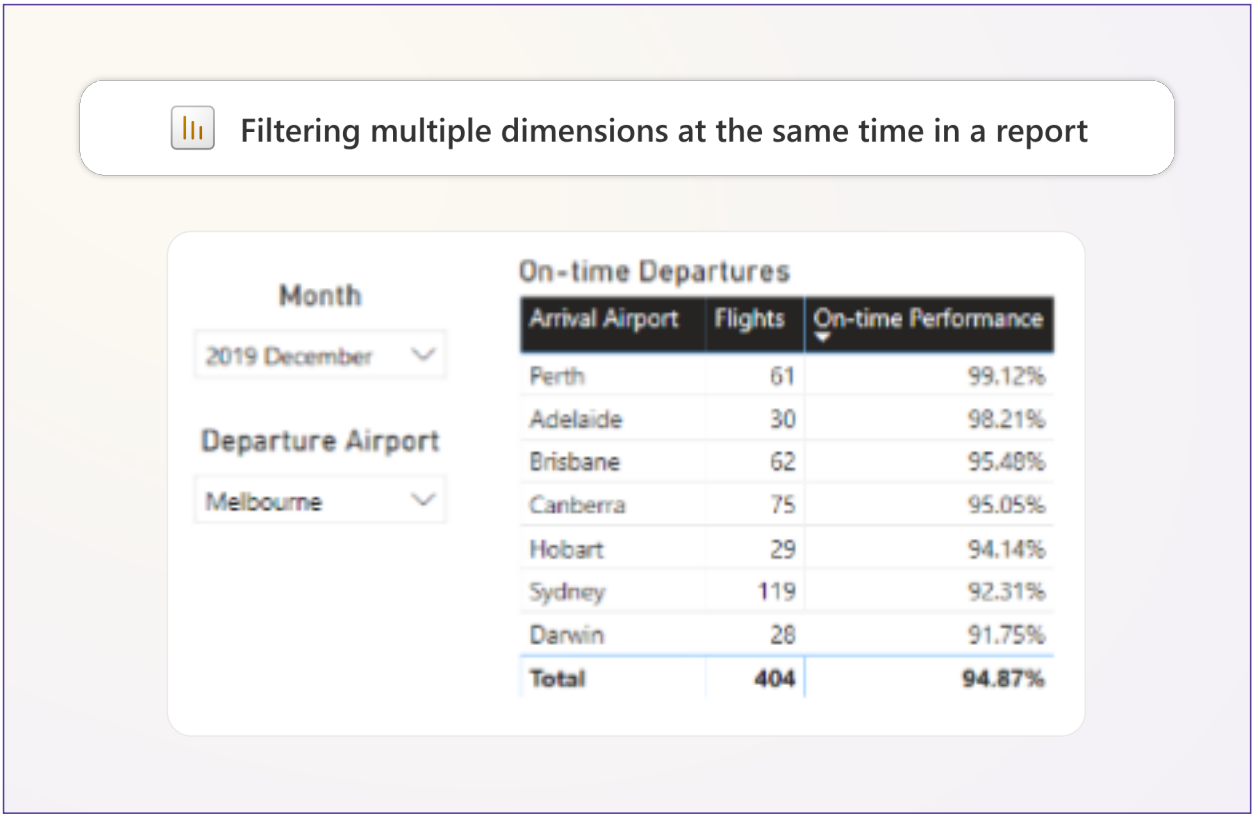
This model design imposes severe limitations on how the data can be reported. Specifically, it's not possible to filter the `Airport` table to automatically isolate flight details for a departure airport. As reports need to filter (or group) by departure and arrival airports *at the same time*, two active relationships are required. Translating this requirement into a Power BI model design means the model must have two airport tables.

Here's the improved model design.



The model now has two airport tables: `Departure Airport` and `Arrival Airport`. Each model relationship between these tables and the `Flight` table is active. Notice also that the column names in the `Departure Airport` and `Arrival Airport` tables are prefixed with the word *Departure* or *Arrival*.

The improved model design supports producing the following report design.



The report page filters by **Melbourne** as the departure airport, and the table visual groups by arrival airports.

> **Note**
>
> For Import models, the addition of another dimension table has resulted in an increased model size, and longer refresh times. As such, it contradicts the recommendations described in the Data reduction techniques for Import modeling article. However, in the example, the requirement to have only active relationships overrides these recommendations.
>
> Further, it's common that dimension tables store low row counts relative to fact table row counts. So, the increased model size and refresh times aren't likely to be excessively large.

Refactoring methodology

Here's a methodology to refactor a model from a single role-playing dimension table, to a design with *one table per role*.

1. Remove any inactive relationships.
2. Consider renaming the role-playing dimension table to better describe its role. In the example in this article, the `Airport` table is related to the `ArrivalAirport` column of the `Flight` table, so it's renamed as `Arrival Airport`.
3. Create a copy of the role-playing table, providing it with a name that reflects its role. If it's an Import table, we recommend you create a calculated table. If it's a DirectQuery table, you can duplicate the Power Query query.

   In the example, the `Departure Airport` table was created by using the following calculated table definition.

   ```
   DAX

   Departure Airport = 'Arrival Airport'
   ```

4. Create an active relationship to relate the new table.
5. Consider renaming the columns in the tables so they accurately reflect their role. In the example in this article, all columns are prefixed with the word *Departure* or *Arrival*. These names ensure report visuals, by default, will have self-describing and non-ambiguous labels. It also improves the Q&A experience, allowing users to easily write accurate questions.

6. Consider adding descriptions to role-playing tables. (In the **Data** pane, a description appears in a tooltip when a report author hovers their cursor over the table.) This way, you can communicate other filter propagation details to report authors.

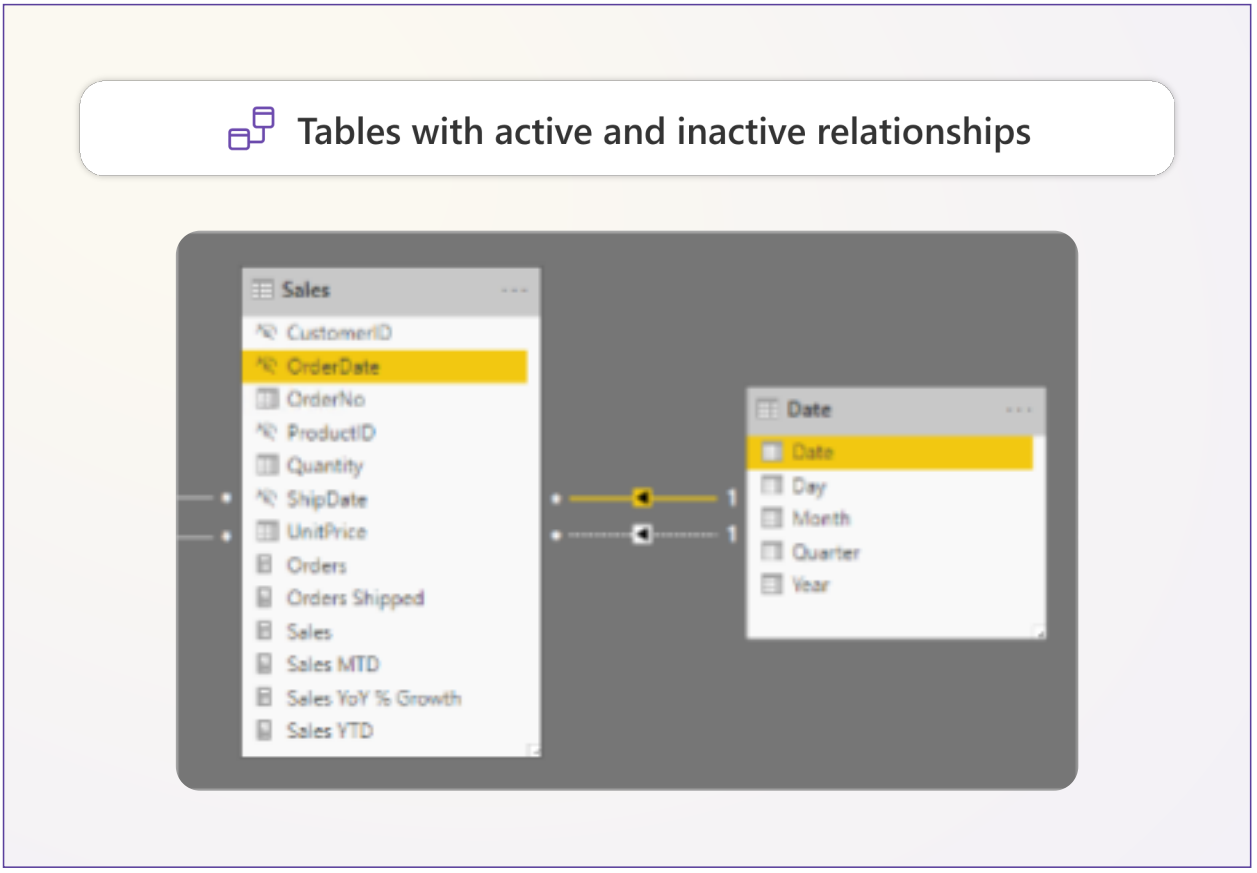
Inactive relationships

In specific circumstances, inactive relationships can address specific reporting needs.

Consider different model and reporting requirements:

- A sales model contains a `Sales` table that has two date columns: `OrderDate` and `ShipDate`.
- Each row in the `Sales` table records a single order.
- Date filters are almost always applied to the `OrderDate` column, which always stores a valid date.
- Only one measure requires date filter propagation to the `ShipDate` column, which can contain BLANKs (until the order is shipped).
- There's no requirement to simultaneously filter (or group) order *and* ship date periods.

Here's a partial model diagram of the two tables.



There are two model relationships between the `Sales` and `Date` tables. In the `Sales` table, the `OrderDate` and `ShipDate` columns relate to the `Date` column of the `Date` table. In this model, the two roles for the `Date` table are *order date* and *ship date*. It's the relationship to the `OrderDate` column that's active.

All of the six measures—except one—must filter by the `OrderDate` column. The `Orders Shipped` measure, however, must filter by the `ShipDate` column.

Here's the `Orders` measure definition. It simply counts the rows of the `Sales` table within the filter context. Any filters applied to the `Date` table propagate to the `OrderDate` column.

DAX

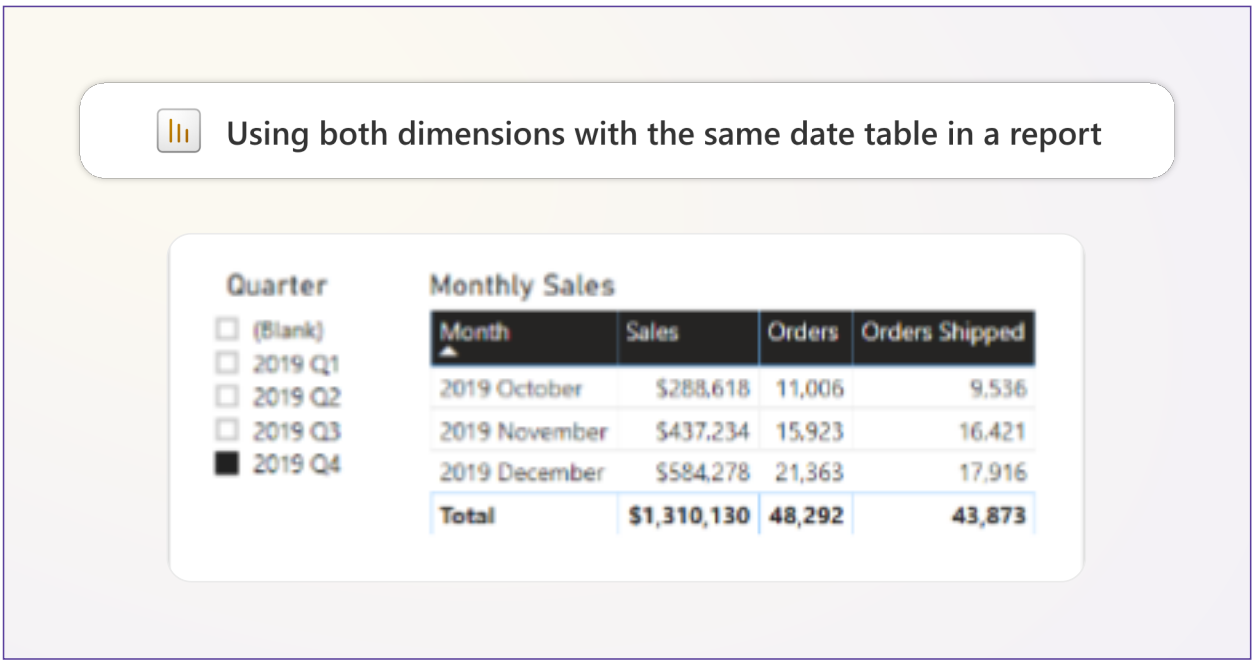
```
Orders = COUNTROWS(Sales)
```

Here's the `Orders Shipped` measure definition. It uses the USERELATIONSHIP DAX function, which activates filter propagation for a specific relationship, but only during the evaluation of the expression. In this example, the relationship to the `ShipDate` column is used.

DAX

```
Orders Shipped =
CALCULATE(
    COUNTROWS(Sales)
    ,USERELATIONSHIP('Date'[Date], Sales[ShipDate])
)
```

This model design supports producing the following report design.



The report page filters by quarter **2019 Q4**. The table visual groups by month and displays various sales statistics. The `Orders` and `Orders Shipped` measures produce different results. They each use the same summarization logic (count rows of the `Sales` table), but different `Date` table filter propagation.

Notice that the quarter slicer includes a BLANK option. This slicer option appears as a result of table expansion. While each `Sales` table row has a valid order date, some rows have a BLANK ship date—these orders are yet to be shipped. Table expansion considers inactive relationships too, and so BLANKs can appear due to BLANKs on the many-side of the relationship (or due to data integrity issues).

> **Note**
>
> Row-level security (RLS) filters only propagate through active relationships. RLS filters never propagate for inactive relationships, even when the USERELATIONSHIP DAX function is used by a measure definition.

Recommendations

We recommend that you define active relationships whenever possible, especially when RLS roles are defined for your data model. They widen the scope and potential of how your model can be used by report authors, and users working with Q&A. It means that role-playing dimension tables should be duplicated in your model.

In specific circumstances, however, you can define one or more inactive relationships for a role-playing dimension table. You can consider this approach when:

- There's no requirement for report visuals to simultaneously filter by different roles.
- You use the USERELATIONSHIP DAX function to activate a specific relationship for relevant model calculations.

Related content

For more information related to this article, check out the following resources:

- Model relationships in Power BI Desktop
- Understand star schema and the importance for Power BI
- Relationship troubleshooting guidance
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Bi-directional relationship guidance

Article • 12/27/2024

This article targets you as a data modeler who works with Power BI Desktop. It provides you with guidance on when to create bi-directional model relationships. A bi-directional relationship is one that filters in *both directions*.

> **Note**
>
> An introduction to model relationships is not covered in this article. If you're not completely familiar with relationships, their properties or how to configure them, we recommend that you first read the Model relationships in Power BI Desktop article.
>
> It's also important that you have an understanding of star schema design. For more information, see Understand star schema and the importance for Power BI.

Generally, we recommend that you minimize the use of bi-directional relationships. That's because they can negatively impact on model query performance, and possibly deliver confusing experiences for your report users.

However, there are three scenarios when bi-directional filtering can solve specific requirements:

- Special model relationships
- Slicer options "with data"
- Dimension-to-dimension analysis

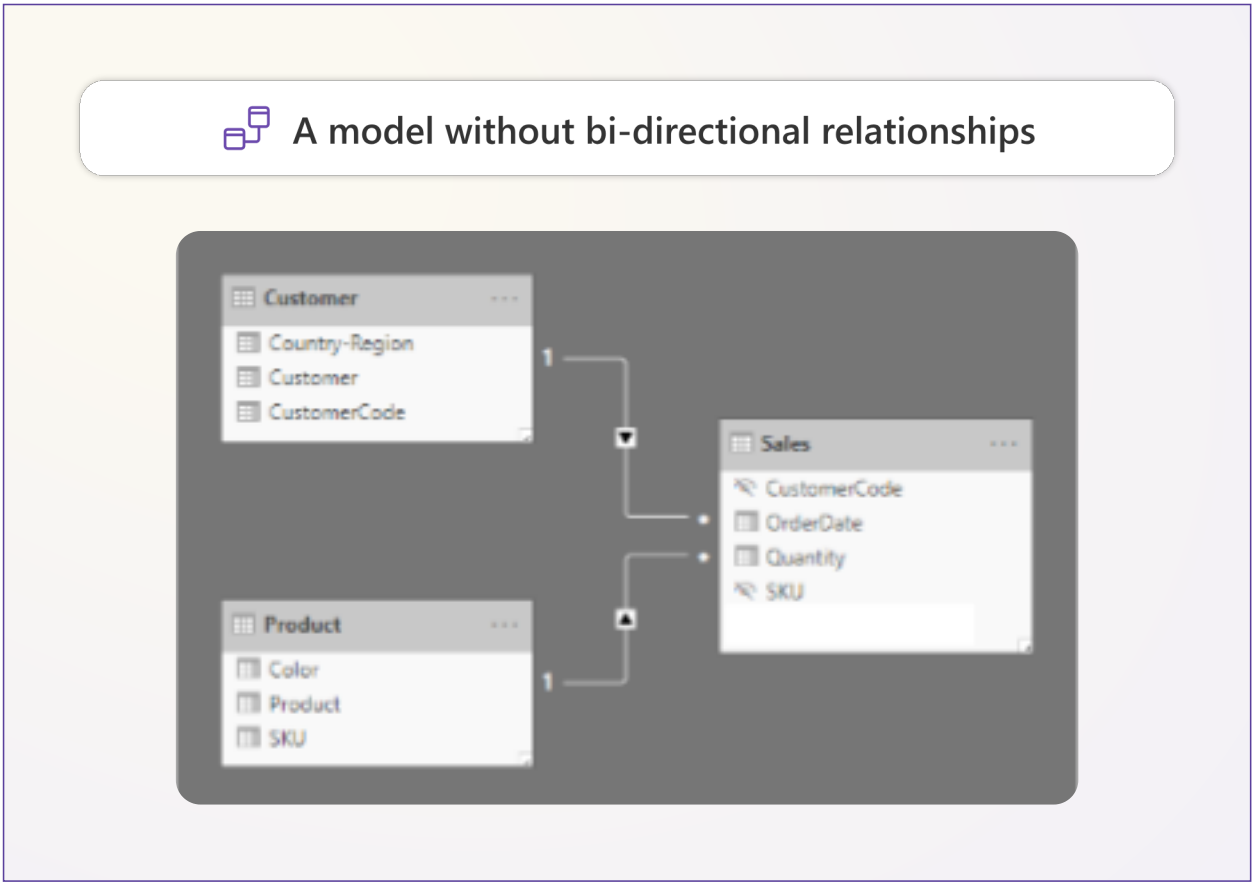
Special model relationships

Bi-directional relationships play an important role when creating the following two special model relationship types:

- **One-to-one**: All one-to-one relationships must be bi-directional—it isn't possible to configure otherwise. Generally, we don't recommend creating these types of relationships. For a complete discussion and alternative design patterns, see One-to-one relationship guidance.
- **Many-to-many**: When relating two dimension tables, a bridging table is required. A bi-directional filter is required to ensure filters propagate across the bridging table. For more information, see Many-to-many relationship guidance.

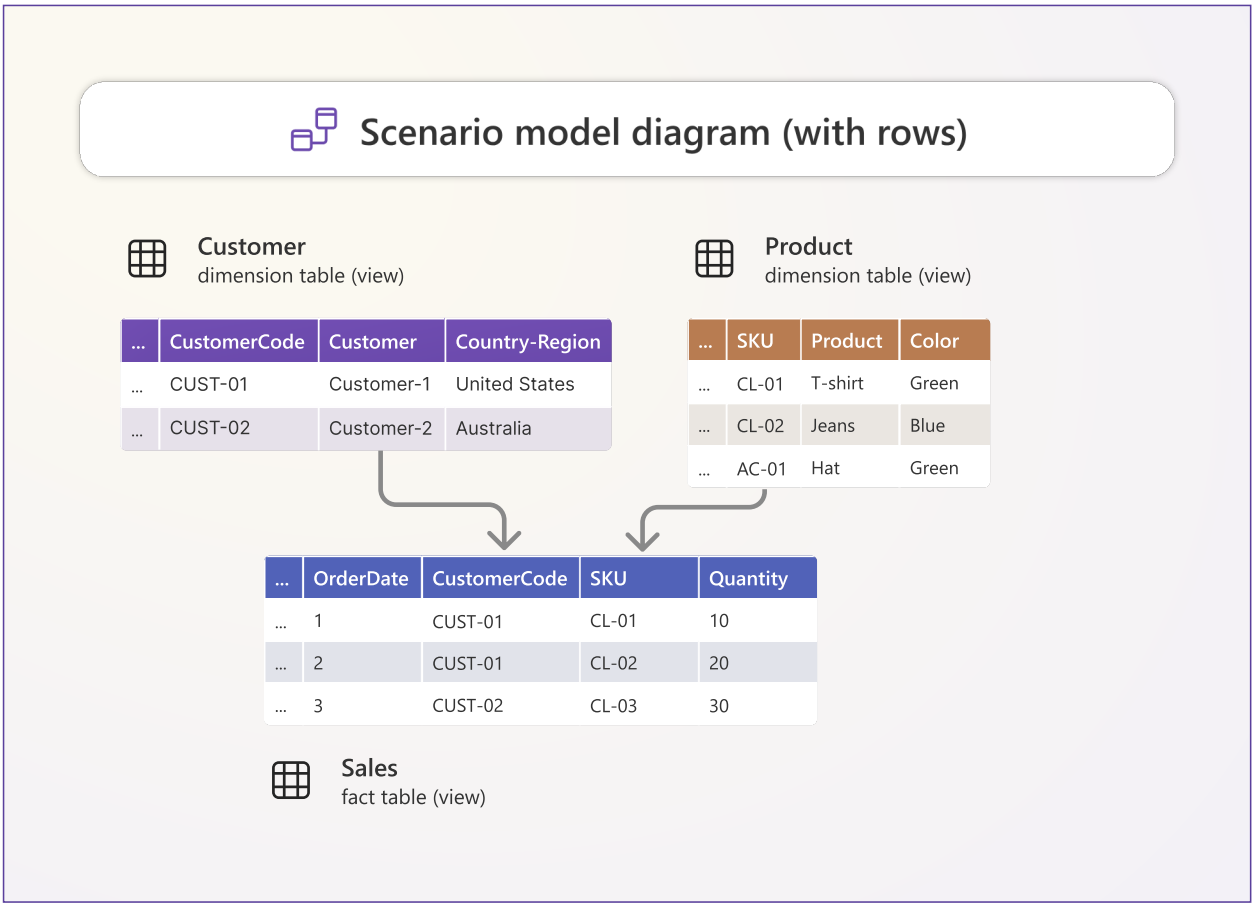
Slicer options "with data"

Bi-directional relationships can deliver slicers that limit options to where data exists. (If you're familiar with Excel PivotTables and slicers, it's the default behavior when sourcing data from a Power BI semantic model or an Analysis Services model.) To help explain what it means, first consider the following model diagram.



The first table is named `Customer`., and it contains three columns: `Country-Region`, `Customer`, and `CustomerCode`. The second table is named `Product`, and it contains three columns: `Color`, `Product`, and `SKU`. The third table is named `Sales`, and it contains four columns: `CustomerCode`, `OrderDate`, `Quantity`, and `SKU`. The `Customer` and `Product` tables are dimension tables, and each has a one-to-many relationship to the `Sales` table. Each relationship filters in a single direction.

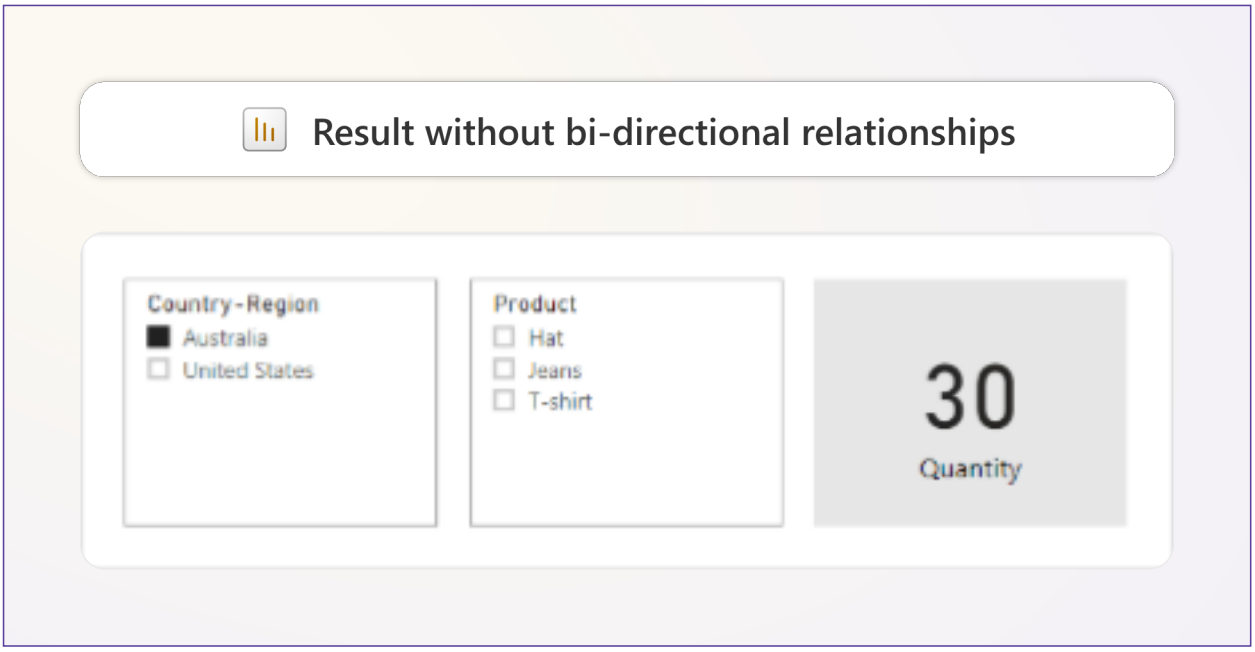
To help describe how bi-directional filtering works, the model diagram has been modified to reveal the table rows. All examples in this article are based on this data.



The row details for the three tables are described in the following bulleted list:

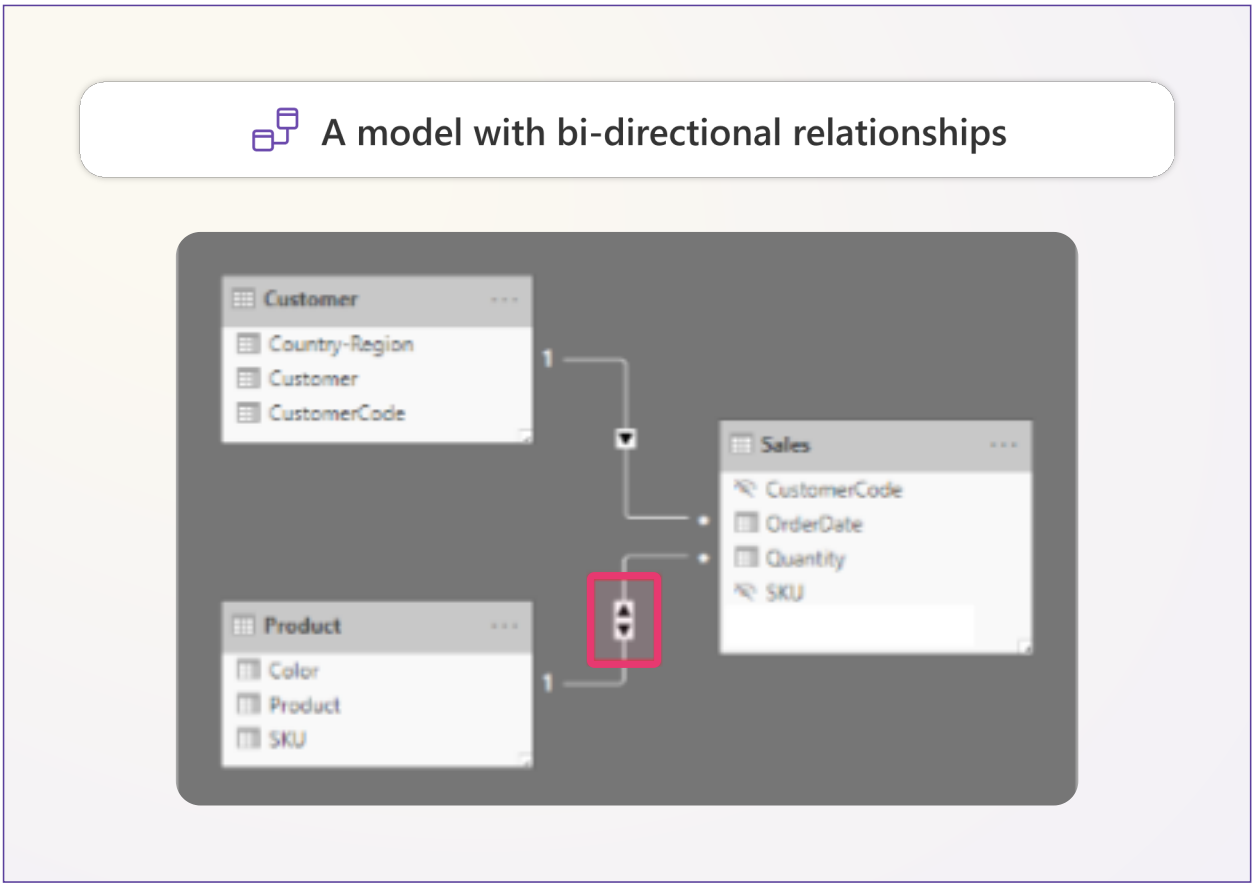
- The `Customer` table has two rows:
  - `CustomerCode` **CUST-01**, `Customer` **Customer-1**, `Country-Region` **United States**
  - `CustomerCode` **CUST-02**, `Customer` **Customer-2**, `Country-Region` **Australia**
- The `Product` table has three rows:
  - `SKU` **CL-01**, `Product` **T-shirt**, `Color` **Green**
  - `SKU` **CL-02**, `Product` **Jeans**, `Color` **Blue**
  - `SKU` **AC-01**, `Product` **Hat**, `Color` **Blue**
- The `Sales` table has three rows:
  - `OrderDate` **January 1 2019**, `CustomerCode` **CUST-01**, `SKU` **CL-01**, `Quantity` **10**
  - `OrderDate` **February 2 2019**, `CustomerCode` **CUST-01**, `SKU` **CL-02**, `Quantity` **20**
  - `OrderDate` **March 3 2019**, `CustomerCode` **CUST-02**, `SKU` **CL-01**, `Quantity` **30**

Now consider the following report page.

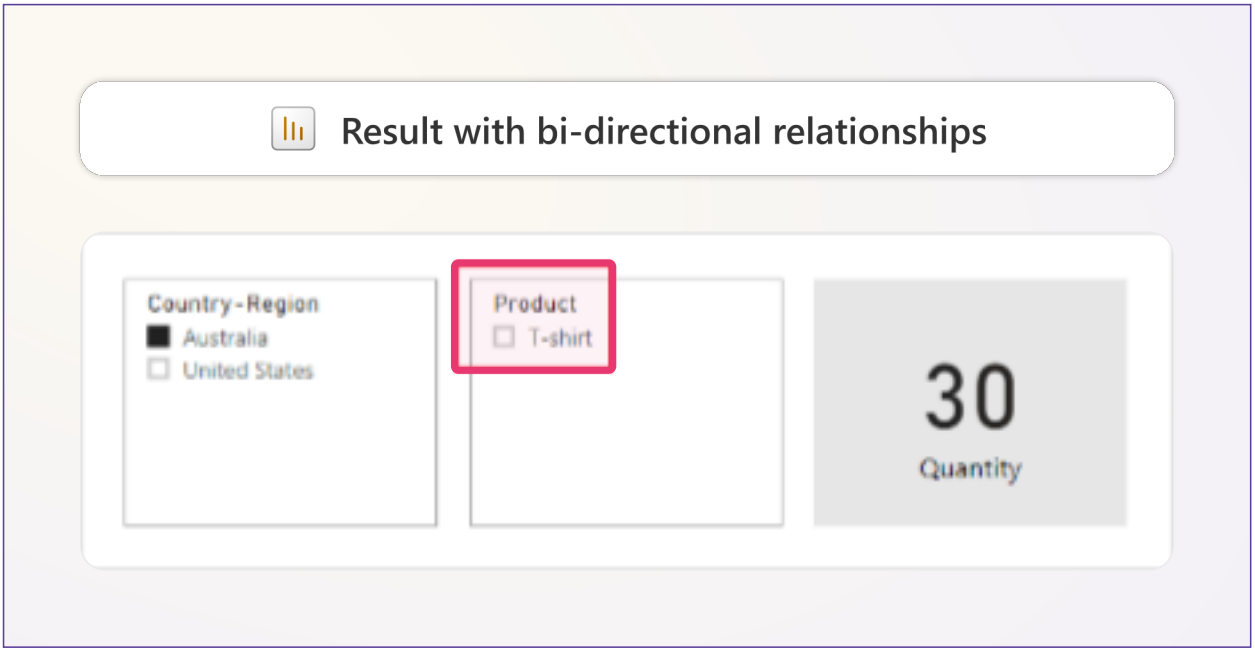


The page consists of two slicers and a card visual. The first slicer is based on the `Country-Region` field, and it has two options: Australia and United States. It currently slices by Australia. The second slicer is based on the `Product` field, and it has three options: Hat, Jeans, and T-shirt. No items are selected (meaning *no products* are filtered). The card visual displays a quantity of 30.

When report users slice by Australia, you might want to limit the product slicer to display options where data *relates* to Australian sales. That's what's meant by showing slicer options "with data". You can achieve this behavior by setting the relationship between the `Product` and `Sales` tables to filter *in both directions*.



The product slicer now lists a single option: T-shirt. This option represents the only product sold to Australian customers.



First, we recommend that you consider carefully whether this design works for your report users. Some report users find the experience confusing because they don't understand why slicer options dynamically appear or disappear when they interact with other slicers.

If you do decide to show slicer options "with data", we don't recommend you set up a bi-directional relationships. Bi-directional relationships require more processing and so

they can negatively impact on query performance—especially as the number of bi-directional relationships in the model increases.

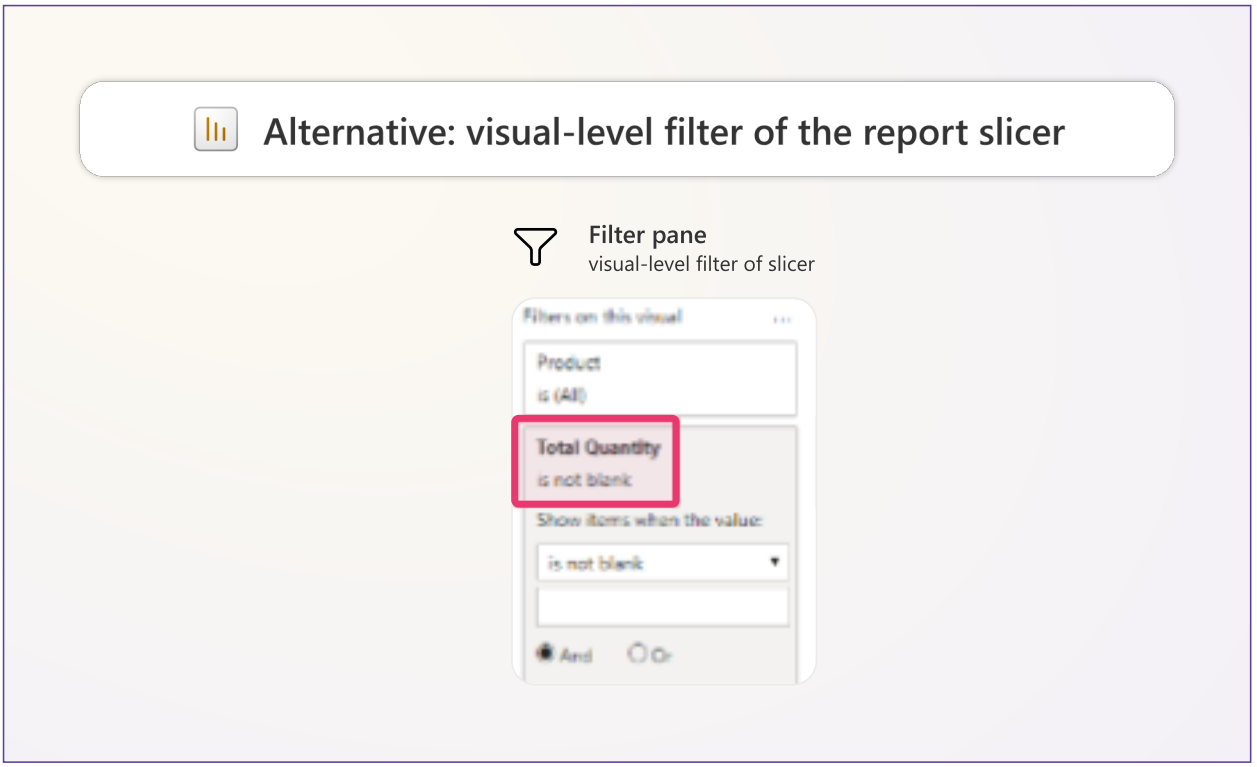
There's a better way to achieve the same result: Instead of using bi-directional filters, you can apply a visual-level filter to the product slicer itself.

Let's now consider that the relationship between the `Product` and `Sales` tables no longer filters in both directions. And, the following measure definition has been added to the `Sales` table.

DAX

```
Total Quantity = SUM(Sales[Quantity])
```

To show the product slicer options "with data", it simply needs to be filtered by the `Total Quantity` measure by using the "is not blank" condition.



Dimension-to-dimension analysis

A different scenario involving bi-directional relationships treats a fact table like a bridging table. This way, it supports analyzing dimension table data within the filter context of a different dimension table.

Using the example model in this article, consider how the following questions can be answered:

- How many colors were sold to Australian customers?
- How many countries/regions purchased jeans?

Both questions can be answered *without* summarizing data in the bridging fact table. They do, however, require that filters propagate from one dimension table to the other. When filters propagate via the fact table, summarization of dimension table columns can be achieved using the DISTINCTCOUNT DAX function—and possibly the MIN and MAX DAX functions.

As the fact table behaves like a bridging table, you can apply the many-to-many relationship guidance to relate two dimension tables. It will require setting up at least one relationship to filter in both directions. For more information, see Many-to-many relationship guidance.

However, as already described in this article, this design will likely result in a negative impact on performance, and the user experience consequences related to slicer options "with data". So, we recommend that you activate bi-directional filtering *in a measure definition* by using the CROSSFILTER DAX function instead. You can use the CROSSFILTER function to modify filter directions—or even disable the relationship—during the evaluation of an expression.

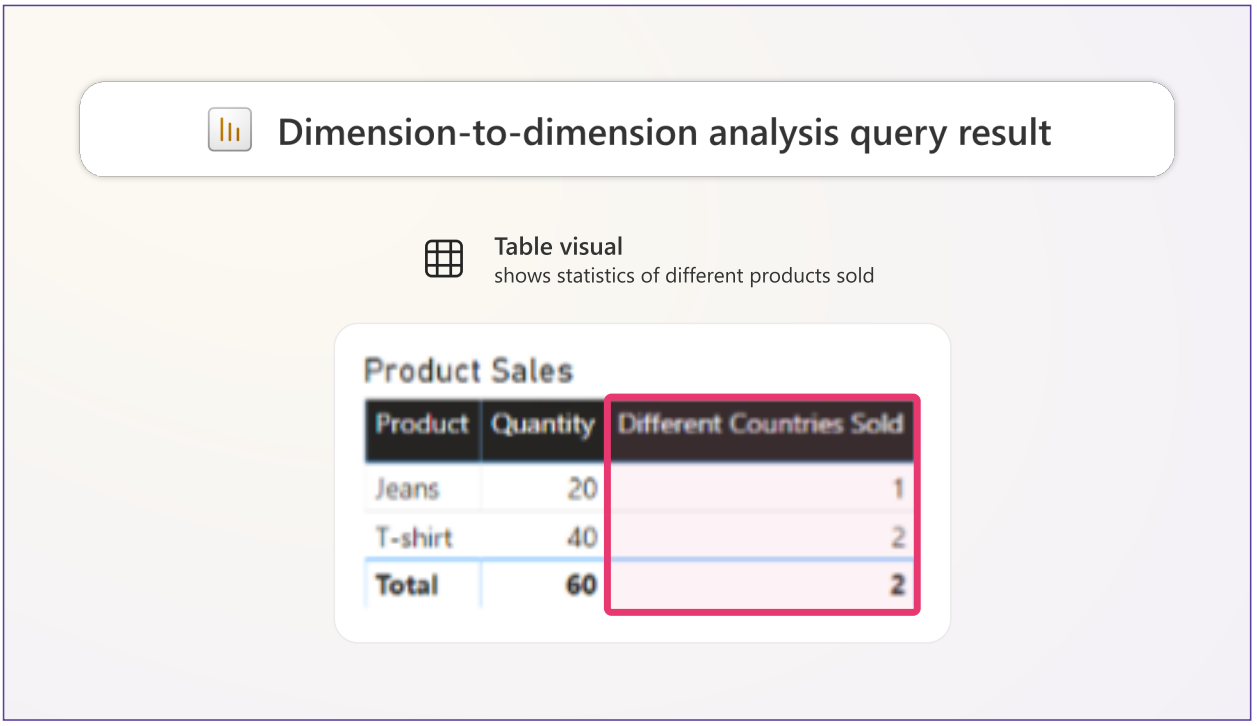
Consider the following measure definition added to the `Sales` table. In this example, the model relationship between the `Customer` and `Sales` tables has been set up to filter in a *single direction*.

DAX

```
Different Countries Sold =
CALCULATE(
    DISTINCTCOUNT(Customer[Country-Region]),
    CROSSFILTER(
        Customer[CustomerCode],
        Sales[CustomerCode],
        BOTH
    )
)
```

During the evaluation of the `Different Countries Sold` measure, the relationship between the `Customer` and `Sales` tables filters in both directions.

The following table visual present statistics for each product sold. The `Quantity` column is simply the sum of quantity values. The `Different Countries Sold` column represents the distinct count of country-region values of all customers who have purchased the product.



Related content

For more information related to this article, check out the following resources:

- Model relationships in Power BI Desktop
- Understand star schema and the importance for Power BI
- One-to-one relationship guidance
- Many-to-many relationship guidance
- Relationship troubleshooting guidance
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Relationship troubleshooting guidance

Article • 12/27/2024

This article targets you as a data modeler who works with Power BI Desktop. It provides guidance on how to troubleshoot specific issues that you might encounter when developing models and reports.

> **Note**
>
> An introduction to model relationships is not covered in this article. If you're not completely familiar with relationships, their properties or how to configure them, we recommend that you first read the Model relationships in Power BI Desktop article.
>
> It's also important that you have an understanding of star schema design. For more information, see Understand star schema and the importance for Power BI.

Troubleshooting

When a report visual is set up to use fields from two (or more) tables, and it doesn't present the correct result (or any result), it's possible that the issue is related to the model relationships.

In this case, here's a general troubleshooting checklist to follow. You can progressively work through the checklist until you identify the issue(s).

1. Switch the visual to a table or matrix, or open the *See Data* pane—it's easier to troubleshoot issues when you can see the query result.
2. If there's an empty query result, switch to Table view—verify that tables have been loaded with rows of data.
3. Switch to Model view—it's easy to see the relationships and quickly determine their properties.
4. Verify that relationships exist between the tables.
5. Verify that cardinality properties are correctly set—they could be incorrect if a "many"-side column presently contains unique values, and it has been incorrectly set as a "one" side.
6. Verify that the relationships are active (solid line).
7. Verify that the filter directions support propagation (interpret arrow heads).
8. Verify that the correct columns are related—either select the relationship, or hover the cursor over it to reveal the related columns.

9. Verify that the related column data types are the same, or at least compatible—it's possible to relate a text column to a whole number column, but filters won't find any matches to propagate filters.
10. Switch to Table view, and verify that matching values can be found in related columns.

Troubleshooting guide

Here's a list of issues and their possible reasons.

Expand table

Issue	Possible reason(s)
The visual displays no result	• The model is yet to be loaded with data. • No data exists within the filter context. • Row-level security (RLS) is enforced. • Relationships aren't propagating between tables—*follow checklist above*. • RLS is enforced, but a bi-directional relationship isn't enabled to propagate—see Row-level security (RLS) with Power BI Desktop.
The visual displays the same value for each grouping	• Relationships don't exist. • Relationships aren't propagating between tables—*follow checklist above*.
The visual displays results, but they aren't correct	• Visual is incorrectly set up. • Measure calculation logic is incorrect. • Model data needs to be refreshed. • Source data is incorrect. • Relationship columns are incorrectly related (for example, the `ProductID` column maps to the `CustomerID` column). • It's a relationship between two DirectQuery tables, and the "one"-side column of a relationship contains duplicate values.
BLANK groupings or slicer/filter items appear, and the source columns don't contain BLANKs	• It's a regular relationship, and "many"-side column contain values not stored in the "one"-side column—see Model relationships in Power BI Desktop. • It's a regular one-to-one relationship, and related columns contain BLANKs—see Model relationships in Power BI Desktop. • An inactive relationship "many"-side column stores BLANKs, or has values not stored on the "one" side.
The visual is missing data	• Incorrect/unexpected filters are applied. • RLS is enforced. • It's a limited relationship, and there are BLANKs in related columns, or data integrity issues—see Model relationships in

Issue	Possible reason(s)
	Power BI Desktop. • It's a relationship between two DirectQuery tables, the relationship is set to assume referential integrity, but there are data integrity issues (mismatched values in related columns).
RLS isn't correctly enforced	• Relationships aren't propagating between tables—*follow checklist above*. • RLS is enforced, but a bi-directional relationship isn't enabled to propagate—see Row-level security (RLS) with Power BI Desktop.

Related content

For more information related to this article, check out the following resources:

- Model relationships in Power BI Desktop
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

DirectQuery model guidance in Power BI Desktop

Article • 12/30/2024

This article targets data modelers developing Power BI DirectQuery models, developed by using either Power BI Desktop or the Power BI service. It describes DirectQuery use cases, limitations, and guidance. Specifically, the guidance is designed to help you determine whether DirectQuery is the appropriate mode for your model, and to improve the performance of your reports based on DirectQuery models. This article applies to DirectQuery models hosted in the Power BI service or Power BI Report Server.

This article isn't intended to provide a complete discussion on DirectQuery model design. For an introduction, refer to the DirectQuery models in Power BI Desktop article. For a deeper discussion, refer directly to the DirectQuery in SQL Server 2016 Analysis Services whitepaper. Bear in mind that the whitepaper describes using DirectQuery in SQL Server Analysis Services. Much of the content, however, is still applicable to Power BI DirectQuery models.

> **Note**
>
> For considerations when using DirectQuery storage mode for Dataverse, see Power BI modeling guidance for Power Platform.

This article doesn't directly cover composite models. A Composite model consists of at least one DirectQuery source, and possibly more. The guidance described in this article is still relevant—at least in part—to Composite model design. However, the implications of combining Import tables with DirectQuery tables aren't in scope for this article. For more information, see Use composite models in Power BI Desktop.

It's important to understand that DirectQuery models impose a different workload on the Power BI environment (Power BI service or Power BI Report Server) and also on the underlying data sources. If you determine that DirectQuery is the appropriate design approach, we recommend that you engage the right people on the project. We often see that a successful DirectQuery model deployment is the result of a team of IT professionals working closely together. The team usually consists of model developers and the source database administrators. It can also involve data architects, and data warehouse and ETL developers. Often, optimizations need to be applied directly to the data source to achieve good performance results.

Optimize data source performance

The relational database source can be optimized in several ways, as described in the following bulleted list.

> **Note**
>
> We understand that not all modelers have the permissions or skills to optimize a relational database. While it is the preferred layer to prepare the data for a DirectQuery model, some optimizations can also be achieved in the model design, without modifying the source database. However, best optimization results are often achieved by applying optimizations to the source database.

- **Ensure data integrity is complete**: It's especially important that dimension tables contain a column of unique values (dimension key) that maps to the fact table(s). It's also important that fact table dimension columns contain valid dimension key values. They'll allow configuring more efficient model relationships that expect matched values on both sides of relationships. When the source data lacks integrity, it's recommended that an "unknown" dimension record is added to effectively repair the data. For example, you can add a row to the `Product` table to represent an unknown product, and then assign it an out-of-range key, like -1. If rows in the `Sales` table contain a missing product key value, substitute them with -1. It ensures every `Sales` table product key value has a corresponding row in the `Product` table.

- **Add indexes**: Define appropriate indexes—on tables or views—to support the efficient retrieval of data for the expected report visual filtering and grouping. For SQL Server, Azure SQL Database or Azure Synapse Analytics (formerly SQL Data Warehouse) sources, see SQL Server Index Architecture and Design Guide for helpful information on index design guidance. For SQL Server or Azure SQL Database volatile sources, see Get started with Columnstore for real-time operational analytics.

- **Design distributed tables**: For Azure Synapse Analytics (formerly SQL Data Warehouse) sources, which use Massively Parallel Processing (MPP) architecture, consider configuring large fact tables as hash distributed, and dimension tables to replicate across all the compute nodes. For more information, see Guidance for designing distributed tables in Azure Synapse Analytics (formerly SQL Data Warehouse).

- **Ensure required data transformations are materialized**: For SQL Server relational database sources (and other relational database sources), computed columns can be added to tables. These columns are based on an expression, like **Quantity** multiplied by **UnitPrice**. Computed columns can be persisted (materialized) and, like regular columns, sometimes they can be indexed. For more information, see Indexes on Computed Columns.

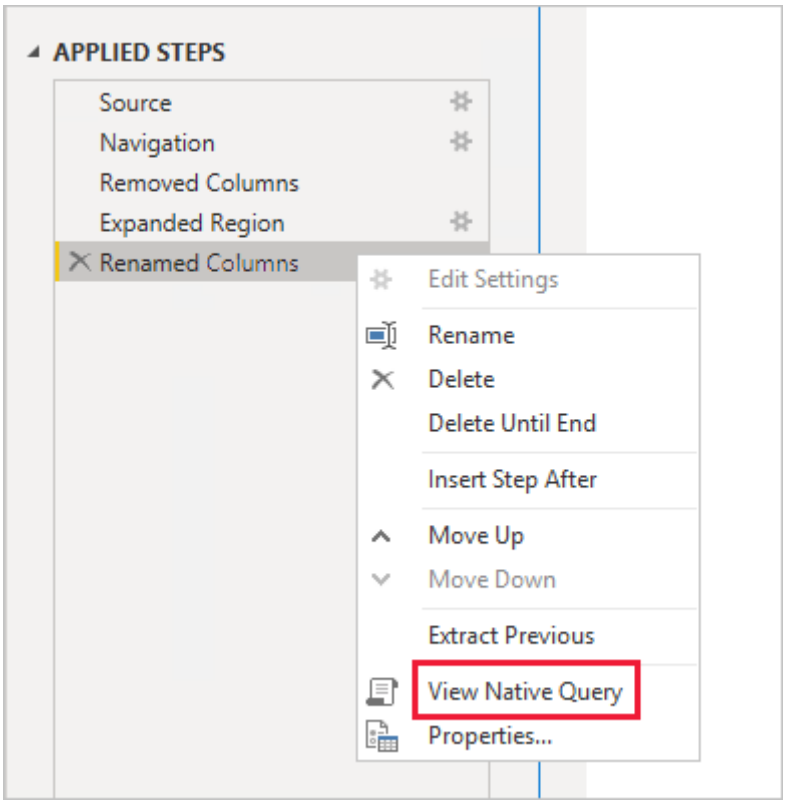
  Consider also indexed views that can pre-aggregate fact table data at a higher grain. For example, if the `Sales` table stores data at order line level, you could create a view to summarize this data. The view could be based on a `SELECT` statement that groups the `Sales` table data by date (at month level), customer, product, and summarizes measure values like sales, quantity, etc. The view can then be indexed. For SQL Server or Azure SQL Database sources, see Create Indexed Views.

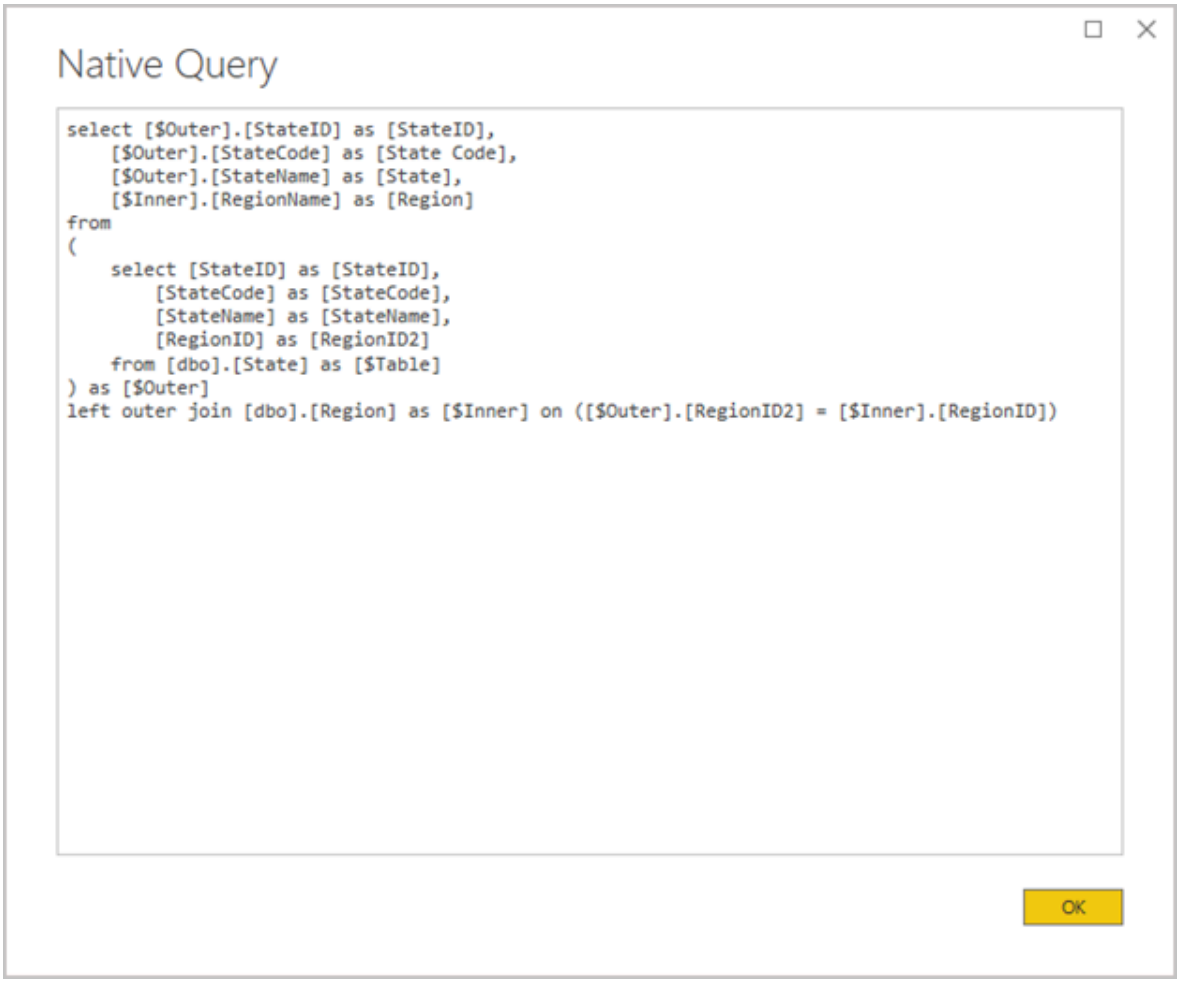
- **Materialize a date table**: A common modeling requirement involves adding a date table to support time-based filtering. To support the known time-based filters in your organization, create a table in the source database, and ensure it's loaded with a range of dates encompassing the fact table dates. Also ensure that it includes columns for useful time periods, like year, quarter, month, week, etc.

Optimize model design

A DirectQuery model can be optimized in many ways, as described in the following bulleted list.

- **Avoid complex Power Query queries**: An efficient model design can be achieved by removing the need for the Power Query queries to apply any transformations. It means that each query maps to a single relational database source table or view. You can preview a representation of the actual SQL query statement for a Power Query applied step, by selecting the **View Native Query** option.





- **Examine the use of calculated columns and data type changes**: DirectQuery models support adding calculations and Power Query steps to convert data types. However, better performance is often achieved by materializing transformation results in the relational database source, when possible.

- **Do not use Power Query relative date filtering**: It's possible to define relative date filtering in a Power Query query. For example, to retrieve to the sales orders that were created in the last year (relative to today's date). This type of filter translates to an inefficient native query, as follows:

  ```
  SQL

  …
  from [dbo].[Sales] as [_]
  where [_].[OrderDate] >= convert(datetime2, '2018-01-01 00:00:00') and
  [_].[OrderDate] < convert(datetime2, '2019-01-01 00:00:00'))
  ```

  A better design approach is to include relative time columns in the date table. These columns store offset values relative to the current date. For example, in a `RelativeYear` column, the value zero represents current year, -1 represents previous year, etc. Preferably, the `RelativeYear` column is materialized in the date table. While less efficient, it could also be added as a model calculated column, based on the expression using the TODAY and DATE DAX functions.

- **Keep measures simple**: At least initially, it's recommended to limit measures to simple aggregates. The aggregate functions include SUM, COUNT, MIN, MAX, and AVERAGE. Then, if the measures are sufficiently responsive, you can experiment with more complex measures, but paying attention to the performance for each. While the CALCULATE DAX function can be used to produce sophisticated measure expressions that manipulate filter context, they can generate expensive native queries that don't perform well.

- **Avoid relationships on calculated columns**: Model relationships can only relate a single column in one table to a single column in a different table. Sometimes, however, it's necessary to relate tables by using multiple columns. For example, the `Sales` and `Geography` tables are related by two columns: `CountryRegion` and `City`. To create a relationship between the tables, a single column is required, and in the `Geography` table, the column must contain unique values. Concatenating the country/region and city with a hyphen separator could achieve this result.

  The combined column can be created with either a Power Query custom column, or in the model as a calculated column. However, it should be avoided as the calculation expression will be embedded into the source queries. Not only is it inefficient, it commonly prevents the use of indexes. Instead, add materialized columns in the relational database source, and consider indexing them. You can also consider adding surrogate key columns to dimension tables, which is a common practice in relational data warehouse designs.

There's one exception to this guidance, and it concerns the use of the COMBINEVALUES DAX function. The purpose of this function is to support multi-column model relationships. Rather than generate an expression that the relationship uses, it generates a multi-column SQL join predicate.

- **Avoid relationships on "Unique Identifier" columns**: Power BI doesn't natively support the unique identifier (GUID) data type. When defining a relationship between columns of this type, Power BI generates a source query with a join involving a cast. This query-time data conversion commonly results in poor performance. Until this case is optimized, the only workaround is to materialize columns of an alternative data type in the underlying database.

- **Hide the one-side column of relationships**: The one-side column of a relationship should be hidden. (It's usually the primary key column of dimension tables.) When hidden, it isn't available in the **Data** pane and so can't be used to configure a visual. The many-side column can remain visible if it is useful to group or filter reports by the column values. For example, consider a model where a relationship exists between `Sales` and `Product` tables. The relationship columns contain product SKU (Stock-Keeping Unit) values. If product SKU must be added to visuals, it should be visible only in the `Sales` table. When this column is used to filter or group in a visual, Power BI generates a query that doesn't need to join the `Sales` and `Product` tables.

- **Set relationships to enforce integrity**: The **Assume Referential Integrity** property of DirectQuery relationships determines whether Power BI generates source queries using an `INNER JOIN` rather than an `OUTER JOIN`. It generally improves query performance, though it does depend on the specifics of the relational database source. For more information, see Assume referential integrity settings in Power BI Desktop.

- **Avoid use of bi-directional relationship filtering**: Use of bi-directional relationship filtering can lead to query statements that don't perform well. Only use this relationship feature when necessary, and it's usually the case when implementing a many-to-many relationship across a bridging table. For more information, see Relationships with a many-many cardinality in Power BI Desktop.

- **Limit parallel queries**: You can set the maximum number of connections DirectQuery opens for each underlying data source. It controls the number of queries concurrently sent to the data source.
  - The setting is only enabled when there's at least one DirectQuery source in the model. The value applies to all DirectQuery sources, and to any new DirectQuery sources added to the model.

- Increasing the **Maximum Connections per Data Source** value ensures more queries (up to the maximum number specified) can be sent to the underlying data source, which is useful when numerous visuals are on a single page, or many users access a report at the same time. Once the maximum number of connections is reached, further queries are queued until a connection becomes available. Increasing this limit does result in more load on the underlying data source, so the setting isn't guaranteed to improve overall performance.
- When the model is published to Power BI, the maximum number of concurrent queries sent to the underlying data source also depends on the environment. Different environments (such as Power BI, Power BI Premium, or Power BI Report Server) each can impose different throughput constraints. For more information about capacity resource limitations, see Microsoft Fabric capacity licenses and Configure and manage capacities in Power BI Premium.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Optimize report designs

Reports based on a DirectQuery semantic model can be optimized in many ways, as described in the following bulleted list.

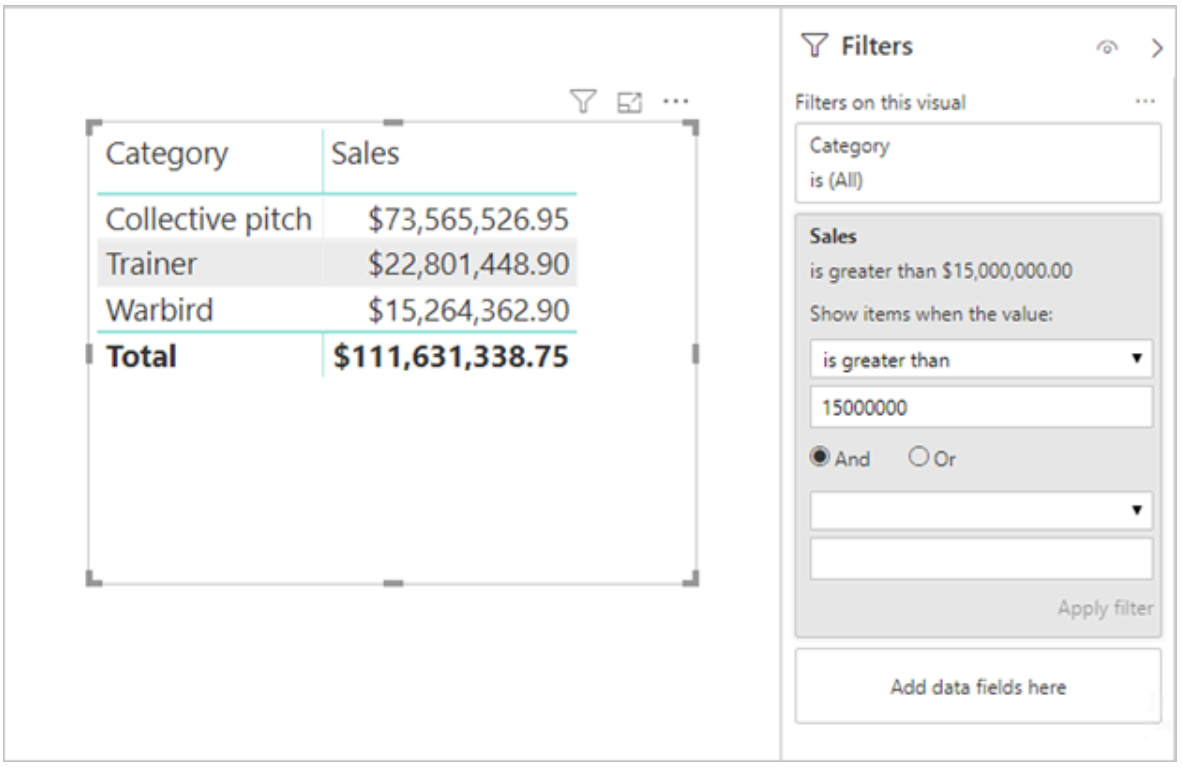
- **Enable query reduction techniques**: Power BI Desktop *Options and Settings* includes a Query Reduction page. This page has three helpful options. It's possible to disable cross-highlighting and cross-filtering by default, though it can be overridden by editing interactions. It's also possible to show an Apply button on slicers and filters. The slicer or filter options won't be applied until the report user clicks the button. If you enable these options, we recommend that you do so when first creating the report.
- **Apply filters first**: When first designing reports, we recommend that you apply any applicable filters—at report, page, or visual level—before mapping fields to the visual fields. For example, rather than dragging in the `CountryRegion` and `Sales` measures, and then filtering by a particular year, apply the filter on the `Year` field

first. It's because each step of building a visual will send a query, and while it's possible to then make another change before the first query has completed, it still places unnecessary load on the underlying data source. By applying filters early, it generally makes those intermediate queries less costly and faster. Also, failing to apply filters early can result in exceeding the 1 million-row limit, as described in about DirectQuery.

- **Limit the number of visuals on a page**: When a report page is opened (and when page filters are applied) all of the visuals on a page are refreshed. However, there's a limit on the number of queries that can be sent in parallel, imposed by the Power BI environment and the **Maximum Connections per Data Source** model setting, as described above. So, as the number of page visuals increases, there's higher chance that they'll be refreshed in a serial manner. It increases the time taken to refresh the entire page, and it also increases the chance that visuals may display inconsistent results (for volatile data sources). For these reasons, it's recommended to limit the number of visuals on any page, and instead have more simpler pages. Replacing multiple card visuals with a single multi-row card visual can achieve a similar page layout.
- **Switch off interaction between visuals**: Cross-highlighting and cross-filtering interactions require queries be submitted to the underlying source. Unless these interactions are necessary, it's recommended they be switched off if the time taken to respond to users' selections would be unreasonably long. These interactions can be switched off, either for the entire report (as described above for Query Reduction options), or on a case-by-case basis. For more information, see How visuals cross-filter each other in a Power BI report.

In addition to the above list of optimization techniques, each of the following reporting capabilities can contribute to performance issues:

- **Measure filters**: Visuals containing measures (or aggregates of columns) can have filters applied to those measures. For example, the visual below shows **Sales** by **Category**, but only for categories with more than $15 million of sales.



It may result in two queries being sent to the underlying source:

- The first query will retrieve the categories meeting the condition (Sales > $15 million)
- The second query will then retrieve the necessary data for the visual, adding the categories that met the condition to the `WHERE` clause

It generally performs fine if there are hundreds or thousands of categories, as in this example. Performance can degrade, however, if the number of categories is much larger (and indeed, the query will fail if there are more than 1 million categories meeting the condition, due to the 1 million-row limit discussed above).

- **TopN filters**: Advanced filters can be defined to filter on only the top (or bottom) N values ranked by a measure. For example, to display only the top five categories in the above visual. Like the measure filters, it will also result in two queries being sent to the underlying data source. However, the first query will return all categories from the underlying source, and then the top N are determined based on the returned results. Depending on the cardinality of the column involved, it can lead to performance issues (or query failures due to the 1 million-row limit).

- **Median**: Generally, any aggregation (Sum, Count Distinct, etc.) is pushed to the underlying source. However, it's not true for Median, as this aggregate isn't supported by the underlying source. In such cases, detail data is retrieved from the underlying source, and Power BI evaluates the median from the returned results. It's fine when the median is to be calculated over a relatively small number of results, but performance issues (or query failures due to the 1 million-row limit) will

occur if the cardinality is large. For example, median country/region population might be reasonable, but median sales price might not be.

- **Multi-select slicers**: Allowing multi-selection in slicers and filters can cause performance issues. It's because as the user selects additional slicer items (for example, building up to the 10 products they're interested in), each new selection results in a new query being sent to the underlying source. While the user can select the next item prior to the query completing, it results in extra load on the underlying source. This situation can be avoided by showing the Apply button, as described above in the query reduction techniques.

- **Visual totals**: By default, tables and matrices display totals and subtotals. In many cases, additional queries must be sent to the underlying source to obtain the values for the totals. It applies whenever using Count Distinct or Median aggregates, and in all cases when using DirectQuery over SAP HANA or SAP Business Warehouse. Such totals should be switched off (by using the Format pane) if not necessary.

Convert to a Composite Model

The benefits of Import and DirectQuery models can be combined into a single model by configuring the storage mode of the model tables. The table storage mode can be Import or DirectQuery, or both, known as Dual. When a model contains tables with different storage modes, it's known as a Composite model. For more information, see Use composite models in Power BI Desktop.

There are many functional and performance enhancements that can be achieved by converting a DirectQuery model to a Composite model. A Composite model can integrate more than one DirectQuery source, and it can also include aggregations. Aggregation tables can be added to DirectQuery tables to import a summarized representation of the table. They can achieve dramatic performance enhancements when visuals query higher-level aggregates. For more information, see Aggregations in Power BI Desktop.

Educate users

It's important to educate your users on how to efficiently work with reports based on DirectQuery semantic models. Your report authors should be educated on the content described in the Optimize report designs section.

We recommend that you educate your report consumers about your reports that are based on DirectQuery semantic models. It can be helpful for them to understand the general data architecture, including any relevant limitations described in this article. Let them know to expect that refresh responses and interactive filtering may at times be slow. When report users understand why performance degradation happens, they're less likely to lose trust in the reports and data.

When delivering reports on volatile data sources, be sure to educate report users on the use of the Refresh button. Let them know also that it may be possible to see inconsistent results, and that a refresh of the report can resolve any inconsistencies on the report page.

Related content

For more information about DirectQuery, check out the following resources:

- DirectQuery models in Power BI Desktop
- Use DirectQuery in Power BI Desktop
- DirectQuery model troubleshooting in Power BI Desktop
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Composite model guidance in Power BI Desktop

Article • 12/30/2024

This article targets data modelers who develop Power BI composite models. It describes composite model use cases, and provides you with design guidance. Specifically, the guidance can help you determine whether a composite model is appropriate for your solution. If it is, then this article will also help you design optimal composite models and reports.

> **Note**
>
> An introduction to composite models isn't covered in this article. If you're not completely familiar with composite models, we recommend that you first read the Use composite models in Power BI Desktop article.
>
> Because composite models consist of at least one DirectQuery source, it's also important that you have a thorough understanding of model relationships, DirectQuery models, and DirectQuery model design guidance.

Composite model use cases

By definition, a composite model combines multiple *source groups*. A source group can represent imported data or a connection to a DirectQuery source. A DirectQuery source can be either a relational database or another tabular model, which can be a Power BI semantic model or an Analysis Services tabular model. When a tabular model connects to another tabular model, it's known as *chaining*. For more information, see Use composite models in Power BI Desktop.

> **Note**
>
> When a model connects to a tabular model but doesn't extend it with additional data, it's not a composite model. In this case, it's a DirectQuery model that connects to a remote model—so it comprises just the one source group. You might create this type of model to modify source model object properties, like a table name, column sort order, format string, or others.

Connecting to tabular models is especially relevant when extending an *enterprise semantic model* (when it's a Power BI semantic model or Analysis Services model). An enterprise semantic model is fundamental to the development and operation of a data warehouse. It provides an abstraction layer over the data in the data warehouse to present business definitions and terminology. It's commonly used as a link between physical data models and reporting tools, like Power BI. In most organizations, it's managed by a central team, and that's why it's described as *enterprise*. For more information, see the enterprise BI usage scenario.

You can consider developing a composite model in the following situations:

- Your model could be a DirectQuery model, and you want to boost performance. In a composite model, you can improve performance by setting up appropriate storage for each table. You can also add user-defined aggregations. Both of these optimizations are described later in this article.
- You want to combine a DirectQuery model with more data, which must be imported into the model. You can load imported data from a different data source, or from calculated tables.
- You want to combine two or more DirectQuery data sources into a single model. These sources could be relational databases or other tabular models.

> **Note**
>
> Composite models can't include connections to certain external analytic databases. These databases include SAP Business Warehouse, and SAP HANA when treating SAP HANA as a multidimensional source.

Evaluate other model design options

While Power BI composite models can solve particular design challenges, they can contribute to slow performance. Also, in some situations, unexpected calculation results can occur (described later in this article). For these reasons, evaluate other model design options when they exist.

Whenever possible, it's best to develop a model in import mode. This mode provides the greatest design flexibility, and best performance. However, challenges related to large data volumes, or reporting on near real-time data, can't always be solved by import models. In either of these cases, you can consider a DirectQuery model, providing your data is stored in a single data source that's supported by DirectQuery mode. For more information, see DirectQuery models in Power BI Desktop.

> **Tip**
>
> If your objective is only to extend an existing tabular model with more data, whenever possible, add that data to the existing data source.

Table storage mode

In a composite model, you can set the storage mode for each table (except calculated tables).

- **DirectQuery**: We recommend that you set this mode for tables that represent large data volumes, or which need to deliver near real-time results. Data will never be imported into these tables. Usually, these tables will be fact tables, which are tables that are summarized.
- **Import**: We recommend that you set this mode for tables that aren't used for filtering and grouping of fact tables in DirectQuery or Hybrid mode. It's also the only option for tables based on sources not supported by DirectQuery mode. Calculated tables are always import tables.
- **Dual**: We recommend that you set this mode for dimension tables, when there's a possibility they'll be queried together with DirectQuery fact tables from the same source.
- **Hybrid**: We recommend that you set this mode by adding import partitions, and one DirectQuery partition to a fact table when you want to include the latest data changes in real time, or when you want to provide fast access to the most frequently used data through import partitions while leaving the bulk of more infrequently used data in the data warehouse.

There are several possible scenarios when Power BI queries a composite model.

- **Queries only import or dual table(s)**: Power BI retrieves all data from the model cache. It will deliver the fastest possible performance. This scenario is common for dimension tables queried by filters or slicer visuals.
- **Queries dual table(s) or DirectQuery table(s) from the same source**: Power BI retrieves all data by sending one or more native queries to the DirectQuery source. It will deliver good performance, especially when appropriate indexes exist on the source tables. This scenario is common for queries that relate dual dimension tables and DirectQuery fact tables. These queries are *intra source group*, and so all one-to-one or one-to-many relationships are evaluated as regular relationships.
- **Queries dual table(s) or hybrid table(s) from the same source**: This scenario is a combination of the previous two scenarios. Power BI retrieves data from the model

cache when it's available in import partitions, otherwise it sends one or more native queries to the DirectQuery source. It will deliver the fastest possible performance because only a slice of the data is queried in the data warehouse, especially when appropriate indexes exist on the source tables. As for the dual dimension tables and DirectQuery fact tables, these queries are intra source group, and so all one-to-one or one-to-many relationships are evaluated as regular relationships.

- **All other queries**: These queries involve cross source group relationships. It's either because an import table relates to a DirectQuery table, or a dual table relates to a DirectQuery table from a different source—in which case it behaves as an import table. All relationships are evaluated as limited relationships. It also means that groupings applied to non-DirectQuery tables must be sent to the DirectQuery source as materialized subqueries (virtual tables). In this case, the native query can be inefficient, especially for large grouping sets.

In summary, we recommend that you:

- Consider carefully that a composite model is the right solution—while it allows model-level integration of different data sources, it also introduces design complexities with possible consequences (described later in this article).
- Set the storage mode to **DirectQuery** when a table is a fact table storing large data volumes, or when it needs to deliver near real-time results.
- Consider using hybrid mode by defining an incremental refresh policy and real-time data, or by partitioning the fact table by using TOM, TMSL, or a third-party tool. For more information, see Incremental refresh and real-time data for semantic models and the Advanced data model management usage scenario.
- Set the storage mode to **Dual** when a table is a dimension table, and it will be queried together with DirectQuery or hybrid fact tables that are in the same source group.
- Set appropriate refresh frequencies to keep the model cache for dual and hybrid tables (and any dependent calculated tables) in sync with the source database(s).
- Strive to ensure data integrity across source groups (including the model cache) because limited relationships will eliminate rows in query results when related column values don't match.
- Whenever possible, optimize DirectQuery data sources with appropriate indexes for efficient joins, filtering, and grouping.

User-defined aggregations

You can add user-defined aggregations to DirectQuery tables. Their purpose is to improve performance for *higher grain* queries.

When aggregations are cached in the model, they behave as import tables (although they can't be used like a model table). Adding import aggregations to a DirectQuery model will result in a composite model.

> ⓘ **Note**
>
> Hybrid tables don't support aggregations because some of the partitions operate in import mode. It's not possible to add aggregations at the level of an individual DirectQuery partition.

We recommend that an aggregation follows a basic rule: Its row count should be at least a factor of 10 smaller than the underlying table. For example, if the underlying table stores 1 billion rows, then the aggregation table shouldn't exceed 100 million rows. This rule ensures that there's an adequate performance gain relative to the cost of creating and maintaining the aggregation.

Cross source group relationships

When a model relationship spans source groups, it's known as a *cross source group relationship*. Cross source group relationships are also *limited* relationships because there's no guaranteed "one" side. For more information, see Relationship evaluation.

> ⓘ **Note**
>
> In some situations, you can avoid creating a cross source group relationship. See the Use Sync slicers topic later in this article.

When defining cross source group relationships, consider the following recommendations.

- **Use low-cardinality relationship columns**: For best performance, we recommend that the relationship columns be low cardinality, meaning they should store less than 50,000 unique values. This recommendation is especially true when combining tabular models, and for non-text columns.
- **Avoid using large text relationship columns**: If you must use text columns in a relationship, calculate the expected text length for the filter by multiplying the cardinality by the average length of the text column. The possible text length shouldn't exceed 1,000,000 characters.
- **Raise the relationship granularity**: If possible, create relationships at a higher level of granularity. For example, instead of relating a date table on its date key, use its

month key instead. This design approach requires that the related table includes a month key column, and reports won't be able to show daily facts.

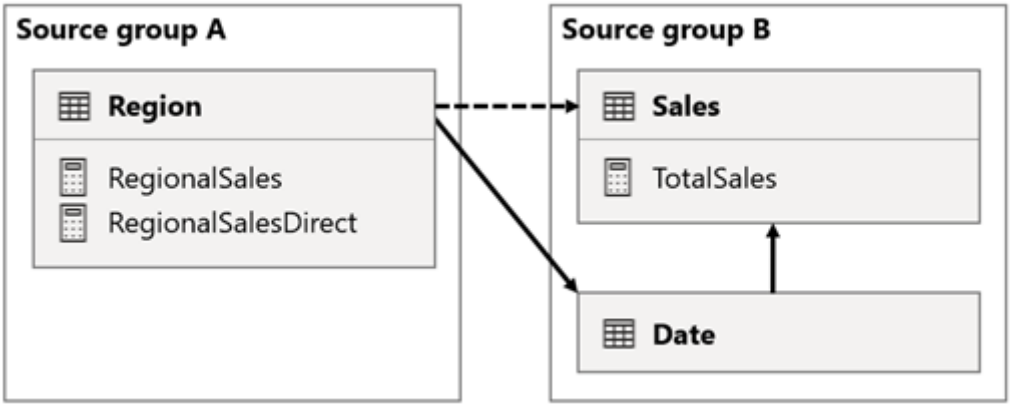
- **Strive to achieve a simple relationship design**: Only create a cross source group relationship when it's needed, and try to limit the number of tables in the relationship path. This design approach will help to improve performance and avoid ambiguous relationship paths.

> ⚠ **Warning**
>
> Because Power BI Desktop doesn't thoroughly validate cross source group relationships, it's possible to create ambiguous relationships.

Cross source group relationship scenario 1

Consider a scenario of a complex relationship design and how it could produce different —yet valid—results.

In this scenario, the `Region` table in source group `A` has a relationship to the `Date` table and `Sales` table in source group `B`. The relationship between the `Region` table and the `Date` table is active, while the relationship between the `Region` table and the `Sales` table is inactive. Also, there's an active relationship between the `Region` table and the `Sales` table, both of which are in source group `B`. The `Sales` table includes a measure named `TotalSales`, and the `Region` table includes two measures named `RegionalSales` and `RegionalSalesDirect`.



Here are the measure definitions.

DAX

```
TotalSales = SUM(Sales[Sales])
RegionalSales = CALCULATE([TotalSales], USERELATIONSHIP(Region[RegionID],
Sales[RegionID]))
```

```
RegionalSalesDirect = CALCULATE(SUM(Sales[Sales]),
USERELATIONSHIP(Region[RegionID], Sales[RegionID]))
```

Notice how the `RegionalSales` measure refers to the `TotalSales` measure, while the `RegionalSalesDirect` measure doesn't. Instead, the `RegionalSalesDirect` measure uses the expression `SUM(Sales[Sales])`, which is the expression of the `TotalSales` measure.

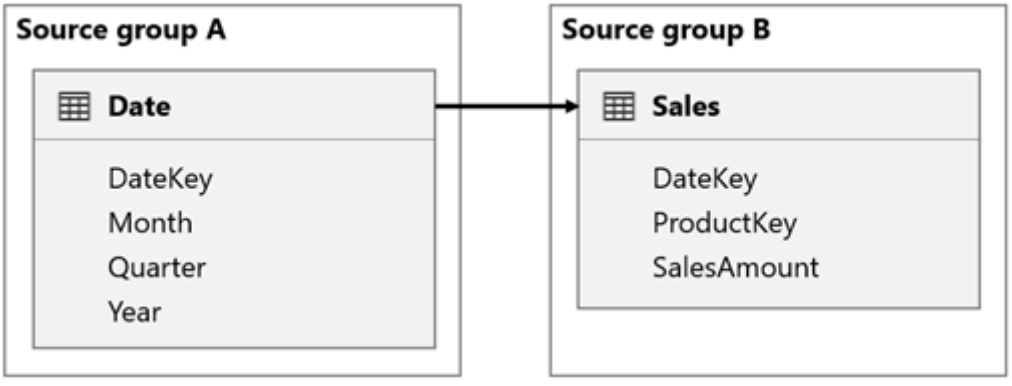
The difference in the result is subtle. When Power BI evaluates the `RegionalSales` measure, it applies the filter from the `Region` table to both the `Sales` table and the `Date` table. Therefore, the filter also propagates from the `Date` table to the `Sales` table. In contrast, when Power BI evaluates the `RegionalSalesDirect` measure, it only propagates the filter from the `Region` table to the `Sales` table. The results returned by `RegionalSales` measure and the `RegionalSalesDirect` measure could differ, even though the expressions are semantically equivalent.

> **Important**
>
> Whenever you use the `CALCULATE` function with an expression that's a measure in a remote source group, test the calculation results thoroughly.

Cross source group relationship scenario 2

Consider a scenario when a cross source group relationship has high-cardinality relationship columns.

In this scenario, the `Date` table is related to the `Sales` table on the `DateKey` columns. The data type of the `DateKey` columns is integer, storing whole numbers that use the *yyyymmdd* format. The tables belong to different source groups. Further, it's a high-cardinality relationship because the earliest date in the `Date` table is January 1, 1900 and the latest date is December 31, 2100—so there's a total of 73,414 rows in the table (one row for each date in the 1900-2100 time span).



There are two cases for concern.

First, when you use the `Date` table columns as filters, filter propagation will filter the `DateKey` column of the `Sales` table to evaluate measures. When filtering by a single year, like 2022, the DAX query will include a filter expression like `Sales[DateKey] IN { 20220101, 20220102, …20221231 }`. The text size of the query can grow to become extremely large when the number of values in the filter expression is large, or when the filter values are long strings. It's expensive for Power BI to generate the long query and for the data source to run the query.

Second, when you use `Date` table columns—like `Year`, `Quarter`, or `Month`—as grouping columns, it results in filters that include all unique combinations of year, quarter, or month, *and* the `DateKey` column values. The string size of the query, which contains filters on the grouping columns and the relationship column, can become extremely large. That's especially true when the number of grouping columns and/or the cardinality of the join column (the `DateKey` column) is large.

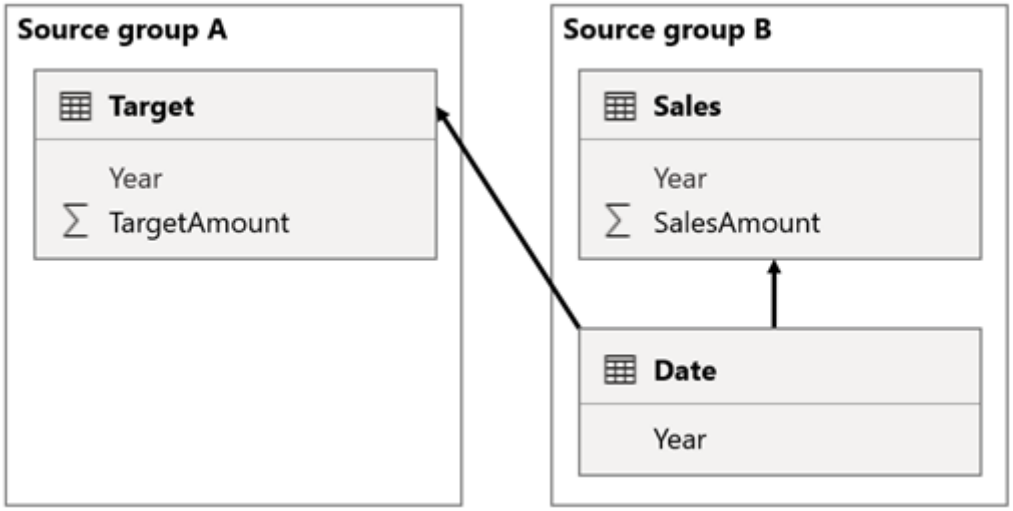
To address any performance issues, you can:

- Add the `Date` table to the data source, resulting in a single source group model (meaning, it's no longer a composite model).
- Raise the granularity of the relationship. For instance, you could add a `MonthKey` column to both tables and create the relationship on those columns. However, by raising the granularity of the relationship, you lose the ability to report on daily sales activity (unless you use the `DateKey` column from the `Sales` table).

Cross source group relationship scenario 3

Consider a scenario when there aren't matching values between tables in a cross source group relationship.

In this scenario, the `Date` table in source group `B` has a relationship to the `Sales` table in that source group, and also to the `Target` table in source group `A`. All relationships are one-to-many from the `Date` table relating the `Year` columns. The `Sales` table includes a `SalesAmount` column that stores sales amounts, while the `Target` table includes a `TargetAmount` column that stores target amounts.



The `Date` table stores the years 2021 and 2022. The `Sales` table stores sales amounts for years 2021 (100) and 2022 (200), while the `Target` table stores target amounts for 2021 (100), 2022 (200), *and 2023 (300)*—a future year.

Date

Year
2021
2022

Sales

Year	SalesAmount
2021	100
2022	200

Target

Year	TargetAmount
2021	100
2022	200
2023	300

When a Power BI table visual queries the composite model by grouping on the `Year` column from the `Date` table and summing the `SalesAmount` and `TargetAmount` columns, it won't show a target amount for 2023. That's because the cross source group relationship is a limited relationship, and so it uses `INNER JOIN` semantics, which eliminate rows where there's no matching value on both sides. It will, however, produce a correct target amount total (600), because a `Date` table filter doesn't apply to its evaluation.

Year	Sum of SalesAmount	Sum of TargetAmount
2021	100	100
2022	200	200
Total	**300**	**600**

If the relationship between the `Date` table and the `Target` table is an intra source group relationship (assuming the `Target` table belonged to source group `B`), the visual will include a *(Blank)* year to show the 2023 (and any other unmatched years) target amount.

> **Important**
>
> To avoid misreporting, ensure that there are matching values in the relationship columns when dimension and fact tables reside in different source groups.

For more information about limited relationships, see Relationship evaluation.

Calculations

You should consider specific limitations when adding calculated columns and calculation groups to a composite model.

Calculated columns

Calculated columns added to a DirectQuery table that source their data from a relational database, like Microsoft SQL Server, are limited to expressions that operate on a single row at a time. These expressions can't use DAX iterator functions, like `SUMX`, or filter context modification functions, like `CALCULATE`.

> **Note**
>
> It's not possible to add calculated columns or calculated tables that depend on chained tabular models.

A calculated column expression on a remote DirectQuery table is limited to intra-row evaluation only. However, you can author such an expression, but it will result in an error when it's used in a visual. For example, if you add a calculated column to a remote DirectQuery table named `DimProduct` by using the expression `[Product Sales] / SUM (DimProduct[ProductSales])`, you'll be able to successfully save the expression in the model. However, it will result in an error when it's used in a visual because it violates the intra-row evaluation restriction.

In contrast, calculated columns added to a remote DirectQuery table that's a tabular model, which is either a Power BI semantic model or Analysis Services model, are more flexible. In this case, all DAX functions are allowed because the expression will be evaluated within the source tabular model.

Many expressions require Power BI to materialize the calculated column before using it as a group or filter, or aggregating it. When a calculated column is materialized over a large table, it can be costly in terms of CPU and memory, depending on the cardinality

of the columns that the calculated column depends on. In this case, we recommend that you add those calculated columns to the source model.

> **Note**
>
> When you add calculated columns to a composite model, be sure to test all model calculations. Upstream calculations may not work correctly because they didn't consider their influence on the filter context.

Calculation groups

If calculation groups exist in a source group that connects to a Power BI semantic model or an Analysis Services model, Power BI could return unexpected results. For more information, see Calculation groups, query and measure evaluation.

Model design

You should always optimize a Power BI model by adopting a star schema design.

> **Tip**
>
> For more information, see **Understand star schema and the importance for Power BI**.

Be sure to create dimension tables that are separate from fact tables so that Power BI can interpret joins correctly and produce efficient query plans. While this guidance is true for any Power BI model, it's especially true for models that you recognize will become a source group of a composite model. It will allow for simpler and more efficient integration of other tables in downstream models.

Whenever possible, avoid having dimension tables in one source group that relate to a fact table in a different source group. That's because it's better to have *intra* source group relationships than *cross* source group relationships, especially for high-cardinality relationship columns. As described earlier, cross source group relationships rely on having matching values in the relationship columns, otherwise unexpected results may be shown in report visuals.

Row-level security

If your model includes user-defined aggregations, calculated columns on import tables, or calculated tables, ensure that any row-level security (RLS) is set up correctly and tested.

If the composite model connects to other tabular models, RLS rules are only applied on the source group (local model) where they're defined. They won't be applied to other source groups (remote models). Also, you can't define RLS rules on a table from another source group nor can you define RLS rules on a local table that has a relationship to another source group.

Report design

In some situations, you can improve the performance of a composite model by designing an optimized report layout.

Single source group visuals

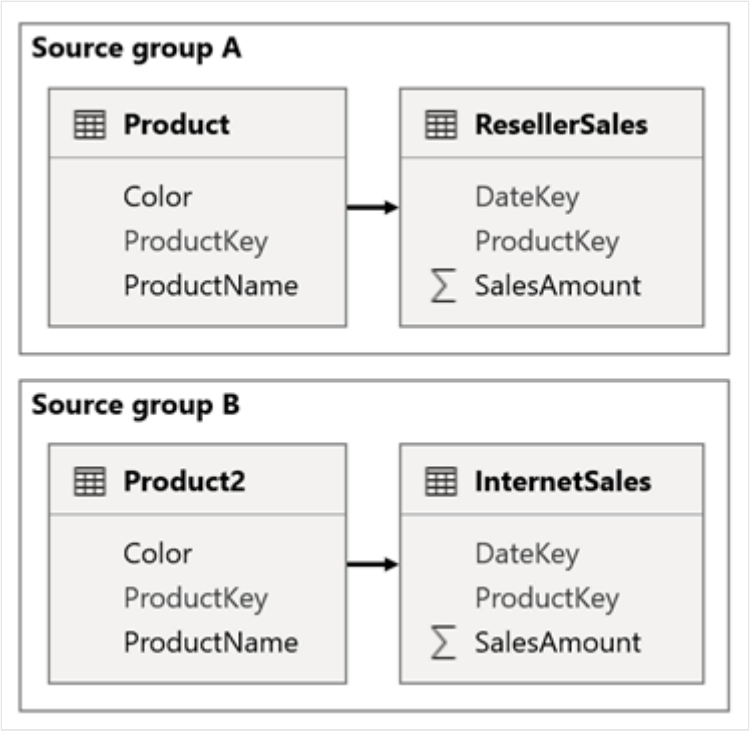
Whenever possible, create visuals that use fields from a single source group. That's because queries generated by visuals will perform better when the result is retrieved from a single source group. Consider creating two visuals positioned side by side that retrieve data from two different source groups.

Use sync slicers

In some situations, you can set up sync slicers to avoid creating a cross source group relationship in your model. It can allow you to combine source groups *visually* that can perform better.

Consider a scenario when your model has two source groups. Each source group has a product dimension table used to filter reseller and internet sales.

In this scenario, source group `A` contains the `Product` table that's related to the `ResellerSales` table. Source group `B` contains the `Product2` table that's related to the `InternetSales` table. There aren't any cross source group relationships.



In the report, you add a slicer that filters the page by using the `Color` column of the `Product` table. By default, the slicer filters the `ResellerSales` table, but not the `InternetSales` table. You then add a hidden slicer by using the `Color` column of the `Product2` table. By setting an identical group name (found in the sync slicers **Advanced options**), filters applied to the visible slicer automatically propagate to the hidden slicer.

> ⓘ Note
>
> While using sync slicers can avoid the need to create a cross source group relationship, it increases the complexity of the model design. Be sure to educate other users on why you designed the model with duplicate dimension tables. Avoid confusion by hiding dimension tables that you don't want other users to use. You can also add description text to the hidden tables to document their purpose.

For more information, see Sync separate slicers.

Other guidance

Here's some other guidance to help you design and maintain composite models.

- **Performance and scale**: If your reports were previously live connected to a Power BI semantic model or Analysis Services model, the Power BI service could reuse visual caches across reports. After you convert the live connection to create a local DirectQuery model, reports will no longer benefit from those caches. As a result, you might experience slower performance or even refresh failures. Also, the

workload for the Power BI service will increase, which might require you to scale up your capacity or distribute the workload across other capacities. For more information about data refresh and caching, see Data refresh in Power BI.

- **Renaming**: We don't recommend that you rename semantic models used by composite models, or rename their workspaces. That's because composite models connect to Power BI semantic models by using the workspace and semantic model names (and not their internal unique identifiers). Renaming a semantic model or workspace could break the connections used by your composite model.
- **Governance**: We don't recommend that your *single version of the truth* model is a composite model. That's because it would be dependent on other data sources or models, which if updated, could result in breaking the composite model. Instead, we recommended that you publish an enterprise semantic model as the single version of truth. Consider this model to be a reliable foundation. Other data modelers can then create composite models that extend the foundation model to create specialized models.
- **Data lineage**: Use the data lineage and semantic model impact analysis features before publishing composite model changes. These features are available in the Power BI service, and they can help you to understand how semantic models are related and used. It's important to understand that you can't perform impact analysis on external semantic models that are displayed in lineage view but are in fact located in another workspace. To perform impact analysis on an external semantic model, you need to navigate to the source workspace.
- **Schema updates**: You should refresh your composite model in Power BI Desktop when schema changes are made to upstream data sources. You'll then need to republish the model to the Power BI service. Be sure to thoroughly test calculations and dependent reports.

Related content

For more information related to this article, check out the following resources.

- Use composite models in Power BI Desktop
- Model relationships in Power BI Desktop
- DirectQuery models in Power BI Desktop
- Use DirectQuery in Power BI Desktop
- Storage mode in Power BI Desktop
- User-defined aggregations
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Row-level security (RLS) guidance in Power BI Desktop

Article • 12/30/2024

This article targets you as a data modeler working with Power BI Desktop. It describes good design practices for enforcing row-levels security (RLS) in your data models.

It's important to understand RLS filters *table rows*. They can't be configured to restrict access to model objects, including tables, columns, or measures.

> **Note**
>
> This article doesn't describe RLS or how to set it up. For more information, see Restrict data access with row-level security (RLS) for Power BI Desktop.
>
> Also, it doesn't cover enforcing RLS in live connections to external-hosted models with Azure Analysis Services or SQL Server Analysis Services. In these cases, RLS is enforced by Analysis Services. When Power BI connects using single-sign on (SSO), Analysis Services will enforce RLS (unless the account has admin privileges).

Create roles

It's possible to create multiple roles. When you're considering the permission needs for a single report user, strive to create a single role that grants all those permissions, instead of a design where a report user will be a member of multiple roles. It's because a report user could map to multiple roles, either directly by using their user account or indirectly by security group membership. Multiple role mappings can result in unexpected outcomes.

When a report user is assigned to multiple roles, RLS filters become additive. It means report users can see table rows that represent the union of those filters. What's more, in some scenarios it's not possible to guarantee that a report user doesn't see rows in a table. So, unlike permissions applied to SQL Server database objects (and other permission models), the "once denied always denied" principle doesn't apply.

Consider a model with two roles: The first role, named **Workers**, restricts access to all `Payroll` table rows by using the following rule expression:

```
DAX
```

```
FALSE()
```

> **Note**
>
> A rule will return no table rows when its expression evaluates to `FALSE`.

Yet, a second role, named **Managers**, allows access to all `Payroll` table rows by using the following rule expression:

DAX

```
TRUE()
```

Take care: Should a report user map to both roles, they'll see all `Payroll` table rows.

Optimize RLS

RLS works by automatically applying filters to every DAX query, and these filters may have a negative impact on query performance. So, efficient RLS comes down to good model design. It's important to follow model design guidance, as discussed in the following articles:

- Understand star schema and the importance for Power BI
- All relationship guidance articles found in the Power BI guidance documentation

In general, it's often more efficient to enforce RLS filters on dimension tables, and not fact tables. And, rely on well-designed relationships to ensure RLS filters propagate to other model tables. RLS filters only propagate through active relationships. So, avoid using the LOOKUPVALUE DAX function when model relationships could achieve the same result.

Whenever RLS filters are enforced on DirectQuery tables and there are relationships to other DirectQuery tables, be sure to optimize the source database. It can involve designing appropriate indexes or using persisted computed columns. For more information, see DirectQuery model guidance in Power BI Desktop.

Measure RLS impact

It's possible to measure the performance impact of RLS filters in Power BI Desktop by using Performance Analyzer. First, determine report visual query durations when RLS

isn't enforced. Then, use the **View As** command on the **Modeling** ribbon tab to enforce RLS and determine and compare query durations.

Configure role mappings

Once published to Power BI, you must map members to semantic model roles. Only semantic model owners or workspace admins can add members to roles. For more information, see Row-level security (RLS) with Power BI (Manage security on your model).

Members can be user accounts, security groups, distribution groups or mail enabled groups. Whenever possible, we recommend you map security groups to semantic model roles. It involves managing security group memberships in Microsoft Entra ID. Possibly, it delegates the task to your network administrators.

Validate roles

Test each role to ensure it filters the model correctly. It's easily done by using the **View As** command on the **Modeling** ribbon tab.

When the model has dynamic rules using the USERNAME DAX function, be sure to test for expected *and unexpected* values. When embedding Power BI content—specifically using the embed for your customers scenario—app logic can pass any value as an effective identity user name. Whenever possible, ensure accidental or malicious values result in filters that return no rows.

Consider an example using Power BI embedded, where the app passes the user's job role as the effective user name: It's either "Manager" or "Worker". Managers can see all rows, but workers can only see rows where the `Type` column value is *Internal*.

The following rule expression is defined:

DAX

```
IF(
    USERNAME() = "Worker",
    [Type] = "Internal",
    TRUE()
)
```

The problem with this rule expression is that all values, except *Worker*, return all table rows. So, an accidental value, like *Wrker*, unintentionally returns all table rows. Therefore,

it's safer to write an expression that tests for each expected value. In the following improved rule expression, an unexpected value results in the table returning no rows.

DAX

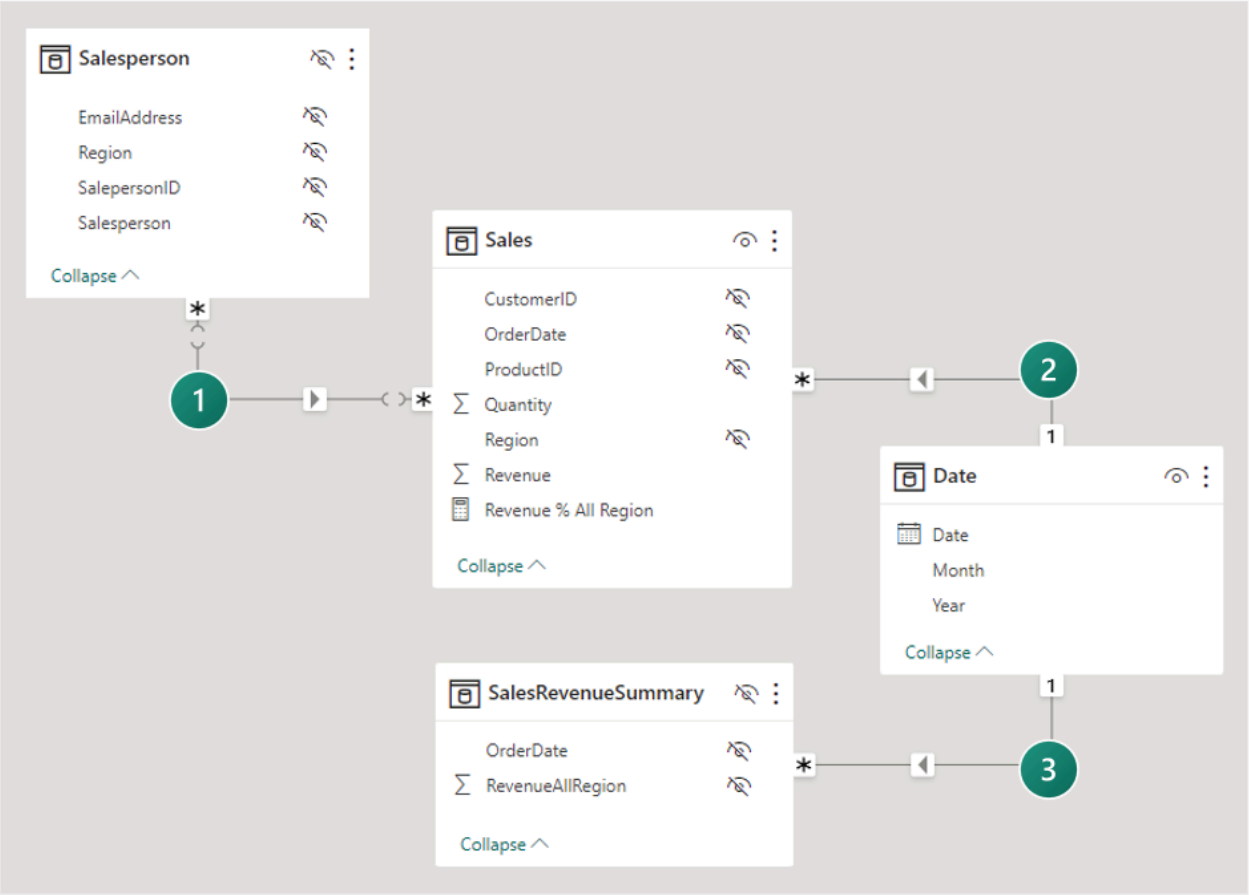
```
IF(
    USERNAME() = "Worker",
    [Type] = "Internal",
    IF(
        USERNAME() = "Manager",
        TRUE(),
        FALSE()
    )
)
```

Design partial RLS

Sometimes, calculations need values that aren't constrained by RLS filters. For example, a report may need to display a ratio of revenue earned for the report user's sales region over *all revenue earned*.

While it's not possible for a DAX expression to override RLS—in fact, it can't even determine that RLS is enforced—you can use a summary model table. The summary model table is queried to retrieve revenue for "all regions" and it's not constrained by any RLS filters.

Let's see how you could implement this design requirement. First, consider the following model design:



The model comprises four tables:

- The `Salesperson` table stores one row per salesperson. It includes the `EmailAddress` column, which stores the email address for each salesperson. This table is hidden.
- The `Sales` table stores one row per order. It includes the `Revenue % All Region` measure, which is designed to return a ratio of revenue earned by the report user's region over revenue earned by all regions.
- The `Date` table stores one row per date and allows filtering and grouping year and month.
- The `SalesRevenueSummary` is a calculated table. It stores total revenue for each order date. This table is hidden.

The following expression defines the `SalesRevenueSummary` calculated table:

DAX

```
SalesRevenueSummary =
SUMMARIZECOLUMNS(
    Sales[OrderDate],
    "RevenueAllRegion", SUM(Sales[Revenue])
)
```

> **Note**
>
> An aggregation table could achieve the same design requirement.

The following RLS rule is applied to the `Salesperson` table:

```DAX
[EmailAddress] = USERNAME()
```

Each of the three model relationships is described in the following table:

Expand table

Relationship	Description
1	There's a many-to-many relationship between the `Salesperson` and `Sales` tables. The RLS rule filters the `EmailAddress` column of the hidden `Salesperson` table by using the USERNAME DAX function. The `Region` column value (for the report user) propagates to the `Sales` table.
2	There's a one-to-many relationship between the `Date` and `Sales` tables.
3	There's a one-to-many relationship between the `Date` and `SalesRevenueSummary` tables.

The following expression defines the `Revenue % All Region` measure:

```DAX
Revenue % All Region =
DIVIDE(
    SUM(Sales[Revenue]),
    SUM(SalesRevenueSummary[RevenueAllRegion])
)
```

> **Note**
>
> Take care to avoid disclosing sensitive facts. If there are only two regions in this example, then it would be possible for a report user to calculate revenue for the other region.

When to avoid using RLS

Sometimes it makes sense to avoid using RLS. If you have only a few simplistic RLS rules that apply static filters, consider publishing multiple semantic models instead. None of the semantic models define roles because each semantic model contains data for a specific report user audience, which has the same data permissions. Then, create one workspace per audience and assign access permissions to the workspace or app.

For example, a company that has just two sales regions decides to publish a semantic model *for each sales region* to different workspaces. The semantic models don't enforce RLS. They do, however, use query parameters to filter source data. This way, the same model is published to each workspace—they just have different semantic model parameter values. Salespeople are assigned access to just one of the workspaces (or published apps).

There are several advantages associated with avoiding RLS:

- **Improved query performance**: It can result in improved performance due to fewer filters.
- **Smaller models**: While it results in more models, they're smaller in size. Smaller models can improve query and data refresh responsiveness, especially if the hosting capacity experiences pressure on resources. Also, it's easier to keep model sizes below size limits imposed by your capacity. Lastly, it's easier to balance workloads across different capacities, because you can create workspaces on—or move workspaces to—different capacities.
- **Additional features**: Power BI features that don't work with RLS, like Publish to web, can be used.

However, there are disadvantages associated with avoiding RLS:

- **Multiple workspaces**: One workspace is required for each report user audience. If apps are published, it also means there's one app per report user audience.
- **Duplication of content**: Reports and dashboards must be created in each workspace. It requires more effort and time to set up and maintain.
- **High privilege users**: High privilege users, who belong to multiple report user audiences, can't see a consolidated view of the data. They'll need to open multiple reports (from different workspaces or apps).

Troubleshoot RLS

If RLS produces unexpected results, check for the following issues:

- Incorrect relationships exist between model tables, in terms of column mappings and filter directions. Keep in mind that RLS filters only propagate through active relationships.
- The **Apply security filter in both directions** relationship property isn't correctly set. For more information, see Bi-directional relationship guidance.
- Tables contain no data.
- Incorrect values are loaded into tables.
- The user is mapped to multiple roles.
- The model includes aggregation tables, and RLS rules don't consistently filter aggregations and details. For more information, see Use aggregations in Power BI Desktop (RLS for aggregations).

When a specific user can't see any data, it could be because their UPN isn't stored or it's entered incorrectly. It can happen abruptly because their user account has changed as the result of a name change.

> **Tip**
>
> For testing purposes, add a measure that returns the **USERNAME** DAX function. You might name it something like "Who Am I". Then, add the measure to a card visual in a report and publish it to Power BI.

Creators and consumers with only Read permission on the semantic model will only be able to view the data they're allowed to see (based on their RLS role mapping).

When a user views a report in either a workspace or an app, RLS might or might not be enforced depending on their semantic model permissions. For this reason, it's critical that content consumers and creators only possess Read permission on the underlying semantic model when RLS must be enforced. For details about the permissions rules that determine whether RLS is enforced, see the Report consumer security planning article.

Related content

For more information related to this article, check out the following resources:

- Row-level security (RLS) with Power BI
- Restrict data access with row-level security (RLS) for Power BI Desktop
- Model relationships in Power BI Desktop
- Power BI implementation planning: Report consumer security planning
- Questions? Try asking the Fabric Community

- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI modeling guidance for Power Platform

Article • 12/30/2024

Microsoft Dataverse is the standard data platform for many Microsoft business application products, including Dynamics 365 Customer Engagement and Power Apps canvas apps, and also Dynamics 365 Customer Voice (formerly Microsoft Forms Pro), Power Automate approvals, Power Apps portals, and others.

This article provides guidance on how to create a Power BI data model that connects to Dataverse. It describes differences between a Dataverse schema and an optimized Power BI schema, and it provides guidance for expanding the visibility of your business application data in Power BI.

Because of its ease of setup, rapid deployment, and widespread adoption, Dataverse stores and manages an increasing volume of data in environments across organizations. That means there's an even greater need—and opportunity—to integrate analytics with those processes. Opportunities include:

- Report on all Dataverse data moving beyond the constraints of the built-in charts.
- Provide easy access to relevant, contextually filtered reports within a specific record.
- Enhance the value of Dataverse data by integrating it with external data.
- Take advantage of Power BI's built-in artificial intelligence (AI) without the need to write complex code.
- Increase adoption of Power Platform solutions by increasing their usefulness and value.
- Deliver the value of the data in your app to business decision makers.

Connect Power BI to Dataverse

Connecting Power BI to Dataverse involves creating a Power BI data model. You can choose from three methods to create a Power BI model.

- **Import Dataverse data by using the Dataverse connector**: This method caches (stores) Dataverse data in a Power BI model. It delivers fast performance thanks to in-memory querying. It also offers design flexibility to modelers, allowing them to integrate data from other sources. Because of these strengths, importing data is the default mode when creating a model in Power BI Desktop.

- **Import Dataverse data by using Azure Synapse Link**: This method is a variation on the import method, because it also caches data in the Power BI model, but does so by connecting to Azure Synapse Analytics. By using Azure Synapse Link for Dataverse, Dataverse tables are continuously replicated to Azure Synapse or Azure Data Lake Storage (ADLS) Gen2. This approach is used to report on hundreds of thousands or even millions of records in Dataverse environments.
- **Create a DirectQuery connection by using the Dataverse connector**: This method is an alternative to importing data. A DirectQuery model consists only of metadata defining the model structure. When a user opens a report, Power BI sends native queries to Dataverse to retrieve data. Consider creating a DirectQuery model when reports must show near real-time Dataverse data, or when Dataverse must enforce role-based security so that users can only see the data they have privileges to access.

> **Important**
>
> While a DirectQuery model can be a good alternative when you need near real-time reporting or enforcement of Dataverse security in a report, it can result in slow performance for that report.
>
> You can learn about considerations for DirectQuery later in this article.

To determine the right method for your Power BI model, you should consider:

- Query performance
- Data volume
- Data latency
- Role-based security
- Setup complexity

> **Tip**
>
> For a detailed discussion on model frameworks (import, DirectQuery, or composite), their benefits and limitations, and features to help optimize Power BI data models, see Choose a Power BI model framework.

Query performance

Queries sent to import models are faster than native queries sent to DirectQuery data sources. That's because imported data is cached in memory and it's optimized for

analytic queries (filter, group, and summarize operations).

Conversely, DirectQuery models only retrieve data from the source after the user opens a report, resulting in seconds of delay as the report renders. Additionally, user interactions on the report require Power BI to requery the source, further reducing responsiveness.

Data volume

When developing an import model, you should strive to minimize the data that's loaded into the model. It's especially true for large models, or models that you anticipate will grow to become large over time. For more information, see Data reduction techniques for import modeling.

A DirectQuery connection to Dataverse is a good choice when the report's query result isn't large. A large query result has more than 20,000 rows in the report's source tables, or the result returned to the report after filters are applied is more than 20,000 rows. In this case, you can create a Power BI report by using the Dataverse connector.

> **Note**
>
> The 20,000 row size isn't a hard limit. However, each data source query must return a result within 10 minutes. Later in this article you will learn how to work within those limitations and about other Dataverse DirectQuery design considerations.

You can improve the performance of larger semantic models by using the Dataverse connector to import the data into the data model.

Even larger semantic models—with several hundreds of thousand or even millions of rows—can benefit from using Azure Synapse Link for Dataverse. This approach sets up an ongoing managed pipeline that copies Dataverse data into ADLS Gen2 as CSV or Parquet files. Power BI can then query an Azure Synapse serverless SQL pool to load an import model.

Data latency

When the Dataverse data changes rapidly and report users need to see up-to-date data, a DirectQuery model can deliver near real-time query results.

> **Tip**

You can create a Power BI report that uses **automatic page refresh** to show real-time updates, but only when the report connects to a DirectQuery model.

Import data models must complete a data refresh to allow reporting on recent data changes. Keep in mind that there are limitations on the number of daily scheduled data refresh operations. You can schedule up to eight refreshes per day on a shared capacity. On a Premium capacity or Microsoft Fabric capacity, you can schedule up to 48 refreshes per day, which can achieve a 15-minute refresh frequency.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

You can also consider using incremental refresh to achieve faster refreshes and near real-time performance (only available with Premium or Fabric).

Role-based security

When there's a need to enforce role-based security, it can directly influence the choice of Power BI model framework.

Dataverse can enforce complex role-based security to control access of specific records to specific users. For example, a salesperson might be permitted to see only their sales opportunities, while the sales manager can see all sales opportunities for all salespeople. You can tailor the level of complexity based on the needs of your organization.

A DirectQuery model based on Dataverse can connect by using the security context of the report user. That way, the report user will only see the data they're permitted to access. This approach can simplify the report design, providing performance is acceptable.

For improved performance, you can create an import model that connects to Dataverse instead. In this case, you can add row-level security (RLS) to the model, if necessary.

> **Note**

It might be challenging to replicate some Dataverse role-based security as Power BI RLS, especially when Dataverse enforces complex permissions. Further, it might require ongoing management to keep Power BI permissions in sync with Dataverse permissions.

For more information about Power BI RLS, see Row-level security (RLS) guidance in Power BI Desktop.

Setup complexity

Using the Dataverse connector in Power BI—whether for import or DirectQuery models—is straightforward and doesn't require any special software or elevated Dataverse permissions. That's an advantage for organizations or departments that are getting started.

The Azure Synapse Link option requires system administrator access to Dataverse and certain Azure permissions. These Azure permissions are required to set up the storage account and a Synapse workspace.

Recommended practices

This section describes design patterns (and anti-patterns) you should consider when creating a Power BI model that connects to Dataverse. Only a few of these patterns are unique to Dataverse, but they tend to be common challenges for Dataverse makers when they go about building Power BI reports.

Focus on a specific use case

Rather than trying to solve *everything*, focus on the specific use case.

This recommendation is probably the most common and easily the most challenging anti-pattern to avoid. Attempting to build a single model that achieves all self-service reporting needs is challenging. The reality is that successful models are built to answer questions around a central set of facts over a single core topic. While that might initially seem to limit the model, it's actually empowering because you can tune and optimize the model for answering questions within that topic.

To help ensure that you have a clear understanding of the model's purpose, ask yourself the following questions.

- What topic area will this model support?

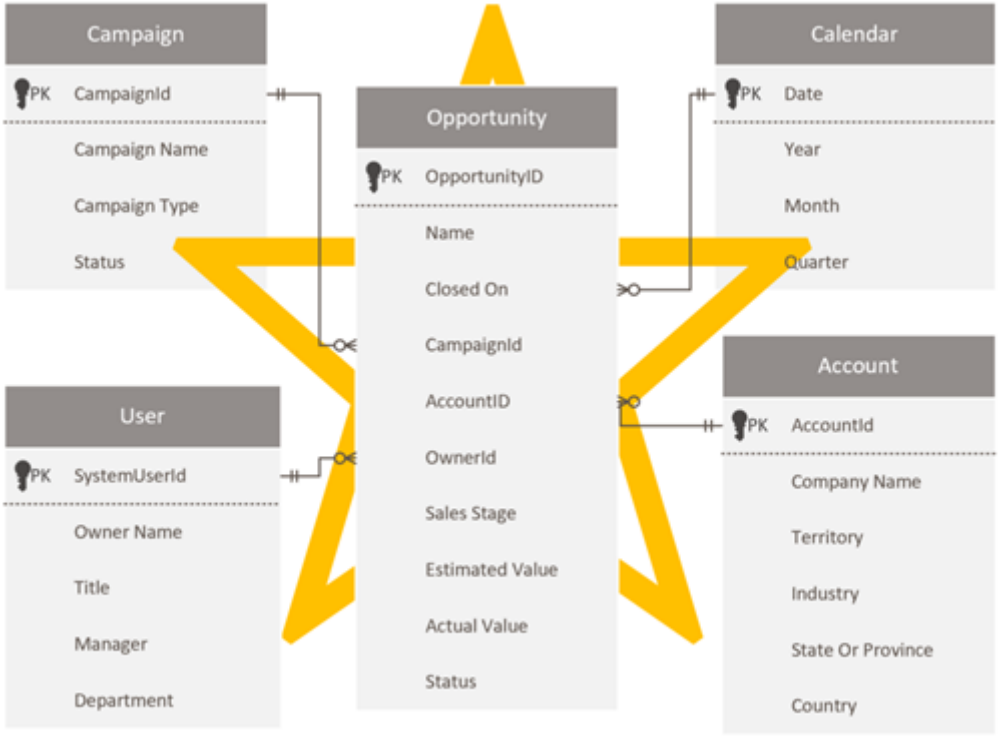
- Who is the audience of the reports?
- What questions are the reports trying to answer?
- What is the minimum viable semantic model?

Resist combining multiple topic areas into a single model just because the report user has questions across multiple topic areas that they want addressed by a single report. By breaking that report out into multiple reports, each with a focus on a different topic (or fact table), you can produce much more efficient, scalable, and manageable models.

Design a star schema

Dataverse developers and administrators who are comfortable with the Dataverse schema might be tempted to reproduce the same schema in Power BI. This approach is an anti-pattern, and it's probably the toughest to overcome because it just *feels right* to maintain consistency.

Dataverse, as a relational model, is well suited for its purpose. However, it's not designed as an analytic model that's optimized for analytical reports. The most prevalent pattern for modeling analytics data is a *star schema* design. Star schema is a mature modeling approach widely adopted by relational data warehouses. It requires modelers to classify their model tables as either dimension or fact. Reports can filter or group by using dimension table columns and summarize fact table columns.



For more information, see Understand star schema and the importance for Power BI.

Optimize Power Query queries

The Power Query mashup engine strives to achieve *query folding* whenever possible for reasons of efficiency. A query that achieves folding delegates query processing to the source system.

The source system, in this case Dataverse, then only needs to deliver filtered or summarized results to Power BI. A folded query is often significantly faster and more efficient than a query that doesn't fold.

For more information on how you can achieve query folding, see Power Query query folding.

> **Note**
>
> Optimizing Power Query is a broad topic. To achieve a better understanding of what Power Query is doing at authoring and at model refresh time in Power BI Desktop, see **Query diagnostics**.

Minimize the number of query columns

By default, when you use Power Query to load a Dataverse table, it retrieves all rows and all columns. When you query a system user table, for example, it could contain more than 1,000 columns. The columns in the metadata include relationships to other entities and lookups to option labels, so the total number of columns grows with the complexity of the Dataverse table.

Attempting to retrieve data from all columns is an anti-pattern. It often results in extended data refresh operations, and it will cause the query to fail when the time needed to return the data exceeds 10 minutes.

We recommend that you only retrieve columns that are required by reports. It's often a good idea to reevaluate and refactor queries when report development is complete, allowing you to identify and remove unused columns. For more information, see Data reduction techniques for import modeling (Remove unnecessary columns).

Additionally, ensure that you introduce the Power Query *Remove columns* step early so that it folds back to the source. That way, Power Query can avoid the unnecessary work of extracting source data only to discard it later (in an unfolded step).

When you have a table that contains many columns, it might be impractical to use the Power Query interactive query builder. In this case, you can start by creating a blank query. You can then use the Advanced Editor to paste in a minimal query that creates a starting point.

Consider the following query that retrieves data from just two columns of the `account` table.

Power Query M

```
let
    Source = CommonDataService.Database("demo.crm.dynamics.com",
[CreateNavigationProperties=false]),
    dbo_account = Source{[Schema="dbo", Item="account"]}[Data],
    #"Removed Other Columns" = Table.SelectColumns(dbo_account,
{"accountid", "name"})
in
    #"Removed Other Columns"
```

Write native queries

When you have specific transformation requirements, you might achieve better performance by using a native query written in Dataverse SQL, which is a subset of Transact-SQL. You can write a native query to:

- Reduce the number of rows (by using a `WHERE` clause).
- Aggregate data (by using the `GROUP BY` and `HAVING` clauses).
- Join tables in a specific way (by using the `JOIN` or `APPLY` syntax).
- Use supported SQL functions.

For more information, see:

- Use SQL to query data
- How Dataverse SQL differs from Transact-SQL

Execute native queries with the EnableFolding option

Power Query executes a native query by using the `Value.NativeQuery` function.

When using this function, it's important to add the `EnableFolding=true` option to ensure queries are folded back to the Dataverse service. A native query won't fold unless this option is added. Enabling this option can result in significant performance improvements—up to 97 percent faster in some cases.

Consider the following query that uses a native query to source selected columns from the `account` table. The native query will fold because the `EnableFolding=true` option is set.

Power Query M

```
let
    Source = CommonDataService.Database("demo.crm.dynamics.com"),
    dbo_account = Value.NativeQuery(
        Source,
        "SELECT A.accountid, A.name FROM account A"
        ,null
        ,[EnableFolding=true]
    )
in
     dbo_account
```

You can expect to achieve the greatest performance improvements when retrieving a subset of data from a large data volume.

> **Tip**
>
> Performance improvement can also depend on how Power BI queries the source database. For example, a measure that uses the `COUNTDISTINCT` DAX function showed almost no improvement with or without the folding hint. When the measure formula was rewritten to use the `SUMX` DAX function, the query folded resulting in a 97 percent improvement over the same query without the hint.

For more information, see Value.NativeQuery. (The `EnableFolding` option isn't documented because it's specific to only certain data sources.)

Speed up the evaluation stage

If you're using the Dataverse connector (formerly known as the Common Data Service), you can add the `CreateNavigationProperties=false` option to speed up the evaluation stage of a data import.

The evaluation stage of a data import iterates through the metadata of its source to determine all possible table relationships. That metadata can be extensive, especially for Dataverse. By adding this option to the query, you're letting Power Query know that you don't intend to use those relationships. The option allows Power BI Desktop to skip that stage of the refresh and move on to retrieving the data.

> **Note**
>
> Don't use this option when the query depends on any expanded relationship columns.

Consider an example that retrieves data from the `account` table. It contains three columns related to territory: `territory`, `territoryid`, and `territoryidname`.

territory	territoryid	territoryidname
Value	2CFF3CA5-D429-EB11-A813-...	Arizona
Value	2EFF3CA5-D429-EB11-A813-0...	California
Value	48FE3CA5-D429-EB11-A813-...	Poland
Value	68FD3CA5-D429-EB11-A813-...	Japan
Value	2EFF3CA5-D429-EB11-A813-0...	California
Value	70FE3CA5-D429-EB11-A813-...	Sweden
Value	12FF3CA5-D429-EB11-A813-...	Kentucky
Value	2CFF3CA5-D429-EB11-A813-...	Arizona

When you set the `CreateNavigationProperties=false` option, the `territoryid` and `territoryidname` columns will remain, but the `territory` column, which is a relationship column (it shows *Value* links), will be excluded. It's important to understand that Power Query relationship columns are a different concept to *model relationships*, which propagate filters between model tables.

Consider the following query that uses the `CreateNavigationProperties=false` option (in the *Source* step) to speed up the evaluation stage of a data import.

Power Query M

```
let
    Source = CommonDataService.Database("demo.crm.dynamics.com"
        ,[CreateNavigationProperties=false]),
    dbo_account = Source{[Schema="dbo", Item="account"]}[Data],
    #"Removed Other Columns" = Table.SelectColumns(dbo_account,
{"accountid", "name", "address1_stateorprovince", "address1_country",
"industrycodename", "territoryidname"}),
    #"Renamed Columns" = Table.RenameColumns(#"Removed Other Columns",
{{"name", "Account Name"}, {"address1_country", "Country"},
{"address1_stateorprovince", "State or Province"}, {"territoryidname",
"Territory"}, {"industrycodename", "Industry"}})
in
    #"Renamed Columns"
```

When using this option, you're likely to experience significant performance improvement when a Dataverse table has many relationships to other tables. For example, because the `SystemUser` table is related to every other table in the database, refresh performance of this table would benefit by setting the `CreateNavigationProperties=false` option.

> **Note**
>
> This option can improve the performance of data refresh of import tables or dual storage mode tables, including the process of applying Power Query Editor window changes. It doesn't improve the performance of interactive cross-filtering of DirectQuery storage mode tables.

Resolve blank choice labels

If you discover that Dataverse choice labels are blank in Power BI, it could be because the labels haven't been published to the Tabular Data Stream (TDS) endpoint.

In this case, open the Dataverse Maker Portal, navigate to the **Solutions** area, and then select **Publish all customizations**. The publication process will update the TDS endpoint with the latest metadata, making the option labels available to Power BI.

Larger semantic models with Azure Synapse Link

Dataverse includes the ability to synchronize tables to Azure Data Lake Storage (ADLS) and then connect to that data through an Azure Synapse workspace. With minimal effort, you can set up Azure Synapse Link to populate Dataverse data into Azure Synapse and enable data teams to discover deeper insights.

Azure Synapse Link enables a continuous replication of the data and metadata from Dataverse into the data lake. It also provides a built-in serverless SQL pool as a convenient data source for Power BI queries.

The strengths of this approach are significant. Customers gain the ability to run analytics, business intelligence, and machine learning workloads across Dataverse data by using various advanced services. Advanced services include Apache Spark, Power BI, Azure Data Factory, Azure Databricks, and Azure Machine Learning.

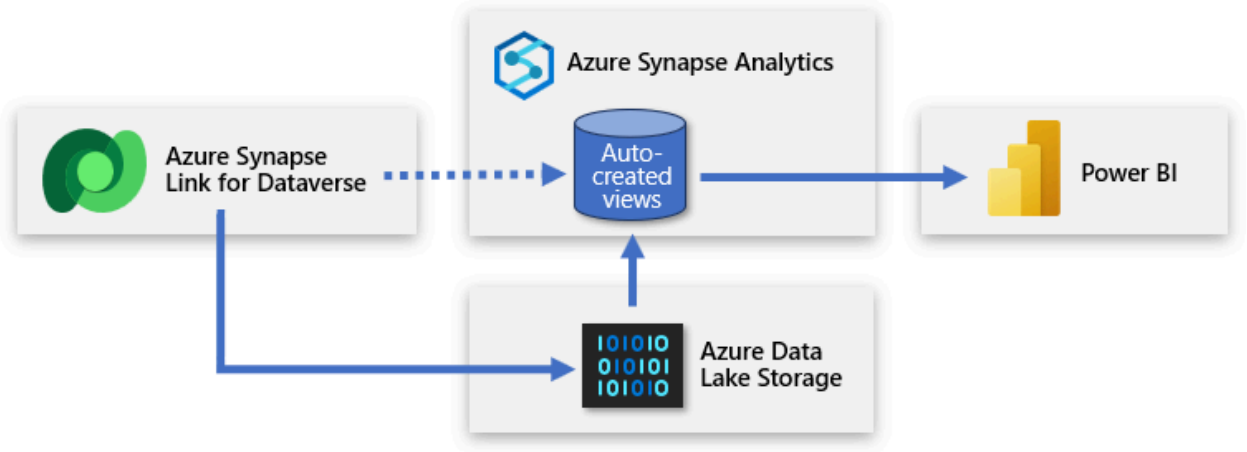
Create an Azure Synapse Link for Dataverse

To create an Azure Synapse Link for Dataverse, you'll need the following prerequisites in place.

- System administrator access to the Dataverse environment.
- For the Azure Data Lake Storage:

    - You must have a storage account to use with ADLS Gen2.
    - You must be assigned **Storage Blob Data Owner** and **Storage Blob Data Contributor** access to the storage account. For more information, see Role-based access control (Azure RBAC).
    - The storage account must enable hierarchical namespace.
    - It's recommended that the storage account use read-access geo-redundant storage (RA-GRS).
- For the Synapse workspace:
    - You must have access to a Synapse workspace and be assigned **Synapse Administrator** access. For more information, see Built-in Synapse RBAC roles and scopes.
    - The workspace must be in the same region as the ADLS Gen2 storage account.

The setup involves signing in to Power Apps and connecting Dataverse to the Azure Synapse workspace. A wizard-like experience allows you to create a new link by selecting the storage account and the tables to export. Azure Synapse Link then copies data to the ADLS Gen2 storage and automatically creates views in the built-in Azure Synapse serverless SQL pool. You can then connect to those views to create a Power BI model.

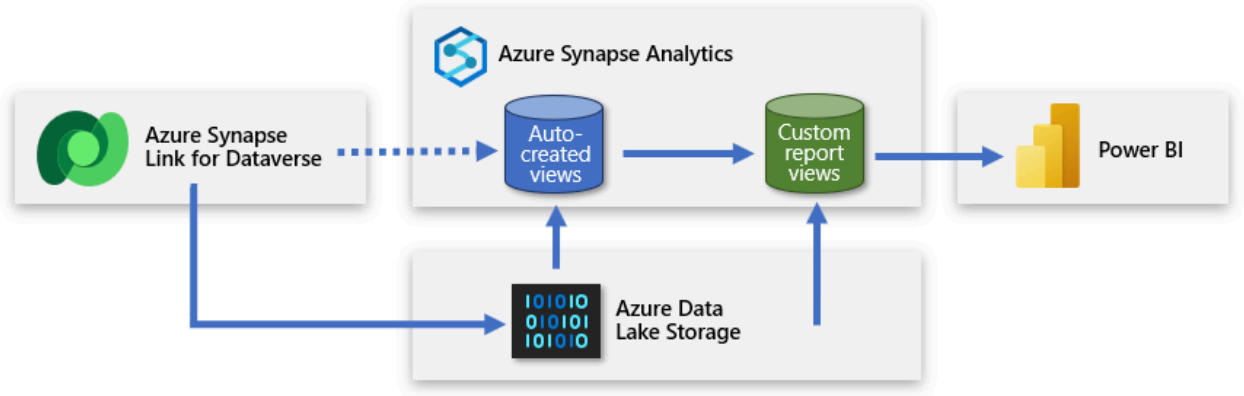


> **Tip**
>
> For complete documentation on creating, managing, and monitoring Azure Synapse Link see **Create an Azure Synapse Link for Dataverse with your Azure Synapse Workspace**.

Create a second serverless SQL database

You can create a second serverless SQL database and use it to add custom report views. That way, you can present a simplified set of data to the Power BI creator that allows

them to create a model based on useful and relevant data. The new serverless SQL database becomes the creator's primary source connection and a friendly representation of the data sourced from the data lake.



This approach delivers data to Power BI that's focused, enriched, and filtered.

You can create a serverless SQL database in the Azure Synapse workspace by using Azure Synapse Studio. Select **Serverless** as the SQL database type and enter a database name. Power Query can connect to this database by connecting to the workspace SQL endpoint.

Create custom views

You can create custom views that wrap serverless SQL pool queries. These views will serve as straightforward, clean sources of data that Power BI connects to. The views should:

- Include the labels associated with choice fields.
- Reduce complexity by including only the columns required for data modeling.
- Filter out unnecessary rows, such as inactive records.

Consider the following view that retrieves campaign data.

SQL

```
CREATE VIEW [VW_Campaign]
AS
    SELECT
        [base].[campaignid] AS [CampaignID]
        [base].[name] AS [Campaign],
        [campaign_status].[LocalizedLabel] AS [Status],
        [campaign_typecode].[LocalizedLabel] AS [Type Code]
    FROM
        [<MySynapseLinkDB>].[dbo].[campaign] AS [base]
        LEFT OUTER JOIN [<MySynapseLinkDB>].[dbo].[OptionsetMetadata] AS
[campaign_typecode]
            ON [base].[typecode] = [campaign_typecode].[option]
```

```
                AND [campaign_typecode].[LocalizedLabelLanguageCode] = 1033
                AND [campaign_typecode].[EntityName] = 'campaign'
                AND [campaign_typecode].[OptionSetName] = 'typecode'
        LEFT OUTER JOIN [<MySynapseLinkDB>].[dbo].[StatusMetadata] AS
[campaign_status]
            ON [base].[statuscode] = [campaign_Status].[status]
                AND [campaign_status].[LocalizedLabelLanguageCode] = 1033
                AND [campaign_status].[EntityName] = 'campaign'
    WHERE
        [base].[statecode] = 0;
```

Notice that the view includes only four columns, each aliased with a friendly name. There's also a `WHERE` clause to return only necessary rows, in this case active campaigns. Also, the view queries the campaign table that's joined to the `OptionsetMetadata` and `StatusMetadata` tables, which retrieve choice labels.

> **Tip**
>
> For more information on how to retrieve metadata, see Access choice labels directly from Azure Synapse Link for Dataverse.

Query appropriate tables

Azure Synapse Link for Dataverse ensures that data is continually synchronized with the data in the data lake. For high-usage activity, simultaneous writes and reads can create locks that cause queries to fail. To ensure reliability when retrieving data, two versions of the table data are synchronized in Azure Synapse.

- **Near real-time data**: Provides a copy of data synchronized from Dataverse via Azure Synapse Link in an efficient manner by detecting what data has changed since it was initially extracted or last synchronized.
- **Snapshot data**: Provides a read-only copy of near real-time data that's updated at regular intervals (in this case every hour). Snapshot data table names have *_partitioned* appended to their name.

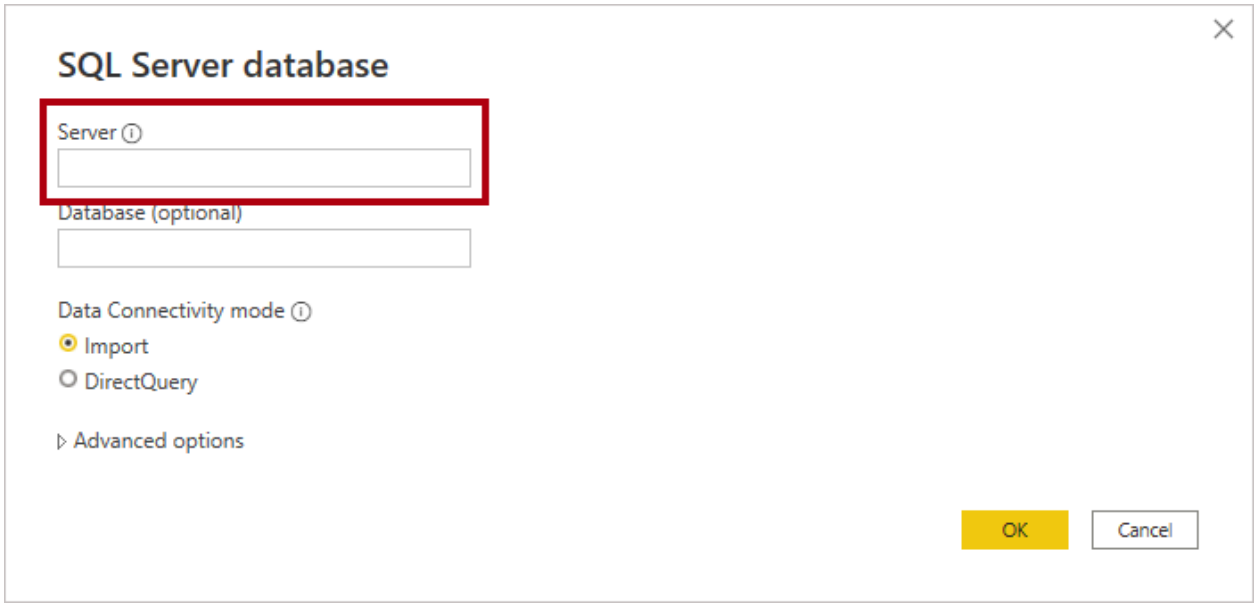
If you anticipate that a high volume of read and write operations will be executed simultaneously, retrieve data from the snapshot tables to avoid query failures.

For more information, see Access near real-time data and read-only snapshot data.

Connect to Synapse Analytics

To query an Azure Synapse serverless SQL pool, you'll need its workspace SQL endpoint. You can retrieve the endpoint from Synapse Studio by opening the serverless SQL pool properties.

In Power BI Desktop, you can connect to Azure Synapse by using the Azure Synapse Analytics SQL connector. When prompted for the server, enter the workspace SQL endpoint.



Considerations for DirectQuery

There are many use cases when using DirectQuery storage mode can solve your requirements. However, using DirectQuery can negatively affect Power BI report performance. A report that uses a DirectQuery connection to Dataverse won't be as fast as a report that uses an import model. Generally, you should import data to Power BI whenever possible.

We recommend that you consider the topics in this section when working with DirectQuery.

For more information about determining when to work with DirectQuery storage mode, see Choose a Power BI model framework.

Use dual storage mode dimension tables

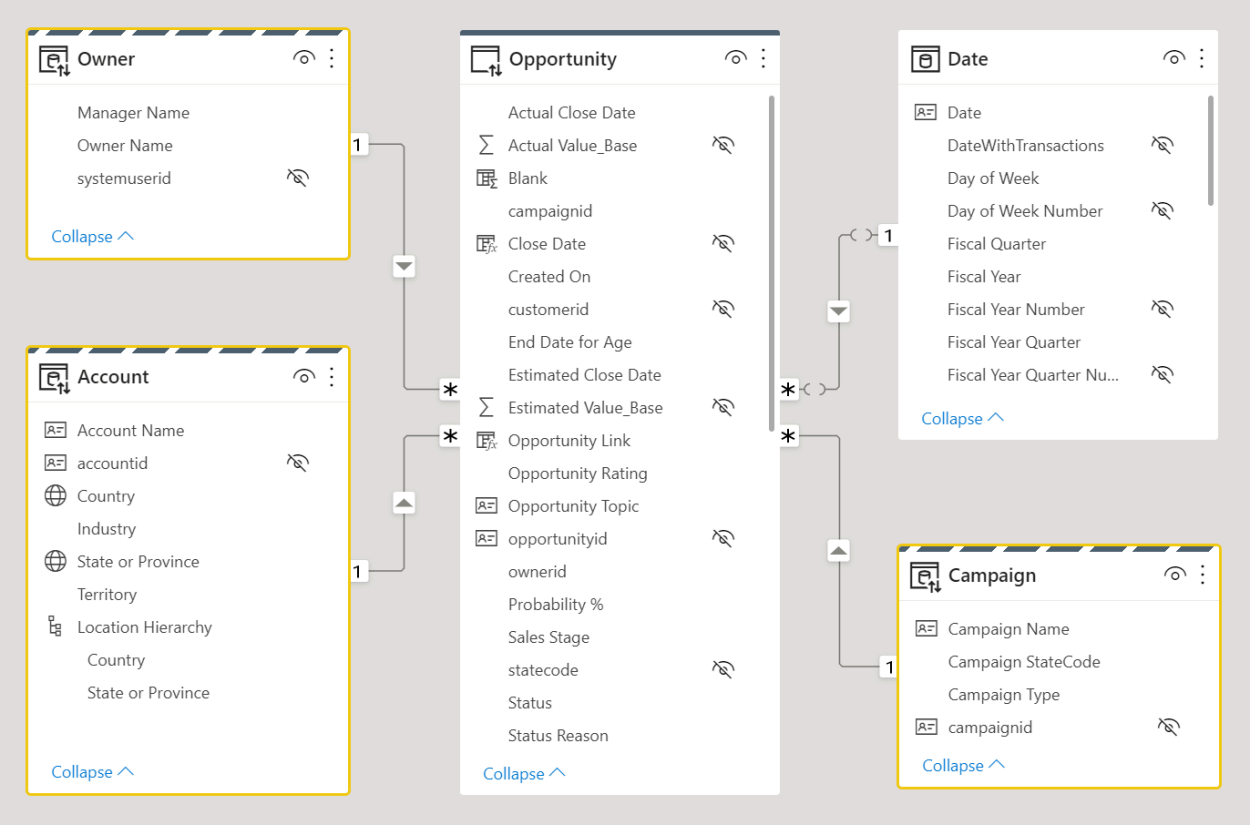
A dual storage mode table is set to use both import and DirectQuery storage modes. At query time, Power BI determines the most efficient mode to use. Whenever possible, Power BI attempts to satisfy queries by using imported data because it's faster.

You should consider setting dimension tables to dual storage mode, when appropriate. That way, slicer visuals and filter card lists—which are often based on dimension table columns—will render more quickly because they'll be queried from imported data.

> **Important**
>
> When a dimension table needs to inherit the Dataverse security model, it isn't appropriate to use dual storage mode.

Fact tables, which typically store large volumes of data, should remain as DirectQuery storage mode tables. They'll be filtered by the related dual storage mode dimension tables, which can be joined to the fact table to achieve efficient filtering and grouping.

Consider the following data model design. Three dimension tables, `Owner`, `Account`, and `Campaign` have a striped upper border, which means they're set to dual storage mode.



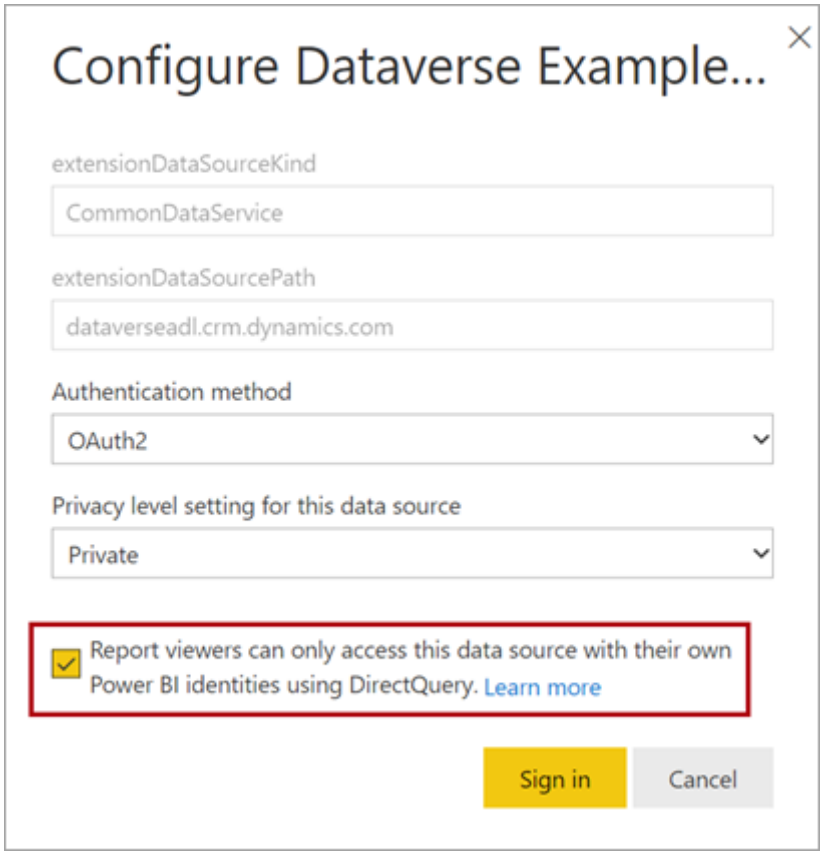
For more information on table storage modes including dual storage, see Manage storage mode in Power BI Desktop.

Enable single-sign on

When you publish a DirectQuery model to the Power BI service, you can use the semantic model settings to enable single sign-on (SSO) by using Microsoft Entra ID

OAuth2 for your report users. You should enable this option when Dataverse queries must execute in the security context of the report user.

When the SSO option is enabled, Power BI sends the report user's authenticated Microsoft Entra credentials in the queries to Dataverse. This option enables Power BI to honor the security settings that are set up in the data source.

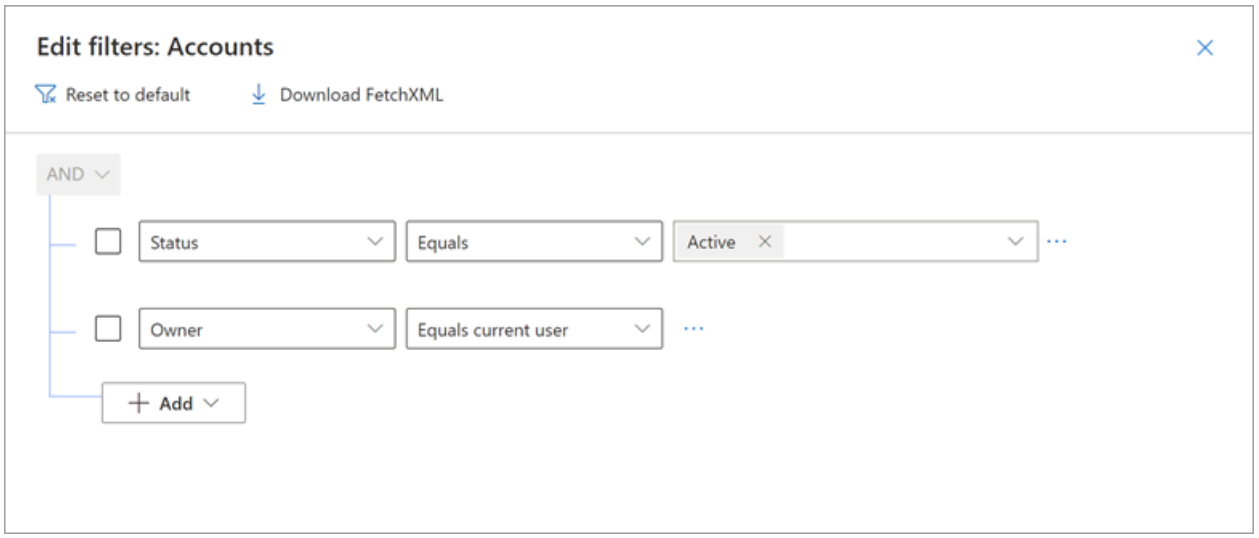


For more information, see Single sign-on (SSO) for DirectQuery sources.

Replicate "My" filters in Power Query

When using Microsoft Dynamics 365 Customer Engagement (CE) and model-driven Power Apps built on Dataverse, you can create views that show only records where a username field, like `Owner`, equals the current user. For example, you might create views named "My open opportunities", "My active cases", and others.

Consider an example of how the Dynamics 365 *My Active Accounts* view includes a filter where *Owner equals current user*.



You can reproduce this result in Power Query by using a native query that embeds the `CURRENT_USER` token.

Consider the following example that shows a native query that returns the accounts for the current user. In the `WHERE` clause, notice that the `ownerid` column is filtered by the `CURRENT_USER` token.

Power Query M

```
let
    Source = CommonDataService.Database("demo.crm.dynamics.com",
[CreateNavigationProperties=false],
    dbo_account = Value.NativeQuery(Source, "
        SELECT
            accountid, accountnumber, ownerid, address1_city,
address1_stateorprovince, address1_country
        FROM account
        WHERE statecode = 0
            AND ownerid = CURRENT_USER
    ", null, [EnableFolding]=true])
in
    dbo_account
```

When you publish the model to the Power BI service, you must enable single sign-on (SSO) so that Power BI will send the report user's authenticated Microsoft Entra credentials to Dataverse.

Create supplementary import models

You can create a DirectQuery model that enforces Dataverse permissions *knowing* that performance will be slow. You can then supplement this model with import models that target specific subjects or audiences that could enforce RLS permissions.

For example, an import model could provide access to all Dataverse data but not enforce any permissions. This model would be suited to executives who already have access to all Dataverse data.

As another example, when Dataverse enforces role-based permissions by sales region, you could create one import model and replicate those permissions using RLS. Alternatively, you could create a model for each sales region. You could then grant read permission to those models (semantic models) to the salespeople of each region. To facilitate the creation of these regional models, you can use parameters and report templates. For more information, see Create and use report templates in Power BI Desktop.

Related content

For more information related to this article, check out the following resources.

- Azure Synapse Link for Dataverse
- Understand star schema and the importance for Power BI
- Data reduction techniques for Import modeling
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Migrate from Azure Analysis Services to Power BI Premium

Article • 12/30/2024

This article targets Azure Analysis Services (AAS) data modelers and administrators. It provides them with guidance and rationale to help migrate their AAS databases to Power BI Premium or Power BI Embedded.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Background

Power BI has evolved into the leading platform for both self-service and IT-managed enterprise business intelligence (BI). With exponential growth in data volumes and complexity, Power BI customers demand enterprise BI solutions that scale to petabytes, are secure, easy to manage, and accessible to all users across the largest of organizations.

For over two decades, Microsoft has continued to make deep investments in enterprise BI. AAS and SQL Server Analysis Services (SSAS) are based on mature BI data modeling technology used by countless enterprises. Today, that same technology is also at the heart of Power BI semantic models.

> **Note**
>
> In this article, the terms data model, BI model, tabular model, database, and Power BI semantic model have the same meaning. This article commonly uses the terms *data model* for AAS model and *semantic model* for Power BI model.

Also, while this article describes the process of migrating to Power BI Premium, it also applies to Power BI Embedded.

In recent years, Microsoft has taken great strides to deliver AAS capabilities to Power BI Premium. To that end, Power BI instantly inherited a large ecosystem of developers, partners, BI tools, and solutions that were built up over decades. Today, the full set of Power BI Premium workloads, features, and capabilities now results in a modern, cloud BI platform that goes far beyond comparable functionality available in AAS or SSAS.

Today, many customers have Power BI reports that live connect to AAS. Naturally, these customers are asking whether there's an opportunity to consolidate by hosting their data models alongside their reports in Power BI. They often ask questions like:

- Does all the AAS functionality we depend on work in Power BI?
- Is Power BI backwards compatible with AAS tools and processes?
- What capabilities are available only in Power BI?
- How do we compare costs between AAS and Power BI?
- Why is Microsoft converging enterprise and self-service BI?
- How do we migrate from AAS to Power BI Premium?
- Is AAS marked for deprecation?
- What's Microsoft's roadmap for enterprise data models?

Answers to many of these questions are described in this article.

> **Note**
>
> The decision to migrate to Power BI Premium depends on the requirements of each customer. Customers should carefully evaluate additional benefits in order to make an informed decision. We expect to see organic migration to Power BI Premium over time, and our intention is that it happens on terms that the customer is comfortable with.
>
> To be clear, currently there aren't any plans to deprecate AAS. There is a priority to focus investment on Power BI Premium for enterprise data modeling, and so the additional value provided by Power BI Premium will increase over time. Customers who choose Power BI Premium can expect to benefit from alignment with the Microsoft BI product roadmap.

Convergence of self-service and enterprise BI

Consolidation of items (like reports and dashboards) in Power BI results in simplified discovery and management due to co-location. Once consolidated, there's no need to bridge the gap between AAS and Power BI. Central IT teams can then more easily adopt self-service items that have become popular yet are resulting in a management burden for the business. IT can take over such items. They can operationalize them for mission-critical decision making based on governed data that's aligned with corporate standards and with lineage transparency. Simplifying this workflow by sharing a common platform promotes better collaboration between the business and IT.

Power BI Premium

Thanks to its distributed architecture, Power BI Premium is less sensitive to overall load, temporal spikes, and high concurrency. By consolidating capacities to larger Power BI Premium SKUs, customers can achieve increased performance and throughput.

Scalability benefits associated with Power BI Premium are described later in this article.

Feature comparison

AAS provides the Analysis Services database engine for hosting data models, which is a core component of a Microsoft enterprise BI architecture. In fact, Power BI Premium is a superset of AAS because it provides much more functionality. The following table lists features supported in AAS and Power BI Premium. The table focuses on - but isn't limited to - Power BI semantic model-related capabilities.

Expand table

Feature	AAS	Power BI Premium
Premium workloads		
Paginated reports, which are ideal for reports that are designed to be printed, especially when table data overflows to multiple pages	No	Yes
Dataflows, which store fragments of data intended for use in a Power BI semantic model	No	Yes
AI with dataflows, which use artificial intelligence (AI) with Cognitive Services, Automated Machine Learning, and Azure Machine Learning (AML) integration	No	Yes
Metrics, which curate key business measures and allow tracking them against objectives	No	Yes

Feature	AAS	Power BI Premium
Business enablement		
Unlimited report distribution to anyone (even outside the organization)	No	Yes
Business-driven interactive reports, workspaces, and apps	No	Yes
Platform scalability and resiliency		
Power BI Premium architecture, which supports increased scale and performance	No	Yes
Optimized semantic model memory management	No	Yes
Scale limits per data model instead of per server	No	Yes
CPU smoothing for refresh resiliency	No	Yes
Autoscale, which automatically adds compute capacity to avoid slowdowns under heavy use	No	Yes
Business continuity and disaster recovery (BCDR) with Azure regions and availability zones	No	Yes
Interactive analysis over big data		
Large model sizes (up to 400 GB with compression)	Yes	Yes
Hybrid tables, which comprise in-memory and DirectQuery partitions that can help deliver near real-time results over large tables	No	Yes
Automatic aggregations, which use state-of-the-art machine learning (ML) to continuously optimize DirectQuery performance	No	Yes
User-defined aggregations, which can improve query performance over very large DirectQuery tables	No	Yes
Query scale-out, which distributes client queries among replicated servers	Yes	Yes
Security		
Bring Your Own Key (BYOK), which allows customers to use their own encryption key to encrypt data stored in the Microsoft cloud	No	Yes
Virtual network connectivity, which allows Power BI to work seamlessly in an organization's virtual network (VNet)	No	Yes
Azure Private Link, which provides secure access for data traffic in Power BI	No	Yes
Single sign-on (SSO) for DirectQuery sources, which allows connecting to data sources by using the report user's identity	No	Yes

Feature	AAS	Power BI Premium
Row-level security (RLS), which restricts access to specific rows of data for specific users	Yes	Yes
Object-level security (OLS), which restricts access to specific tables or columns for specific users	Yes	Yes
Firewall, which when enabled, allows setting allowed IP address ranges	Yes	No [1]
Governance		
Microsoft Purview integration, which helps customers manage and govern Power BI items	No	Yes
Microsoft Information Protection (MIP) sensitivity labels and integration with Microsoft Defender for Cloud Apps for data loss prevention	No	Yes
Content endorsement, to promote or certify valuable, high-quality Power BI items	No	Yes
Semantic modeling		
Compatibility with Power BI Desktop	No	Yes
Composite models including using DirectQuery for Power BI semantic models and AAS	No	Yes
Translations for multi-language model versions observed by the Power BI service	No	Yes
Analysis Service engine semantic modeling	Yes	Yes
Model management		
Incremental refresh, which uses policies to automate partition management and can help deliver near real-time reporting (see hybrid tables)	No	Yes
Deployment pipelines, which manage the lifecycle of Power BI content	No	Yes
Scheduled refresh, which keeps cached semantic model data current	No	Yes
Enhanced refresh, which allows any programming language to perform asynchronous semantic model refreshes by using a REST API call	Yes	Yes
Backup and restore	Yes	Yes
Semantic model workload settings, which control Premium capacity workloads	No	Yes
Server properties, which control Analysis Services server instance properties	Yes	Yes

Feature	AAS	Power BI Premium
Alias server names, which allow connecting to an Analysis Services server instance by using a shorter alias	Yes	No
XMLA endpoint enabled APIs for scripting and compatibility with services for automation and ALM including Azure Functions, Azure Automation and Azure DevOps	Yes	Yes
Connectivity		
Support for all Power BI data sources	No	Yes
XMLA endpoint, which allows open-platform connectivity for data model consumption and visualization tools, including third-party tools	Yes	Yes
Multi-Geo feature, which helps multinational customers address regional, industry-specific, or organizational data residency requirements	Yes	Yes
Discoverability		
OneLake catalog integration, which helps users discover, explore, and use Power BI semantic models	No	Yes
Data lineage view and semantic model impact analysis, which help users understand and assess Power BI item dependencies	No	Yes
Monitoring and diagnostic logging		
Fabric Capacity Metrics app, which provides monitoring capabilities for Power BI capacities	No	Yes
Audit log, which tracks user activities across Power BI and Microsoft 365	No	Yes
Azure Log Analytics (LA) integration, which allows administrators to configure a Log Analytics connection for a Power BI workspace	Yes	Yes
Metric alerts in Azure Monitor, which provide a way to get notified when one of your multi-dimensional metrics crosses a threshold	Yes	No
XMLA endpoint, which allows diagnostic logging tool connections, including SQL Server Profiler	Yes	Yes
SQL Server Extended Events (xEvents), which is a light-weight tracing and performance monitoring system useful for diagnosing issues	Yes	No

[1] Use VNet connectivity and Azure Private Link instead

Cost comparison

When comparing Power BI Premium to AAS costs, be sure to consider factors beyond price per core. Power BI provides reduced cost of ownership and business value, and with many features that are only available to Power BI data models.

Also, assuming you already use Power BI in your organization, calculate costs based on the existing profile that *combines* AAS and Power BI. Compare the existing profile with the target profile on Power BI Premium. To determine the target profile, be sure to consider the following points:

- Region requirements.
- The largest AAS data model size in each region.
- The number of users in each region.
- The number of users required to develop and manage content.
- CPU consumption across AAS and Power BI Premium.

> **Important**
>
> CPU consumption across AAS and Power BI Premium may vary significantly due to numerous factors. Factors can include the use of other workloads on the same capacities, refresh patterns, and query patterns. We recommended that you perform in-depth analysis to quantify comparative CPU consumption across AAS and Power BI Premium for migrated models.

> **Tip**
>
> To help determine the right type and number of licenses for your business requirements and circumstances, see **this related article**.

Consolidation opportunity

Many AAS customers already have Power BI reports that connect to AAS. So, migration to Power BI can represent an opportunity to consolidate BI items in Power BI Premium. Consolidation makes the larger sized Premium SKUs more economically viable and can help to provide higher levels of throughput and scalability.

PPU licenses

The Premium Per User (PPU) license is a per-user license that provides a lower-cost price point for Premium. PPU licenses are typically purchased by small and medium-sized companies. They support all the Premium capabilities for data modeling listed earlier.

> **Tip**
>
> It's possible to incrementally upgrade Power BI Pro licenses to PPU licenses.

Pro licenses

A Pro (or PPU) license is required to publish and manage Power BI content. Pro licenses are typically assigned to developers and administrators, not end users.

Development and test environments

AAS offers the **D** and **B** SKUs at lower cost with reduced service-level agreements and/or fewer features than the **S** SKUs. Some AAS customers use these SKUs for development and test environments. While there's no direct equivalent in Power BI, it might make sense to use PPU licenses for development and test environments. Such environments typically don't have large numbers of users because they're limited to developers and testers. Alternatively, consider using an **A** SKU in Azure for testing Premium capacity functionality.

For more information, see:

- Power BI pricing
- Azure Analysis Services pricing
- Purchase A SKUs for testing and other scenarios

Scalability benefits

Power BI Premium delivers scalability, performance, and cost-of-ownership benefits not available in AAS.

Power BI Premium provides features that enable fast interactive analysis over big data. Such features include aggregations, composite models, and hybrid tables. Each feature offers a different way to optimally combine import and DirectQuery storage modes, effectively reducing memory use. AAS, on the other hand, doesn't support these capabilities; the entire data model uses either import or DirectQuery storage mode.

Power BI Premium limits memory per semantic model, and not per capacity or server. Conversely, AAS requires all data models fit in memory on a single server. That requirement can compel customers with large data models to purchase larger SKU sizes.

Thanks to the distributed nature of the Premium architecture, more semantic models can be refreshed in parallel. Performing concurrent refreshes on the same AAS server can lead to refresh errors due to exceeding server memory limits.

In Power BI Premium, CPU consumption during refresh is spread across 24-hour periods. Power BI Premium evaluates capacity throughput to provide resilience to temporal spikes in demand for compute resources. When necessary, it can delay refreshes until sufficient resources become available. This automatic behavior reduces the need for customers to perform detailed analysis and manage automation scripts to scale servers up or down. Premium customers should decide on the optimal SKU size for their overall CPU consumption requirements.

Another advantage of Power BI Premium is that it's able to dynamically balance the semantic models depending on the load of the system. This automatic behavior ensures busy/active semantic models get the necessary memory and CPU resources, while more idle semantic models can be evicted or migrated to other nodes. Semantic models are candidates for eviction when they're not used. They'll be loaded on-demand so that only the required data is loaded into memory without having to load the whole semantic model. On the other hand, AAS requires all data models be fully loaded in memory always. This requirement means queries to AAS can rely on the data model being available, but – especially for Power BI capacities with a high number of data models when some of them are used infrequently – dynamic memory management can make more efficient use of memory.

Lastly, Power BI Premium is able to better utilize next-generation hardware rollouts to benefit from scalability and performance enhancements.

Considerations and limitations

There are considerations and limitations to factor into your planning before migrating to Power BI Premium.

Permissions

AAS and SSAS use roles to manage data model access. There are two types of roles: the *server role* and *database roles*. The server role is a fixed role that grants administrator access to the Analysis Services server instance. Database roles, which are set by data modelers and administrators, control access to the database and data for non-administrator users.

Unlike AAS, in Power BI, you only use roles to enforce RLS or OLS. To grant permissions beyond RLS and OLS, use the Power BI security model (workspace roles and semantic

model permissions). For more information, see Semantic model permissions.

For more information about Power BI model roles, see Semantic model connectivity with the XMLA endpoint (Model roles).

When you migrate a data model from AAS to Power BI Premium, you must take the following points into consideration:

- Users who were granted **Read** permission on a model in AAS must be granted **Build** permission on the migrated Power BI semantic model.
- Users who were granted the **Administrator** permission on a model in AAS must be granted **Write** permission on the migrated Power BI semantic model.

Refresh automation

Power BI Premium supports XMLA endpoint-enabled APIs for scripting, such as Tabular Model Scripting Language (TMSL), Tabular Object Model (TOM), and the PowerShell SqlServer module. These APIs have almost symmetric interfaces to AAS. For more information, see Semantic model connectivity with the XMLA endpoint (Client applications and tools).

Compatibility with services for automation, including Azure Functions, Azure Automation, and Azure Logic Apps, is enabled in the same way.

Generally, scripts and processes that automate partition management and processing in AAS will work in Power BI Premium. Bear in mind that Power BI Premium semantic models support the incremental refresh feature, which provides automated partition management for tables that frequently load new and updated data.

Like for AAS, you can use a service principal as an automation account for Power BI semantic model management operations, such as refreshes. For more information, see Semantic model connectivity with the XMLA endpoint (Service principals).

Custom security

Like for AAS, applications can use a service principal to query a Power BI Premium per capacity or Power BI Embedded semantic model by using the CustomData feature.

However, you can't assign a service principal to a model role in Power BI Premium. Instead, a service principal gains access by assignment to the workspace **admin** or **member** role.

> **Note**

You can't use the CustomData feature when querying Premium Per User (PPU) semantic models because it would be in violation of the **license terms and conditions**.

Impersonation for testing

Impersonation techniques, including the EffectiveUserName and the Roles connection string properties, are supported by AAS and Power BI Premium. You typically use them when testing security roles.

Network security

Setting up network security in AAS requires enabling the firewall and configuring IP address ranges for only those computers accessing the server.

Power BI doesn't have a firewall feature. Instead, Power BI offers a superior network security model by using VNets and Private Links. For more information, see What is a virtual network (VNet)?.

Data sources and credentials

AAS defines credentials for each data source declared in the TOM tabular metadata. However, Power BI doesn't work that way. Because Power BI can share data sources credentials across multiple semantic models, credentials are set in the Power BI service.

Any XMLA-based process that sets data source credentials must be replaced. For more information, see Semantic model connectivity with the XMLA endpoint (Deploy model projects from Visual Studio).

Backup and restore

Backup and restore in AAS requires Azure Blob storage, while in Power BI Premium it requires an Azure Data Lake Storage Gen2 (ADLS Gen2) account. Apart from the storage account difference, backup and restore work the same way in both products.

For more information, see Backup and restore semantic models with Power BI Premium.

On-premises data gateway

Both AAS and Power BI Premium use the same on-premises data gateway to connect to data sources. However, the setup steps are different.

For information on how to set up gateway data sources for Power BI Premium, see Add or remove a gateway data source.

Server properties

Unlike AAS, Power BI Premium doesn't support server properties. Instead, you manage Premium capacity settings.

Link files

Unlike AAS, Power BI Premium doesn't support alias server names.

Dynamic management views (DMVs)

Some DMVs that work in AAS aren't accessible in Power BI Premium because they require Analysis Services server-admin permissions. Power BI has workspace roles, but there isn't a workspace role that grants the equivalent of Analysis Services server-admin permissions.

PowerShell

You can use the SqlServer PowerShell module AAS cmdlets to automate semantic model management tasks, including refresh operations. For more information, see Analysis Services PowerShell Reference.

However, the Az.AnalysisServices module AAS cmdlets aren't supported for Power BI semantic models. Instead, use the Microsoft Power BI Cmdlets for Windows PowerShell and PowerShell Core.

Diagnostic logging

AAS integrates with Azure Monitor for diagnostic logging. The most common target for AAS logs is to Log Analytics workspaces.

Power BI Premium also supports logging to Log Analytics workspaces. Currently, the events sent to Log Analytics are mainly AS engine events. However, not all events supported for AAS are supported for Power BI. The Log Analytics schema for Power BI contains differences compared to AAS, which means existing queries on AAS may not work in Power BI.

Power BI offers another diagnostic logging capability that isn't offered in AAS. For more information, see Use the Microsoft Fabric Capacity Metrics app.

SQL Server Extended Events (xEvents) are supported in AAS but not in Power BI Premium. For more information, see Monitor Analysis Services with SQL Server Extended Events.

Business-to-business (B2B)

Both AAS and Power BI support Microsoft Entra B2B collaboration, which enables and governs sharing with external users. Notably, the User Principal Name (UPN) format required by AAS is different to Power BI.

To identify the user, Power BI utilizes a unique name claim in Microsoft Entra ID while AAS uses an email claim. While there may be many instances where these two identifiers align, the unique name format is more stringent. If using dynamic RLS in Power BI, ensure that the value in the user identity table matches the account used to sign in to Power BI.

Scale-out

Azure Analysis Services scale-out is supported by Power BI Premium. For more information see Power BI semantic model scale out.

Migration feature

The Microsoft Azure Analysis Services to Microsoft Power BI Premium migration feature in Power BI migrates as AAS database to a semantic model in Power BI Premium, Power BI Premium Per User, or Power BI Embedded workspace. For more information, see Migrate Azure Analysis Services to Power BI.

Related content

For more information about this article, check out the following resources:

- Migrate from Azure Analysis Services to Power BI Premium: Migration scenarios
- Migrate Azure Analysis Services to Power BI
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Migrate from Azure Analysis Services to Power BI Premium: Migration scenarios

Article • 12/30/2024

This article compares six hypothetical scenarios when migrating from Azure Analysis Services (AAS) to Power BI Premium. These scenarios can help you to determine the right type and number of licenses for your business requirements and circumstances.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

> **Note**
>
> An attempt has been made to ensure these scenarios are representative of real customer migrations, however individual customer scenarios will of course differ. Also, this article doesn't include pricing details. You can find current pricing here:
>
> - Power BI pricing
> - Azure Analysis Services pricing

When comparing Power BI Premium to AAS costs, be sure to consider factors beyond price per core. Power BI provides reduced cost of ownership and business value, and with many features that are only available to Power BI data models.

Also, assuming you already use Power BI in your organization, calculate costs based on the existing profile that *combines* AAS and Power BI. Compare the existing profile with the target profile on Power BI Premium. To determine the target profile, be sure to consider the following points:

- Region requirements.
- The largest AAS data model size in each region.
- The number of users in each region.

- The number of users required to develop and manage content.
- CPU consumption across AAS and Power BI Premium.

> **Important**
>
> CPU consumption across AAS and Power BI Premium may vary significantly due to numerous factors. Factors can include the use of other workloads on the same capacities, refresh patterns, and query patterns. We recommended that you perform in-depth analysis to quantify comparative CPU consumption across AAS and Power BI Premium for migrated models.

Migration scenario 1

In the first migration scenario, the customer uses Power BI Premium for the frontend and AAS for the backend. There are 20 developers who are each responsible for the development and test environments, and for deployment to production.

Here are their current AAS licenses:

Expand table

Environment	Largest model	AAS SKU
Production	60 GB	S4
Production	30 GB	S2
Production	15 GB	S1
Test	5 GB	B1
Development	1 GB	D1

Here are their current Power BI licenses:

Expand table

Environment	Power BI license	Users
Production	Premium P2	5,000
Test/development	Premium P1	20
Production/test/development	Pro	20

Once migrated to Power BI Premium:

- The three existing production AAS models can be consolidated to run in a Premium P3 capacity.
- The 20 developers will need Premium Per User (PPU) licenses to access test models above 1 GB in size.

Here are the proposed Power BI licenses:

Expand table

Environment	Power BI license	Users	Largest model
Production	Premium P3	5,000	60 GB
Production/test/development	PPU	20	5 GB

Migration scenario 2

In this migration scenario, the customer uses Power BI Premium for the frontend and AAS for the backend. Production environments are running *in different regions*. There are 20 developers who are each responsible for the development and test environments, and for deployment to production.

Here are their current AAS licenses:

Expand table

Region	Environment	Largest model	AAS SKU
West Europe	Production	60 GB	S4
Brazil South	Production	30 GB	S2
West US	Production	15 GB	S1
West US	Test	5 GB	B1
West US	Development	1 GB	D1

Here are their current Power BI licenses:

Expand table

Region	Environment	Power BI license	Users
West Europe	Production	Premium P1	2,000
Brazil South	Production	Premium P1	2,000
West US	Production	Premium P1	2,000
West US	Test/development	Premium P1	20
West US	Production/test/development	Pro	20

Once migrated to Power BI Premium:

- The customer needs a Premium capacity in each of the three regions (because the three existing production AAS models run in different regions). Each capacity size is based on the largest model.
- The 20 developers will need PPU licenses to access test models above 1 GB in size.

Here are the proposed Power BI licenses:

Expand table

Region	Environment	Power BI license	Users	Largest model
West Europe	Production	Premium P3	2,000	60 GB
Brazil South	Production	Premium P2	2,000	30 GB
West US	Production	Premium P1	2,000	15 GB
West US	Production/test/development	PPU	20	5 GB

Migration scenario 3

In this migration scenario, the customer has Power BI Pro licenses for all users available with their Office 365 E5 subscription, and they use AAS for the backend. There are 15 developers who are each responsible for the development and test environments, and for deployment to production.

Here are their current AAS licenses:

Expand table

Environment	Largest model	AAS SKU
Production	35 GB	S2
Production	30 GB	S2
Test	5 GB	B1
Development	1 GB	D1

Here are their current Power BI licenses:

Expand table

Environment	Power BI license	Users
Production	Pro (as part of E5)	4,000
Production/test/development	Pro (as part of E5)	15

Once migrated to Power BI Premium:

- The two existing production AAS models can be consolidated to run in a Premium P2 capacity.
- The 15 developers will need PPU licenses to access test models above 1 GB in size. (An add-on is available to step up from Pro to PPU.)

Here are the proposed Power BI licenses:

Expand table

Environment	Power BI license	Users	Largest model
Production	Premium P2	4,000	35 GB
Production/test/development	PPU	15	5 GB

Migration scenario 4

In this migration scenario, the customer has Power BI Pro licenses for all users, and they use AAS for the backend. There are five developers who are each responsible for the development and test environments, and for deployment to production.

Here are their current AAS licenses:

Expand table

Environment	Largest model	AAS SKU
Production	35 GB	S2
Production	10 GB	S1
Test	5 GB	B1
Development	1 GB	D1

Here are their current Power BI licenses:

Expand table

Environment	Power BI license	Users
Production	Pro	350
Production/test/development	Pro	5

Once migrated to Power BI Premium:

- The two existing production AAS models can run in PPU workspaces.
- All end users and developers will need PPU licenses.

Here are the proposed Power BI licenses:

Expand table

Environment	Power BI license	Users	Largest model
Production	PPU	350	35 GB
Production/test/development	PPU	5	5 GB

Migration scenario 5

In this migration scenario, the customer uses Power BI Premium for the frontend and AAS for the backend. There are 25 developers who are each responsible for the development and test environments, and for deployment to production.

Here are their current AAS licenses:

Expand table

Environment	Largest model	AAS SKU
Production	220 GB	S9
Production	150 GB	S8
Production	60 GB	S4
Test	5 GB	B1
Development	1 GB	D1

Here are their current Power BI licenses:

Expand table

Environment	Power BI license	Users
Production	Premium P3	7,500
Production	Premium P2	4,500
Test/development	Premium P1	25
Production/test/development	Pro	25

Once migrated to Power BI Premium:

- The three existing production AAS models can be consolidated to run in a Premium P5 capacity.
- The 20 developers will need PPU licenses to access test models above 1 GB in size.

Here are the proposed Power BI licenses:

Expand table

Environment	Power BI license	Users	Largest model
Production	Premium P5	12,000	220 GB
Production/test/development	PPU	25	5 GB

Migration scenario 6

In this migration scenario, an ISV company has 400 customers. Each customer has its own SQL Server Analysis Services (SSAS) multidimensional model (also known as a *cube*).

The analysis below compares Azure Analysis Services with the Power BI Embedded alternative.

- The 400 tenants are mainly accessed by 50 analysts from the ISV company as well as two users (on average) from each customer.
- The total size of the models is about 100 GB.

Here are their estimated AAS licenses:

Expand table

Environment	Largest model	AAS SKU
Production	8 GB	S4
Test	8 GB	B1
Development	1 GB	D1

Here are their current Power BI licenses:

Expand table

Users	Power BI license	Users
Customers	Pro	800
Analysts	Pro	50
Developers	Pro	20

Once migrated to Power BI Premium:

- The A1/P4 SKU was chosen to allow for future model size growth (EM3/A3 SKU can work also).
- The 50 analysts will need PPU licenses to access test models above 1 GB in size.
- The total size of the 400 models isn't relevant for pricing; only the largest model size is important.

Here are their proposed Power BI licenses:

Expand table

Environment	Power BI license	Users	Largest model
Production	Premium P1 / Power BI Embedded A4	Not applicable	25 GB
Test/development	Premium EM3 / Power BI Embedded A3	Not applicable	10 GB
Developers	Pro	20	Not applicable
Production/test/development	PPU	50	Not applicable

Premium migration benefits

Customers can realize many benefits when they migrate from AAS to Power BI Premium.

- Customers can consolidate to a single platform that reduces cost duplication of paying for both AAS and Power BI Premium.
- By using Premium for their entire BI stack, customers can unlock increased performance and features. They only need Pro licenses for developers and admins, but not for end users.
- Customers can use Power BI Premium scalability to reduce their capacity requirements, since memory is limited per semantic model and isn't compared to total over the server as it is in AAS. For more information, see Memory allocation.
- For development and test environments, customers can take advantage of PPU licensing instead of having Premium capacities. PPU licenses provide users access to Premium features like the XMLA endpoint, deployment pipelines, and Premium dataflow features. Furthermore, they can work with models that above 1 GB in size.

Related content

For more information about this article, check out the following resources:

- Migrate from Azure Analysis Services to Power BI Premium
- What is Power BI Premium?
- Power BI pricing
- Azure Analysis Services pricing
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Appropriate use of error functions

Article • 09/20/2022

As a data modeler, when you write a DAX expression that might raise an evaluation-time error, you can consider using two helpful DAX functions.

- The ISERROR function, which takes a single expression and returns TRUE if that expression results in error.
- The IFERROR function, which takes two expressions. Should the first expression result in error, the value for the second expression is returned. It is in fact a more optimized implementation of nesting the ISERROR function inside an IF function.

However, while these functions can be helpful and can contribute to writing easy-to-understand expressions, they can also significantly degrade the performance of calculations. It can happen because these functions increase the number of storage engine scans required.

Most evaluation-time errors are due to unexpected BLANKs or zero values, or invalid data type conversion.

Recommendations

It's better to avoid using the ISERROR and IFERROR functions. Instead, apply defensive strategies when developing the model and writing expressions. Strategies can include:

- **Ensuring quality data is loaded into the model:** Use Power Query transformations to remove or substitute invalid or missing values, and to set correct data types. A Power Query transformation can also be used to filter rows when errors, like invalid data conversion, occur.

  Data quality can also be controlled by setting the model column **Is Nullable** property to Off, which will fail the data refresh should BLANKs be encountered. If this failure occurs, data loaded as a result of a successful refresh will remain in the tables.

- **Using the IF function:** The IF function logical test expression can determine whether an error result would occur. Note, like the ISERROR and IFERROR functions, this function can result in additional storage engine scans, but will likely perform better than them as no error needs to be raised.

- **Using error-tolerant functions:** Some DAX functions will test and compensate for error conditions. These functions allow you to enter an alternate result that would

be returned instead. The DIVIDE function is one such example. For additional guidance about this function, read the DAX: DIVIDE function vs divide operator (/) article.

Example

The following measure expression tests whether an error would be raised. It returns BLANK in this instance (which is the case when you do not provide the IF function with a value-if-false expression).

```
DAX

Profit Margin
= IF(ISERROR([Profit] / [Sales]))
```

This next version of the measure expression has been improved by using the IFERROR function in place of the IF and ISERROR functions.

```
DAX

Profit Margin
= IFERROR([Profit] / [Sales], BLANK())
```

However, this final version of the measure expression achieves the same outcome, yet more efficiently and elegantly.

```
DAX

Profit Margin
= DIVIDE([Profit], [Sales])
```

See also

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Get help at Microsoft Q&A

Avoid converting BLANKs to values

Article • 09/20/2022

As a data modeler, when writing measure expressions you might come across cases where a meaningful value can't be returned. In these instances, you may be tempted to return a value—like zero—instead. It's suggested you carefully determine whether this design is efficient and practical.

Consider the following measure definition that explicitly converts BLANK results to zero.

DAX

```
Sales (No Blank) =
IF(
    ISBLANK([Sales]),
    0,
    [Sales]
)
```

Consider another measure definition that also converts BLANK results to zero.

DAX

```
Profit Margin =
DIVIDE([Profit], [Sales], 0)
```

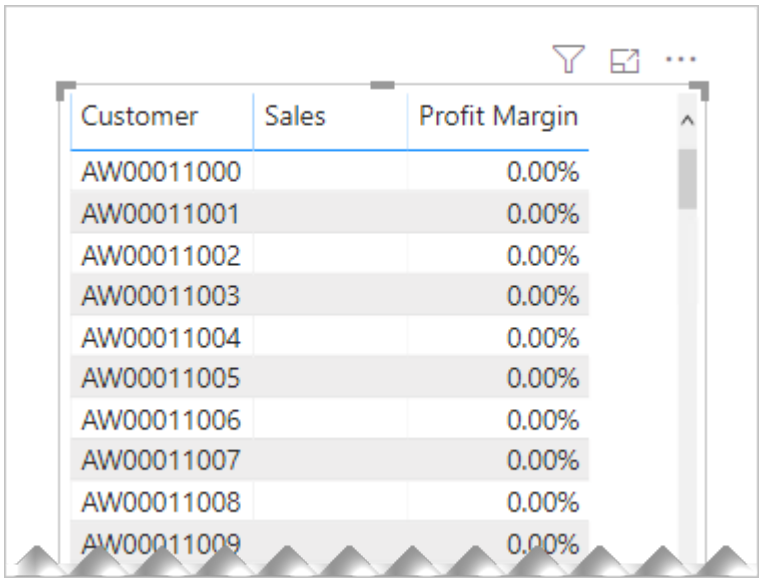
The DIVIDE function divides the **Profit** measure by the **Sales** measure. Should the result be zero or BLANK, the third argument—the alternate result (which is optional)—is returned. In this example, because zero is passed as the alternate result, the measure is guaranteed to always return a value.

These measure designs are inefficient and lead to poor report designs.

When they're added to a report visual, Power BI attempts to retrieve all groupings within the filter context. The evaluation and retrieval of large query results often leads to slow report rendering. Each example measure effectively turns a sparse calculation into a dense one, forcing Power BI to use more memory than necessary.

Also, too many groupings often overwhelm your report users.

Let's see what happens when the **Profit Margin** measure is added to a table visual, grouping by customer.



Customer	Sales	Profit Margin
AW00011000		0.00%
AW00011001		0.00%
AW00011002		0.00%
AW00011003		0.00%
AW00011004		0.00%
AW00011005		0.00%
AW00011006		0.00%
AW00011007		0.00%
AW00011008		0.00%
AW00011009		0.00%

The table visual displays an overwhelming number of rows. (There are in fact 18,484 customers in the model, and so the table attempts to display all of them.) Notice that the customers in view haven't achieved any sales. Yet, because the **Profit Margin** measure always returns a value, they are displayed.

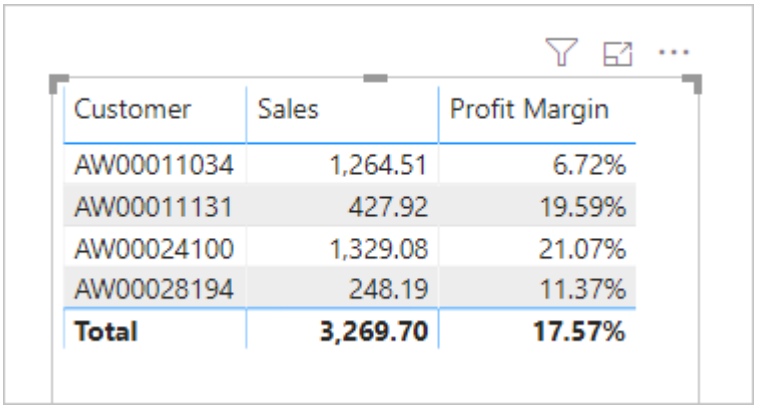
> **Note**
>
> When there are too many data points to display in a visual, Power BI may use data reduction strategies to remove or summarize large query results. For more information, see **Data point limits and strategies by visual type**.

Let's see what happens when the **Profit Margin** measure definition is improved. It now returns a value only when the **Sales** measure isn't BLANK (or zero).

DAX

```
Profit Margin =
DIVIDE([Profit], [Sales])
```

The table visual now displays only customers who have made sales within the current filter context. The improved measure results in a more efficient and practical experience for your report users.



Customer	Sales	Profit Margin
AW00011034	1,264.51	6.72%
AW00011131	427.92	19.59%
AW00024100	1,329.08	21.07%
AW00028194	248.19	11.37%
Total	**3,269.70**	**17.57%**

> **Tip**
>
> When necessary, you can configure a visual to display all groupings (that return values or BLANK) within the filter context by enabling the **Show Items With No Data** option.

Recommendation

It's recommended that your measures return BLANK when a meaningful value cannot be returned.

This design approach is efficient, allowing Power BI to render reports faster. Also, returning BLANK is better because report visuals—by default—eliminate groupings when summarizations are BLANK.

See also

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Get help at Microsoft Q&A

Avoid using FILTER as a filter argument

Article • 09/20/2022

As a data modeler, it's common you'll write DAX expressions that need to be evaluated in a modified filter context. For example, you can write a measure definition to calculate sales for "high margin products". We'll describe this calculation later in this article.

> **Note**
>
> This article is especially relevant for model calculations that apply filters to Import tables.

The CALCULATE and CALCULATETABLE DAX functions are important and useful functions. They let you write calculations that remove or add filters, or modify relationship paths. It's done by passing in filter arguments, which are either Boolean expressions, table expressions, or special filter functions. We'll only discuss Boolean and table expressions in this article.

Consider the following measure definition, which calculates red product sales by using a table expression. It will replace any filters that might be applied to the **Product** table.

DAX

```
Red Sales =
CALCULATE(
    [Sales],
    FILTER('Product', 'Product'[Color] = "Red")
)
```

The CALCULATE function accepts a table expression returned by the FILTER DAX function, which evaluates its filter expression for each row of the **Product** table. It achieves the correct result—the sales result for red products. However, it could be achieved much more efficiently by using a Boolean expression.

Here's an improved measure definition, which uses a Boolean expression instead of the table expression. The KEEPFILTERS DAX function ensures any existing filters applied to the **Color** column are preserved, and not overwritten.

DAX

```
Red Sales =
CALCULATE(
    [Sales],
```

```
    KEEPFILTERS('Product'[Color] = "Red")
)
```

It's recommended you pass filter arguments as Boolean expressions, whenever possible. It's because Import model tables are in-memory column stores. They are explicitly optimized to efficiently filter columns in this way.

There are, however, restrictions that apply to Boolean expressions when they're used as filter arguments. They:

- Cannot reference columns from multiple tables
- Cannot reference a measure
- Cannot use nested CALCULATE functions
- Cannot use functions that scan or return a table

It means that you'll need to use table expressions for more complex filter requirements.

Consider now a different measure definition. The requirement is to calculate sales, but only for months that have achieved a profit.

DAX

```
Sales for Profitable Months =
CALCULATE(
    [Sales],
    FILTER(
        VALUES('Date'[Month]),
        [Profit] > 0)
    )
)
```

In this example, the FILTER function must be used. It's because it requires evaluating the **Profit** measure to eliminate those months that didn't achieve a profit. It's not possible to use a measure in a Boolean expression when it's used as a filter argument.

Recommendations

For best performance, it's recommended you use Boolean expressions as filter arguments, whenever possible.

Therefore, the FILTER function should only be used when necessary. You can use it to perform filter complex column comparisons. These column comparisons can involve:

- Measures
- Other columns

- Using the OR DAX function, or the OR logical operator (||)

See also

- Filter functions (DAX)
- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Get help at Microsoft Q&A

Column and measure references

Article • 09/20/2022

As a data modeler, your DAX expressions will refer to model columns and measures. Columns and measures are always associated with model tables, but these associations are different, so we have different recommendations on how you'll reference them in your expressions.

Columns

A column is a table-level object, and column names must be unique within a table. So it's possible that the same column name is used multiple times in your model—providing they belong to different tables. There's one more rule: a column name cannot have the same name as a measure name or hierarchy name that exists in the same table.

In general, DAX will not force using a *fully qualified* reference to a column. A fully qualified reference means that the table name precedes the column name.

Here's an example of a calculated column definition using only column name references. The **Sales** and **Cost** columns both belong to a table named **Orders**.

DAX

```
Profit = [Sales] - [Cost]
```

The same definition can be rewritten with fully qualified column references.

DAX

```
Profit = Orders[Sales] - Orders[Cost]
```

Sometimes, however, you'll be required to use fully qualified column references when Power BI detects ambiguity. When entering a formula, a red squiggly and error message will alert you. Also, some DAX functions like the LOOKUPVALUE DAX function, require the use of fully qualified columns.

It's recommended you always fully qualify your column references. The reasons are provided in the Recommendations section.

Measures

A measure is a model-level object. For this reason, measure names must be unique within the model. However, in the **Fields** pane, report authors will see each measure associated with a single model table. This association is set for cosmetic reasons, and you can configure it by setting the **Home Table** property for the measure. For more information, see Measures in Power BI Desktop (Organizing your measures).

It's possible to use a fully qualified measure in your expressions. DAX intellisense will even offer the suggestion. However, it isn't necessary, and it's not a recommended practice. If you change the home table for a measure, any expression that uses a fully qualified measure reference to it will break. You'll then need to edit each broken formula to remove (or update) the measure reference.

It's recommended you never qualify your measure references. The reasons are provided in the Recommendations section.

Recommendations

Our recommendations are simple and easy to remember:

- Always use fully qualified column references
- Never use fully qualified measure references

Here's why:

- **Formula entry**: Expressions will be accepted, as there won't be any ambiguous references to resolve. Also, you'll meet the requirement for those DAX functions that require fully qualified column references.
- **Robustness**: Expressions will continue to work, even when you change a measure home table property.
- **Readability**: Expressions will be quick and easy to understand—you'll quickly determine that it's a column or measure, based on whether it's fully qualified or not.

See also

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Get help at Microsoft Q&A

DIVIDE function vs. divide operator (/)

Article • 09/20/2022

As a data modeler, when you write a DAX expression to divide a numerator by a denominator, you can choose to use the DIVIDE function or the divide operator (/ - forward slash).

When using the DIVIDE function, you must pass in numerator and denominator expressions. Optionally, you can pass in a value that represents an *alternate result*.

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

The DIVIDE function was designed to automatically handle division by zero cases. If an alternate result is not passed in, and the denominator is zero or BLANK, the function returns BLANK. When an alternate result is passed in, it's returned instead of BLANK.

The DIVIDE function is convenient because it saves your expression from having to first test the denominator value. The function is also better optimized for testing the denominator value than the IF function. The performance gain is significant since checking for division by zero is expensive. Further using DIVIDE results in a more concise and elegant expression.

Example

The following measure expression produces a safe division, but it involves using four DAX functions.

DAX

```
Profit Margin =
IF(
    OR(
        ISBLANK([Sales]),
        [Sales] == 0
    ),
    BLANK(),
    [Profit] / [Sales]
)
```

This measure expression achieves the same outcome, yet more efficiently and elegantly.

DAX

```
Profit Margin =
DIVIDE([Profit], [Sales])
```

Recommendations

It's recommended that you use the DIVIDE function whenever the denominator is an expression that *could* return zero or BLANK.

In the case that the denominator is a constant value, we recommend that you use the divide operator. In this case, the division is guaranteed to succeed, and your expression will perform better because it will avoid unnecessary testing.

Carefully consider whether the DIVIDE function should return an alternate value. For measures, it's usually a better design that they return BLANK. Returning BLANK is better because report visuals—by default—eliminate groupings when summarizations are BLANK. It allows the visual to focus attention on groups where data exists. When necessary, in Power BI, you can configure the visual to display all groups (that return values or BLANK) within the filter context by enabling the Show items with no data option.

See also

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Get help at Microsoft Q&A

Use COUNTROWS instead of COUNT

Article • 12/13/2023

As a data modeler, sometimes you might need to write a DAX expression that counts table rows. The table could be a model table or an expression that returns a table.

Your requirement can be achieved in two ways. You can use the COUNT function to count column values, or you can use the COUNTROWS function to count table rows. Both functions will achieve the same result, providing that the counted column contains no BLANKs.

The following measure definition presents an example. It calculates the number of **OrderDate** column values.

DAX

```
Sales Orders =
COUNT(Sales[OrderDate])
```

Providing that the granularity of the **Sales** table is one row per sales order, and the **OrderDate** column does not contain BLANKs, then the measure will return a correct result.

However, the following measure definition is a better solution.

DAX

```
Sales Orders =
COUNTROWS(Sales)
```

There are three reasons why the second measure definition is better:

- It's more efficient, and so it will perform better.
- It doesn't consider BLANKs contained in any column of the table.
- The intention of formula is clearer, to the point of being self-describing.

Recommendation

When it's your intention to count table rows, it's recommended you always use the COUNTROWS function.

Related content

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Use SELECTEDVALUE instead of VALUES

Article • 12/13/2023

As a data modeler, sometimes you might need to write a DAX expression that tests whether a column is filtered by a specific value.

In earlier versions of DAX, this requirement was safely achieved by using a pattern involving three DAX functions; IF, HASONEVALUE and VALUES. The following measure definition presents an example. It calculates the sales tax amount, but only for sales made to Australian customers.

DAX

```
Australian Sales Tax =
IF(
    HASONEVALUE(Customer[Country-Region]),
    IF(
        VALUES(Customer[Country-Region]) = "Australia",
        [Sales] * 0.10
    )
)
```

In the example, the HASONEVALUE function returns TRUE only when a single value of the **Country-Region** column is visible in the current filter context. When it's TRUE, the VALUES function is compared to the literal text "Australia". When the VALUES function returns TRUE, the **Sales** measure is multiplied by 0.10 (representing 10%). If the HASONEVALUE function returns FALSE—because more than one value filters the column—the first IF function returns BLANK.

The use of the HASONEVALUE is a defensive technique. It's required because it's possible that multiple values filter the **Country-Region** column. In this case, the VALUES function returns a table of multiple rows. Comparing a table of multiple rows to a scalar value results in an error.

Recommendation

It's recommended that you use the SELECTEDVALUE function. It achieves the same outcome as the pattern described in this article, yet more efficiently and elegantly.

Using the SELECTEDVALUE function, the example measure definition is now rewritten.

DAX

```
Australian Sales Tax =
IF(
    SELECTEDVALUE(Customer[Country-Region]) = "Australia",
    [Sales] * 0.10
)
```

Tip

It's possible to pass an *alternate result* value into the SELECTEDVALUE function. The alternate result value is returned when either no filters—or multiple filters—are applied to the column.

Related content

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Use variables to improve your DAX formulas

Article • 10/31/2022

As a data modeler, writing and debugging some DAX calculations can be challenging. It's common that complex calculation requirements often involve writing compound or complex expressions. Compound expressions can involve the use of many nested functions, and possibly the reuse of expression logic.

Using variables in your DAX formulas can help you write more complex and efficient calculations. Variables can improve performance, reliability, readability, and reduce complexity.

In this article, we'll demonstrate the first three benefits by using an example measure for year-over-year (YoY) sales growth. (The formula for YoY sales growth is period sales, minus sales for the same period last year, divided by sales for the same period last year.)

Let's start with the following measure definition.

DAX

```
Sales YoY Growth % =
DIVIDE(
    ([Sales] - CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12,
MONTH))),
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
)
```

The measure produces the correct result, yet let's now see how it can be improved.

Improve performance

Notice that the formula repeats the expression that calculates "same period last year". This formula is inefficient, as it requires Power BI to evaluate the same expression twice. The measure definition can be made more efficient by using a variable, VAR.

The following measure definition represents an improvement. It uses an expression to assign the "same period last year" result to a variable named **SalesPriorYear**. The variable is then used twice in the RETURN expression.

DAX

```
Sales YoY Growth % =
VAR SalesPriorYear =
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
RETURN
    DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
```

The measure continues to produce the correct result, and does so in about half the query time.

Improve readability

In the previous measure definition, notice how the choice of variable name makes the RETURN expression simpler to understand. The expression is short and self-describing.

Simplify debugging

Variables can also help you debug a formula. To test an expression assigned to a variable, you temporarily rewrite the RETURN expression to output the variable.

The following measure definition returns only the **SalesPriorYear** variable. Notice how it comments-out the intended RETURN expression. This technique allows you to easily revert it back once your debugging is complete.

DAX

```
Sales YoY Growth % =
VAR SalesPriorYear =
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
RETURN
    --DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
    SalesPriorYear
```

Reduce complexity

In earlier versions of DAX, variables were not yet supported. Complex expressions that introduced new filter contexts were required to use the EARLIER or EARLIEST DAX functions to reference outer filter contexts. Unfortunately, data modelers found these functions difficult to understand and use.

Variables are always evaluated outside the filters your RETURN expression applies. For this reason, when you use a variable within a modified filter context, it achieves the same result as the EARLIEST function. The use of the EARLIER or EARLIEST functions can

therefore be avoided. It means you can now write formulas that are less complex, and that are easier to understand.

Consider the following calculated column definition added to the **Subcategory** table. It evaluates a rank for each product subcategory based on the **Subcategory Sales** column values.

DAX

```
Subcategory Sales Rank =
COUNTROWS(
    FILTER(
        Subcategory,
        EARLIER(Subcategory[Subcategory Sales]) < Subcategory[Subcategory
Sales]
    )
) + 1
```

The EARLIER function is used to refer to the **Subcategory Sales** column value *in the current row context*.

The calculated column definition can be improved by using a variable instead of the EARLIER function. The **CurrentSubcategorySales** variable stores the **Subcategory Sales** column value *in the current row context*, and the RETURN expression uses it within a modified filter context.

DAX

```
Subcategory Sales Rank =
VAR CurrentSubcategorySales = Subcategory[Subcategory Sales]
RETURN
    COUNTROWS(
        FILTER(
            Subcategory,
            CurrentSubcategorySales < Subcategory[Subcategory Sales]
        )
    ) + 1
```

See also

- VAR DAX article
- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community

Feedback

Was this page helpful? Yes No

Get help at Microsoft Q&A

DAX sample model

Article • 12/13/2023

The **Adventure Works DW 2020** Power BI Desktop sample model is designed to support your DAX learning. The model is based on the Adventure Works data warehouse sample for AdventureWorksDW2017—however, the data has been modified to suit the objectives of the sample model.

The sample model does not contain any DAX formulas. It does however support hundreds or even thousands of potential calculation formulas and queries. Some function examples, like those in CALCULATE, DATESBETWEEN, DATESIN PERIOD, IF, and LOOKUPVALUE can be added to the sample model without modification. We're working on including more examples in other function reference articles that work with the sample model.

Scenario

The Adventure Works company represents a bicycle manufacturer that sells bicycles and accessories to global markets. The company has their data warehouse data stored in an Azure SQL Database.

Model structure

The model has seven tables:

Expand table

Table	Description
Customer	Describes customers and their geographic location. Customers purchase products online (Internet sales).
Date	There are three relationships between the **Date** and **Sales** tables, for order date, ship date, and due date. The order date relationship is active. The company's reports

Table	Description
	sales using a fiscal year that commences on July 1 of each year. The table is marked as a date table using the **Date** column.
Product	Stores finished products only.
Reseller	Describes resellers and their geographic location. Reseller on sell products to their customers.
Sales	Stores rows at sales order line grain. All financial values are in US dollars (USD). The earliest order date is July 1, 2017, and the latest order date is June 15, 2020.
Sales Order	Describes sales order and order line numbers, and also the sales channel, which is either **Reseller** or **Internet**. This table has a one-to-one relationship with the **Sales** table.
Sales Territory	Sales territories are organized into groups (North America, Europe, and Pacific), countries, and regions. Only the United States sells products at the region level.

Download sample

Download the Power BI Desktop sample model file here.

Related content

- Learning path: Use DAX in Power BI Desktop
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Feedback

Was this page helpful? Yes No

Separate reports from models in Power BI Desktop

Article • 12/30/2024

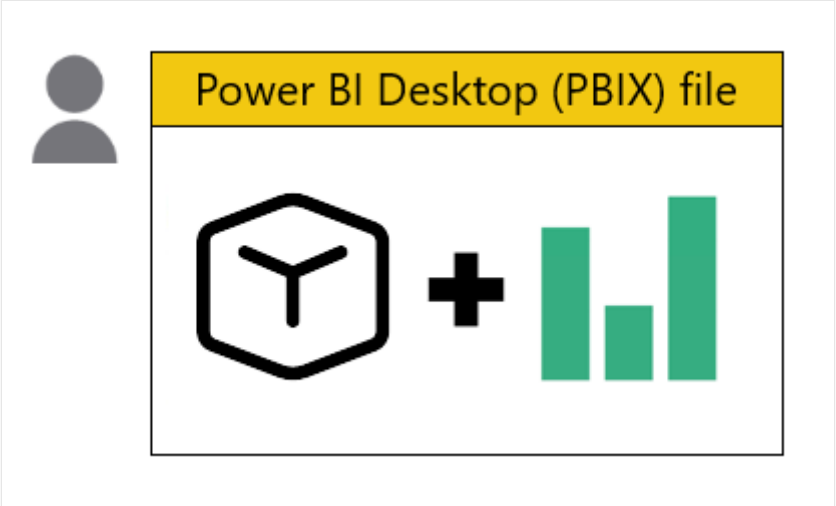
When creating a new Power BI Desktop solution, one of the first tasks you need to do is "get data". Getting data can result in two distinctly different outcomes. It could:

- Create a live connection to an already-published model, which could be either a Power BI semantic model or a remote-hosted Analysis Services model.
- Commence the development of a new model, which could be either an Import, DirectQuery, or Composite model.

This article is concerned with the second scenario. It provides guidance on whether a report and model should be combined into a single Power BI Desktop file.

Single file solution

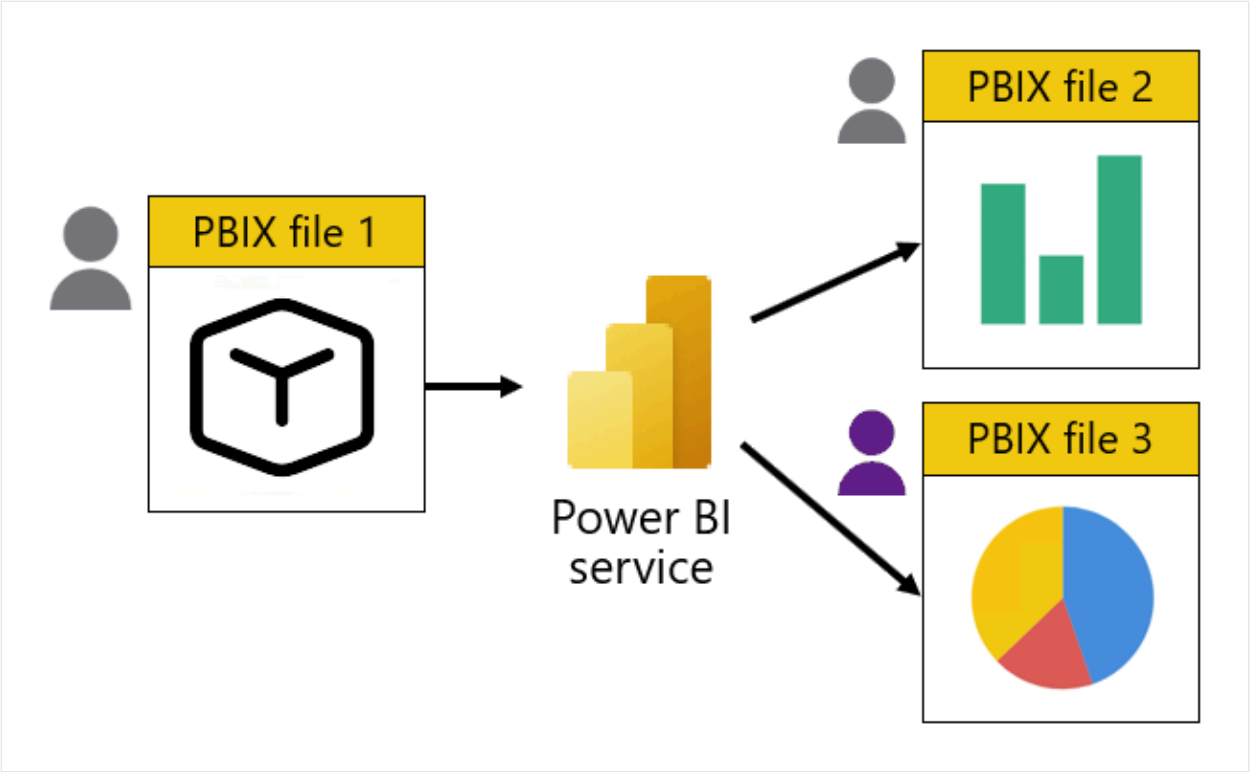
A *single file solution* works well when there's only ever a single report based on the model. In this case, it's likely that both the model and report are the efforts of the same person. We define it as a *Personal BI* solution, though the report could be shared with others. Such solutions can represent role-scoped reports or one-time assessments of a business challenge—often described as *ad hoc* reports.



Separate report files

It makes sense to separate model and report development into separate Power BI Desktop files when:

- Data modelers and report authors are different people.
- It's understood that a model will be the source for multiple reports, now or in the future.



Data modelers can still use the Power BI Desktop report authoring experience to test and validate their model designs. However, just after publishing their file to the Power BI service they should remove the report from the workspace. And, they must remember to remove the report each time they republish and overwrite the semantic model.

Preserve the model interface

Sometimes, model changes are inevitable. Data modelers must take care, then, not the break the model interface. If they do, it's possible that related report visuals or dashboard tiles will break. Broken visuals appear as errors, and they can result in frustration for report authors and consumers. And worse—they can reduce trust in the data.

So, manage model changes carefully. If possible, avoid the following changes:

- Renaming tables, columns, hierarchies, hierarchy levels, or measures.
- Modifying column data types.
- Modifying measure expressions so they return a different data type.
- Moving measures to a different home table. It's because moving a measure could break report-scoped measures that fully qualify measures with their home table name. We don't recommend you write DAX expressions using fully qualified measures names. For more information, see DAX: Column and measure references.

Adding new tables, columns, hierarchies, hierarchy levels, or measures is safe, with one exception: It's possible that a new measure name could collide with a report-scoped measure name. To avoid collision, we recommend report authors adopt a naming convention when defining measures in their reports. They can prefix report-scoped measure names with an underscore or some other character(s).

If you must make breaking changes to your models, we recommend you either:

- View related content for the semantic model in the Power BI service.
- Explore Data lineage view in the Power BI service.

Both options allow you to quickly identify any related reports and dashboards. Data lineage view is probably the better choice because it's easy to see the contact person for each related item. In fact, it's a hyperlink that opens an email message addressed to the contact.

We recommend you contact the owner of each related item to let them know of any planned breaking changes. This way, they can be prepared and ready to fix and republish their reports, helping to minimize downtime and frustration.

Related content

For more information related to this article, check out the following resources:

- Connect to semantic model in the Power BI service from Power BI Desktop
- View related content in the Power BI service
- Data lineage
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Extend visuals with report page tooltips

Article • 12/30/2024

This article targets you as a report author designing Power BI reports. It provides suggestions and recommendations when creating report page tooltips.

Suggestions

Report page tooltips can enhance the experience for your report users. Page tooltips allow your report users to quickly and efficiently gain deeper insights from a visual. They can be associated with different report objects:

- **Visuals**: On a visual-by-visual basis, you can configure which visuals will reveal your page tooltip. Per visual, it's possible to have the visual reveal no tooltip, default to the visual tooltips (configured in the visual fields pane), or use a specific page tooltip.
- **Visual headers**: You can configure specific visuals to display a page tooltip. Your report users can reveal the page tooltip when they hover their cursor over the visual header icon—be sure to educate your users about this icon.

> **Note**
>
> A report visual can only reveal a page tooltip when tooltip page filters are compatible with the visual's design. For example, a visual that groups by *product* is compatible with a tooltip page that filters by *product*.
>
> Page tooltips don't support interactivity. If you want your report users to interact, create a drillthrough page instead.

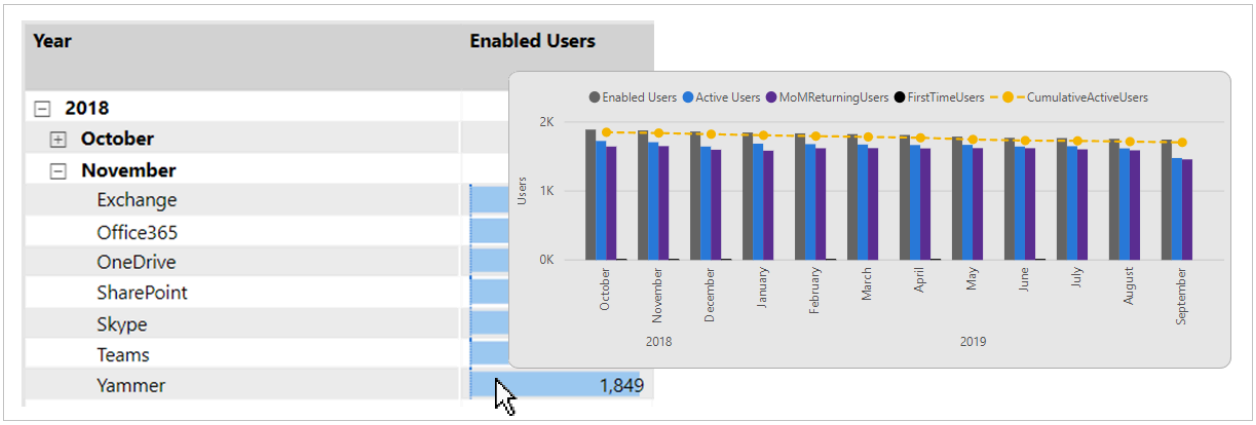
Here are some suggested design scenarios:

- Different perspective
- Add detail
- Add help

Different perspective

A page tooltip can visualize the same data as the source visual. It's done by using the same visual and pivoting groups, or by using different visual types. Page tooltips can also apply different filters than those filters applied to the source visual.

The following example shows what happens when the report user hovers their cursor over the **EnabledUsers** value. The filter context for the value is Yammer in November 2018.

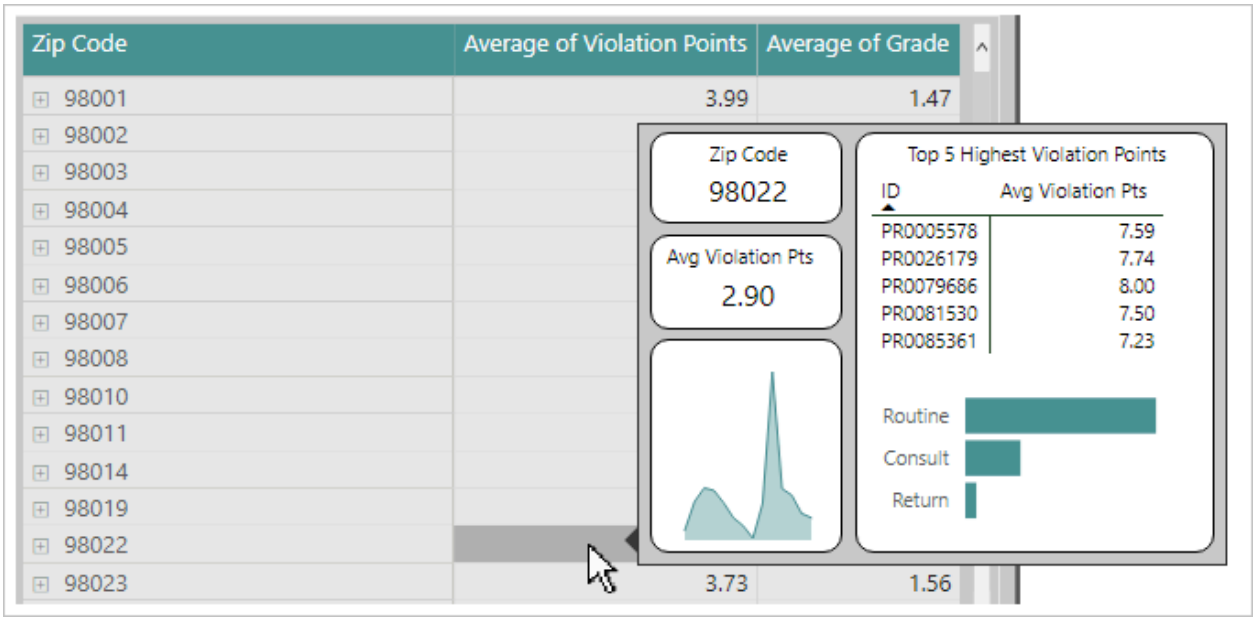


A page tooltip is revealed. It presents a different data visual (line and clustered column chart) and applies a contrasting time filter. Notice that the filter context for the data point is November 2018. Yet the page tooltip displays trend over *a full year of months*.

Add detail

A page tooltip can display additional details and add context.

The following example shows what happens when the report user hovers their cursor over the **Average of Violation Points** value, for zip code 98022.



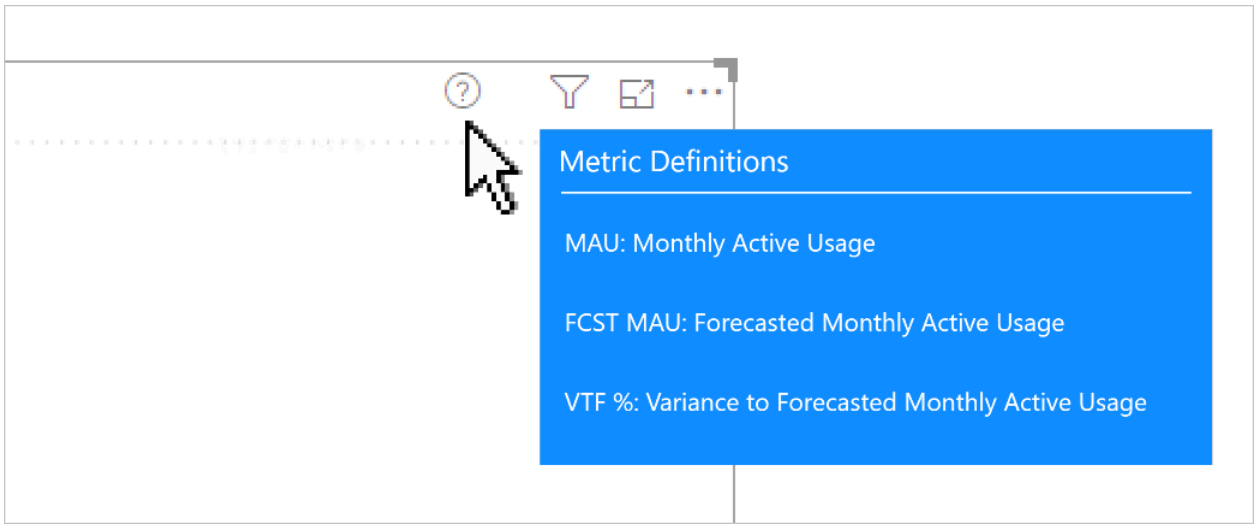
A page tooltip is revealed. It presents specific attributes and statistics for zip code 98022.

Add help

Visual headers can be configured to reveal page tooltips to visual headers. You can add help documentation to a page tooltip by using richly formatted text boxes. It's also possible to add images and shapes.

Interestingly, buttons, images, text boxes, and shapes can also reveal a visual header page tooltip.

The following example shows what happens when the report user hovers their cursor over the visual header icon.



A page tooltip is revealed. It presents rich formatted text in four text boxes, and a shape (line). The page tooltip conveys help by describing each acronym displayed in the visual.

Recommendations

At report design time, we recommend the following practices:

- **Page size**: Configure your page tooltip to be small. You can use the built-in **Tooltip** option (320 pixels wide, 240 pixels high). Or, you can set custom dimensions. Take care not to use a page size that's too large—it can obscure the visuals on the source page.
- **Page view**: In report designer, set the page view to **Actual Size** (page view defaults to **Fit to Page**). This way, you can see the true size of the page tooltip as you design it.
- **Style**: Consider designing your page tooltip to use the same theme and style as the report. This way, users feel like they are in the same report. Or, design a complimentary style for your tooltips, and be sure to apply this style to all page tooltips.
- **Tooltip filters**: Assign filters to the page tooltip so that you can preview a realistic result as you design it. Be sure to remove these filters before you publish your report.

- **Page visibility**: Always hide tooltip pages—users shouldn't navigate directly to them.

Related content

For more information related to this article, check out the following resources:

- Create tooltips based on report pages in Power BI Desktop
- Customizing tooltips in Power BI Desktop
- Use visual elements to enhance Power BI reports
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Use report page drillthrough

Article • 12/30/2024

This article targets you as a report author who designs Power BI reports. It provides suggestions and recommendations when creating report page drillthrough.

It's recommended that you design your report to allow report users to achieve the following flow:

1. View a report page.
2. Identify a visual element to analyze more deeply.
3. Right-click the visual element to drill through.
4. Perform complimentary analysis.
5. Return to the source report page.

Suggestions

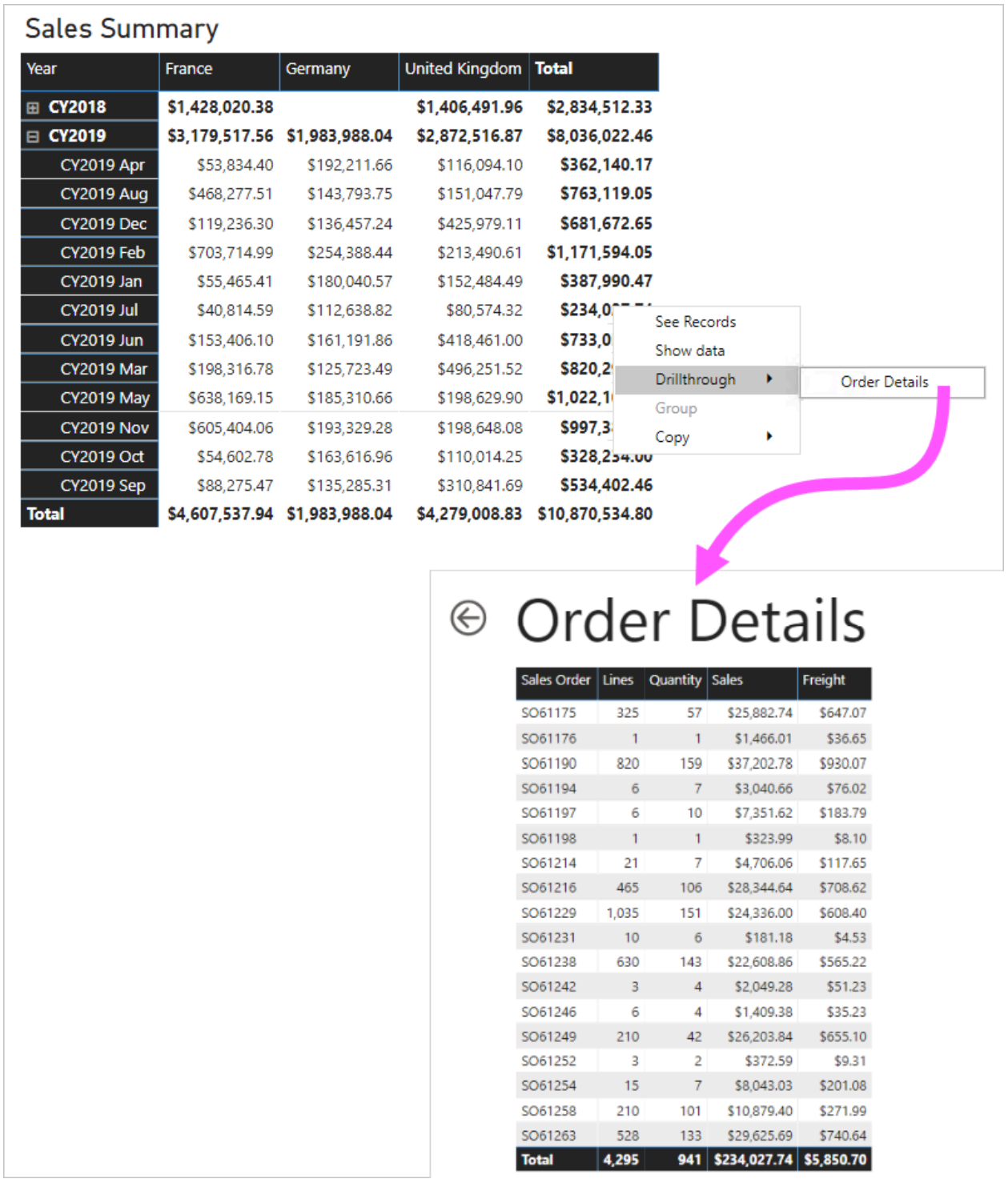
We suggest that you consider two types of drillthrough scenarios:

- Additional depth
- Broader perspective

Additional depth

When your report page displays summarized results, a drillthrough page can lead report users to transaction-level details. This design approach allows them to view supporting transactions, and only when needed.

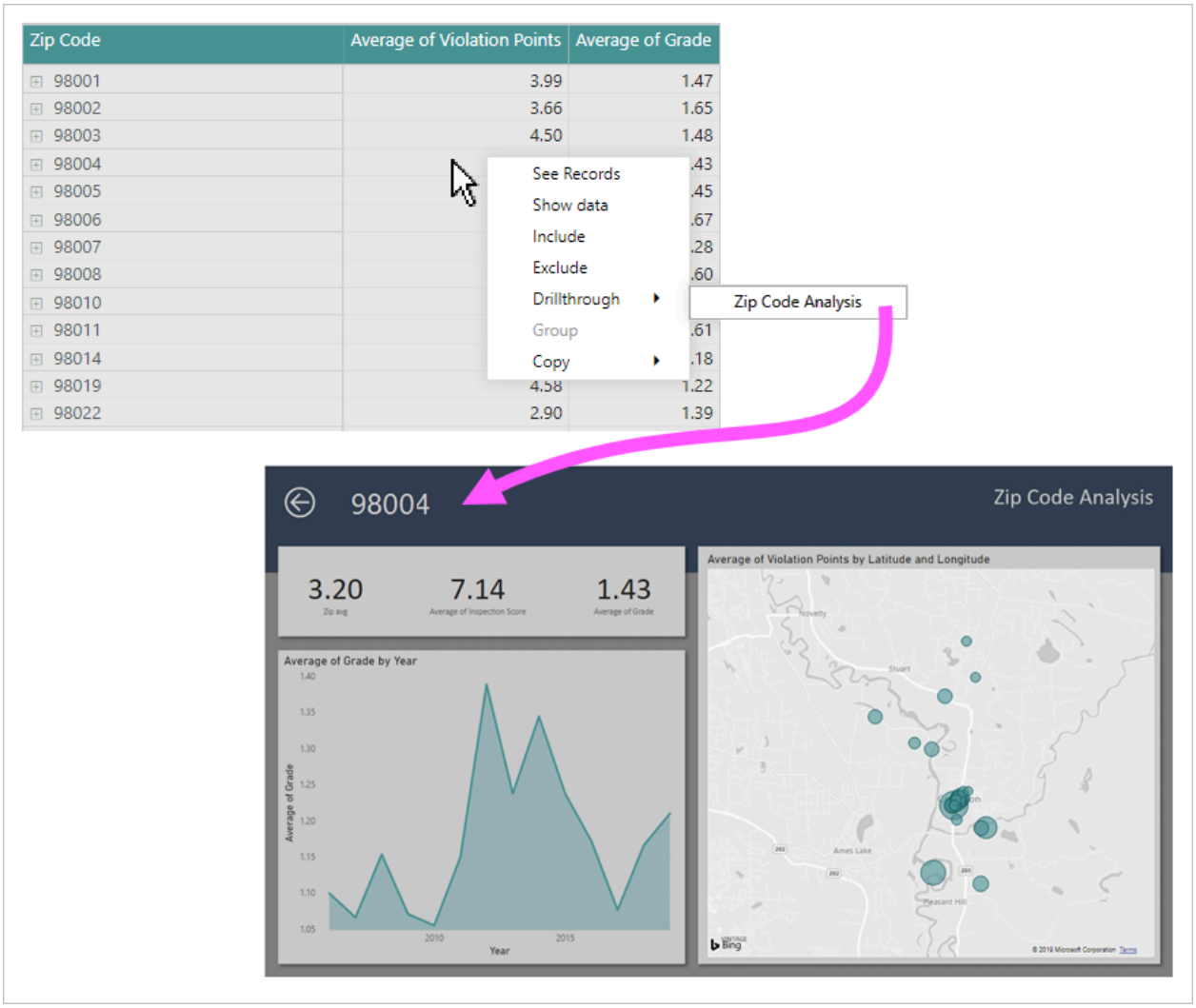
The following example shows what happens when a report user drills through from a monthly sales summary. The drillthrough page contains a detailed list of orders for a specific month.



Broader perspective

A drillthrough page can achieve the opposite of additional depth. This scenario is great for drilling through to a holistic view.

The following example shows what happens when a report user drills through from a zip code. The drillthrough page displays general information about that zip code.



Recommendations

At report design time, we recommend the following practices:

- **Style**: Consider designing your drillthrough page to use the same theme and style as the report. This way, users feel like they are in the same report.
- **Drillthrough filters**: Set drillthrough filters so you can preview a realistic result as you design the drillthrough page. Be sure to remove these filters before you publish the report.
- **Additional capabilities**: A drillthrough page is like any report page. You can even enhance it with additional interactive capabilities, including slicers or filters.
- **Blanks**: Avoid adding visuals that could display BLANK, or produce errors when drillthrough filters are applied.
- **Page visibility**: Consider hiding drillthrough pages. If you decide to keep a drillthrough page visible, be sure to add a button that allows users to clear any previously-set drillthrough filters. Assign a bookmark to the button. The bookmark should be configured to remove all filters.
- **Back button**: A back button is added automatically when you assign a drillthrough filter. It's a good idea to keep it. This way, your report users can easily return to the

source page.

- **Discovery**: Help promote awareness of a drillthrough page by setting visual header icon text, or adding instructions to a text box. You can also design an overlay, as described in this blog post.

> **Tip**
>
> It's also possible to configure drillthrough to your Power BI paginated reports. You can do this be adding links to Power BI reports. Links can define **URL parameters**.

Related content

For more information related to this article, check out the following resources:

- Use drillthrough in Power BI Desktop
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

When to use paginated reports in Power BI

Article • 12/30/2024

This article targets you as a report author who designs reports for Power BI. It provides suggestions to help you choose when to develop Power BI paginated reports.

Power BI paginated reports are optimized for **printing**, or **PDF generation**. They also provide you with the ability to produce highly formatted, print-ready layouts. So, paginated reports are ideal for operational reports, like sales invoices.

In contrast, Power BI reports are optimized for **exploration and interactivity**. Also, they can present your data using a comprehensive range of ultra-modern visuals. Power BI reports, therefore, are ideal for analytic reports, enabling your report users to explore data, and to discover relationships and patterns.

We recommend you consider using a Power BI paginated report when:

- You know the report must be printed, or output as a PDF document.
- Data grid layouts could expand and overflow. Consider that a table, or matrix, in a Power BI report can't dynamically resize to display all data—it will provide scroll bars instead. But, if printed, it won't be possible to scroll to reveal any out-of-view data.
- Power BI paginated features and capabilities work in your favor. Many such report scenarios are described later in this article.

Legacy reports

When you already have SQL Server Reporting Services (SSRS) Report Definition Language (RDL) reports, you can choose to redevelop them as Power BI reports, or migrate them as paginated reports to Power BI. For more information, see Migrate SQL Server Reporting Services reports to Power BI.

Once published to a Power BI workspace, paginated reports are available side by side with Power BI reports. They can then be easily distributed using Power BI apps.

You might consider redeveloping SSRS reports, rather than migrating them. It's especially true for those reports that are intended to deliver analytic experiences. In these cases, Power BI reports will likely deliver better report user experiences.

Paginated report scenarios

There are many compelling scenarios when you might favor developing a Power BI paginated report. Many are features or capabilities not supported by Power BI reports.

- **Print-ready**: Paginated reports are optimized for printing, or PDF generation. When necessary, data regions can expand and overflow to multiple pages in a controlled way. Your report layouts can define margins, and page headers and footers.
- **Render formats**: Power BI can render paginated reports in different formats. Formats include Microsoft Excel, Microsoft Word, Microsoft PowerPoint, PDF, CSV, XML, and MHTML. (The MHTML format is used by the Power BI service to render reports.) Your report users can decide to export in the format that suits them.
- **Precision layout**: You can design highly formatted, print-ready layouts—to the exact size and location configured in fractions of inches, or centimeters.
- **Dynamic layout**: You can produce highly responsive layouts by setting many report properties to use VB.NET expressions. Expressions have access to many core .NET Framework libraries.
- **Render-specific layout**: You can use expressions to modify the report layout based on the rendering format applied. For example, you can design the report to disable toggling visibility (to drill down and drill up) when it's rendered using a non-interactive format, like PDF.
- **Native queries**: You don't need to first publish a Power BI semantic model. It's possible to author native queries (or use stored procedures) for any supported data source. Queries can include parameterization.
- **Graphic query designers**: Power BI Report Builder includes graphic query designers to help you write, and test, your dataset queries.
- **Static datasets**: You can define a dataset, and enter data directly into your report definition. This capability is especially useful to support a demo, or for delivering a proof of concept (POC).
- **Data integration**: You can combine data from different data sources, or with static datasets. It's done by creating custom fields using VB.NET expressions.
- **Parameterization**: You can design highly customized parameterization experiences, including data-driven, and cascading parameters. It's also possible to define parameter defaults. These experiences can be designed to help your report users quickly set appropriate filters. Also, parameters don't need to filter report data; they can be used to support "what-if" scenarios, or dynamic filtering or styling.
- **Image data**: Your report can render images when they're stored in binary format in a data source.
- **Custom code**: You can develop code blocks of VB.NET functions in your report, and use them in any report expression.

- **Subreports**: You can embed other Power BI paginated reports (from the same workspace) into your report.
- **Flexible data grids**: You have fine-grained control of grid layouts by using the tablix data region. It supports complex layouts, too, including nested and adjacent groups. And, it can be configured to repeat headings when printed over multiple pages. Also, it can embed a subreport or other visualizations, including data bars, sparklines, and indicators.
- **Spatial data types**: The map data region can visualize SQL Server spatial data types. So, the GEOGRAPHY and GEOMETRY data types can be used to visualize points, lines, or polygons. It's also possible to visualize polygons defined in ESRI shape files.
- **Modern gauges**: Radial and linear gauges can be used to display KPI values and status. They can even be embedded into grid data regions, repeating within groups.
- **HTML rendering**: You can display richly formatted text when it's stored as HTML.
- **Mail merge**: You can use text box placeholders to inject data values into text. This way, you can produce a mail merge report.
- **Interactivity features**: Interactive features include toggling visibility (to drill down and drill up), links, interactive sorting, and tooltips. You can also add links that drillthrough to Power BI reports, or other Power BI paginated reports. Links can even jump to another location within the same report.
- **Subscriptions**: Power BI can deliver paginated reports on a schedule as emails, with report attachments in any supported format.
- **Per-user layouts**: You can create responsive report layouts based on the authenticated user who opens the report. You can design the report to filter data differently, hide data regions or visualizations, apply different formats, or set user-specific parameter defaults.

Related content

For more information related to this article, check out the following resources:

- What are paginated reports in Power BI?
- Migrate SQL Server Reporting Services reports to Power BI
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Data retrieval guidance for paginated reports

Article • 12/30/2024

This article targets you as a report author designing Power BI paginated reports. It provides recommendations to help you design effective and efficient data retrieval.

Data source types

Paginated reports natively support both relational and analytic data sources. These sources are further categorized, as either cloud-based or on-premises. On-premises data sources—whether hosted on-premises, or in a virtual machine—require a data gateway so Power BI can connect. Cloud-based means that Power BI can connect directly using an Internet connection.

If you can choose the data source type (possibly the case in a new project), we recommend that you use cloud-based data sources. Paginated reports can connect with lower network latency, especially when the data sources reside in the same region as your Power BI tenant. Also, it's possible to connect to these sources by using Single Sign-On (SSO). It means the report user's identity can flow to the data source, allowing per-user row-level permissions to be enforced. Currently, SSO is only supported for on-premises data sources SQL Server and Oracle (see Supported data sources for Power BI paginated reports).

> **Note**
>
> While it's currently not possible to connect to on-premises databases using SSO, you can still enforce row-level permissions. It's done by passing the **UserID** built-in field to a dataset query parameter. The data source will need to store User Principal Name (UPN) values in a way that it can correctly filter query results.
>
> For example, consider that each salesperson is stored as a row in the **Salesperson** a table. The table has columns for UPN, and also the salesperson's sales region. At query time, the table is filtered by the UPN of the report user, and it's also related to sales facts using an `INNER JOIN`. This way, the query effectively filters sales fact rows to those of the report user's sales region.

Relational data sources

Generally, relational data sources are well suited to operational style reports, like sales invoices. They're also suited for reports that need to retrieve very large datasets (in excess of 10,000 rows). Relational data sources can also define stored procedures, which can be executed by report datasets. Stored procedures deliver several benefits:

- Parameterization
- Encapsulation of programming logic, allowing for more complex data preparation (for example, temporary tables, cursors, or scalar user-defined functions)
- Improved maintainability, allowing stored procedure logic to be easily updated. In some cases, it can be done without the need to modify and republish paginated reports (providing column names and data types remain unchanged).
- Better performance, as their execution plans are cached for reuse
- Reuse of stored procedures across multiple reports

In Power BI Report Builder, you can use the relational query designer to graphically construct a query statement—but only for Microsoft data sources.

Analytic data sources

Analytic data sources—also known as *data models* or just *models*—are well suited to both operational and analytic reports, and can deliver fast summarized query results even over very large data volumes. Model measures and KPIs can encapsulate complex business rules to achieve summarization of data. These data sources, however, aren't suited to reports that need to retrieve very large volumes of data (in excess of 10,000 rows).

In Power BI Report Builder, you have a choice of two query designers: The Analysis Services DAX query designer, and the Analysis Services MDX query designer. These designers can be used for Power BI semantic model data sources, or any SQL Server Analysis Services or Azure Analysis Services model—tabular or multidimensional.

We recommend that you use the DAX query designer—providing it entirely meets your query needs. If the model doesn't define the measures you need, you'll need to switch to query mode. In this mode, you can customize the query statement by adding expressions (to achieve summarization).

The MDX query designer requires your model to include measures. The designer has two capabilities not supported by the DAX query designer. Specifically, it allows you to:

- Define query-level calculated members (in MDX).
- Configure data regions to request server aggregates in non-detail groups. If your report needs to present summaries of semi- or non-additive measures (like time intelligence calculations, or distinct counts), it will likely be more efficient to use

server aggregates than to retrieve low-level detail rows and have the report compute summarizations.

Query result size

In general, it's best practice to retrieve only the data required by your report. So, don't retrieve columns or rows that aren't required by the report.

To limit rows, you should always apply the most restrictive filters, and define aggregate queries. Aggregate queries group and summarize source data to retrieve higher-grain results. For example, consider that your report needs to present a summary of salesperson sales. Instead of retrieving all sales order rows, create a dataset query that groups by salesperson, and summarizes sales for each group.

Expression-based fields

It's possible to extend a report dataset with fields based on expressions. For example, if your dataset sources customer first name and last name, you might want a field that concatenates the two fields to produce the customer full name. To achieve this calculation, you have two options. You can:

- Create a *calculated field*, which is a dataset field based on an expression.
- Inject an expression directly into the dataset query (using the native language of your data source), which results in a regular dataset field.

We recommend the latter option, whenever possible. There are two good reasons why injecting expressions directly into your dataset query is better:

- It's possible your data source is optimized to evaluate the expression more efficiently than Power BI (it's especially the case for relational databases).
- Report performance is improved because there's no need for Power BI to materialize calculated fields prior to report rendering. Calculated fields can noticeably extend report render time when datasets retrieve a large number of rows.

Field names

When you create a dataset, its fields are automatically named after the query columns. It's possible these names aren't friendly or intuitive. It's also possible that source query column names contain characters prohibited in Report Definition Language (RDL) object

identifiers (like spaces and symbols). In this case, the prohibited characters are replaced with an underscore character (_).

We recommend that you first verify that all field names are friendly, concise, yet still meaningful. If not, we suggest you rename them *before* you commence the report layout. It's because renamed fields don't ripple changes through to the expressions used in your report layout. If you do decide to rename fields after you've commenced the report layout, you'll need to find and update all broken expressions.

Filter vs parameter

It's likely that your paginated report designs will have report parameters. Report parameters are commonly used to prompt your report user to filter the report. As a paginated report author, you have two ways to achieve report filtering. You can map a report parameter to:

- A dataset *filter*, in which case the report parameter value(s) are used to filter the data already retrieved by the dataset.
- A dataset *parameter*, in which case the report parameter value(s) are injected into the native query sent to the data source.

> ⓘ **Note**
>
> All report datasets are cached on a *per-session basis* for up to 10 minutes *beyond their last use*. A dataset can be re-used when submitting new parameter values (filtering), rendering the report in a different format, or interacting with the report design in some way, like toggling visibility, or sorting.

Consider, then, an example of a sales report that has a single report parameter to filter the report by a single year. The dataset retrieves sales for *all years*. It does so because the report parameter maps to the dataset filters. The report displays data for the requested year, which is a subset of the dataset data. When the report user changes the report parameter to a different year—and then views the report—Power BI doesn't need to retrieve any source data. Instead, it applies a different filter to the already-cached dataset. Once the dataset is cached, filtering can be very fast.

Now, consider a different report design. This time the report design maps the sales year report parameter to a dataset parameter. This way, Power BI injects the year value into the native query, and the dataset retrieves data only for that year. Each time the report user changes the year report parameter value—and then views the report—the dataset retrieves a new query result for just that year.

Both design approaches can filter report data, and both designs can work well for your report designs. An optimized design, however, will depend on the anticipated volumes of data, data volatility, and the anticipated behaviors of your report users.

We recommend *dataset filtering* when you anticipate a different subset of the dataset rows will be reused many times (thereby saving rendering time because new data doesn't need to be retrieved). In this scenario, you recognize that the cost of retrieving a larger dataset can be traded off against the number of times it will be reused. So, it's helpful for queries that are time consuming to generate. But take care—caching large datasets on a per-user basis may negatively impact on performance, and capacity throughput.

We recommend *dataset parameterization* when you anticipate it's unlikely that a different subset of dataset rows will be requested—or, when the number of the dataset rows to be filtered is likely to be very large (and inefficient to cache). This design approach work well, too, when your data store is volatile. In this case, each report parameter value change will result in the retrieval of up-to-date data.

Non-native data sources

If you need to develop paginated reports based on data sources that aren't natively supported by paginated reports, you should first develop a data model in Power BI Desktop. That way, you can connect to hundreds of data sources supported by Power BI. Once published to the Power BI service, you can then develop a paginated report that connects to the Power BI semantic model.

Data integration

If you need to combine data from multiple data sources, you have two options:

- **Combine report datasets**: If the data sources are natively supported by paginated reports, you can consider creating calculated fields that use the Lookup or LookupSet Report Builder functions.
- **Develop a Power BI Desktop model**: It's likely more efficient, however, that you develop a data model in Power BI Desktop. You can use Power Query to combine queries based on any supported data source. Once published to the Power BI service, you can then develop a paginated report that connects to the Power BI semantic model.

Network latency

Network latency can impact report performance by increasing the time required for requests to reach the Power BI service, and for responses to be delivered. Tenants in Power BI are assigned to a specific region.

> **Tip**
>
> To determine where your tenant is located, see **Where is my Power BI tenant located?**

When users from a tenant access the Power BI service, their requests always route to this region. As requests reach the Power BI service, the service may then send additional requests—for example, to the underlying data source, or a data gateway—which are also subject to network latency. In general, to minimize the impact of network latency, strive to keep data sources, gateways, and your Power BI capacity as close as possible. Preferably, they reside within the same region. If network latency is an issue, try locating gateways and data sources closer to your Power BI capacity by placing them inside cloud-hosted virtual machines.

SQL Server complex data types

Because SQL Server is an on-premises data source, Power BI must connect via a gateway. The gateway, however, doesn't support retrieving data for complex data types. Complex data types include built-in types like the GEOMETRY and GEOGRAPHY spatial data types, and hierarchyid. They can also include user-defined types implemented through a class of an assembly in the Microsoft.NET Framework common language runtime (CLR).

Plotting spatial data and analytics in the map visualization requires SQL Server spatial data. Therefore, it's not possible to work with the map visualization when SQL Server is your data source. To be clear, it will work if your data source is Azure SQL Database because Power BI doesn't connect via a gateway.

Data-related images

Images can be used to add logos or pictures to your report layout. When images relate to the rows retrieved by a report dataset, you have two options:

- It's possible that image data can also be retrieved from your data source (if already stored in a table).

- When the images are stored on a web server, you can use a dynamic expression to create the image URL path.

For more information and suggestions, see Image guidance for paginated reports.

Redundant data retrieval

It's possible your report retrieves redundant data. This can happen when you delete dataset query fields, or the report has unused datasets. Avoid these situations, as they result in an unnecessary burden on your data sources, the network, and Power BI capacity resources.

Deleted query fields

On the **Fields** page of the **Dataset Properties** window, it's possible to delete dataset *query fields* (query fields map to columns retrieved by the dataset query). However, Report Builder doesn't remove corresponding columns from the dataset query.

If you need to delete query fields from your dataset, we recommend you remove the corresponding columns from the dataset query. Report Builder will automatically remove any redundant query fields. If you do happen to delete query fields, be sure to also modify the dataset query statement to remove the columns.

Unused datasets

When a report is run, all datasets are evaluated—even if they're not bound to report objects. For this reason, be sure to remove any test or development datasets before you publish a report.

Related content

For more information related to this article, check out the following resources:

- Supported data sources for Power BI paginated reports
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Performance and scalability considerations for paginated reports in Power BI service

Article • 04/10/2025

This article summarizes scalability and performance differences between paginated reports in the Power BI service versus SQL Server Reporting Services/Power BI Report Server. The article is intended for users migrating paginated reports from on-premises to the Power BI service. Additionally, it also provides insights on how to optimize the performance of paginated reports in the service.

Scalability considerations

When comparing the scalability of paginated reports in the Power BI service versus on-premises, there are two main considerations: whether the execution environment is optimized to handle the expressions you are using and the volume of data. These are some of the factors that are crucial in improving the scalability of your paginated reports in the service.

Execution environment

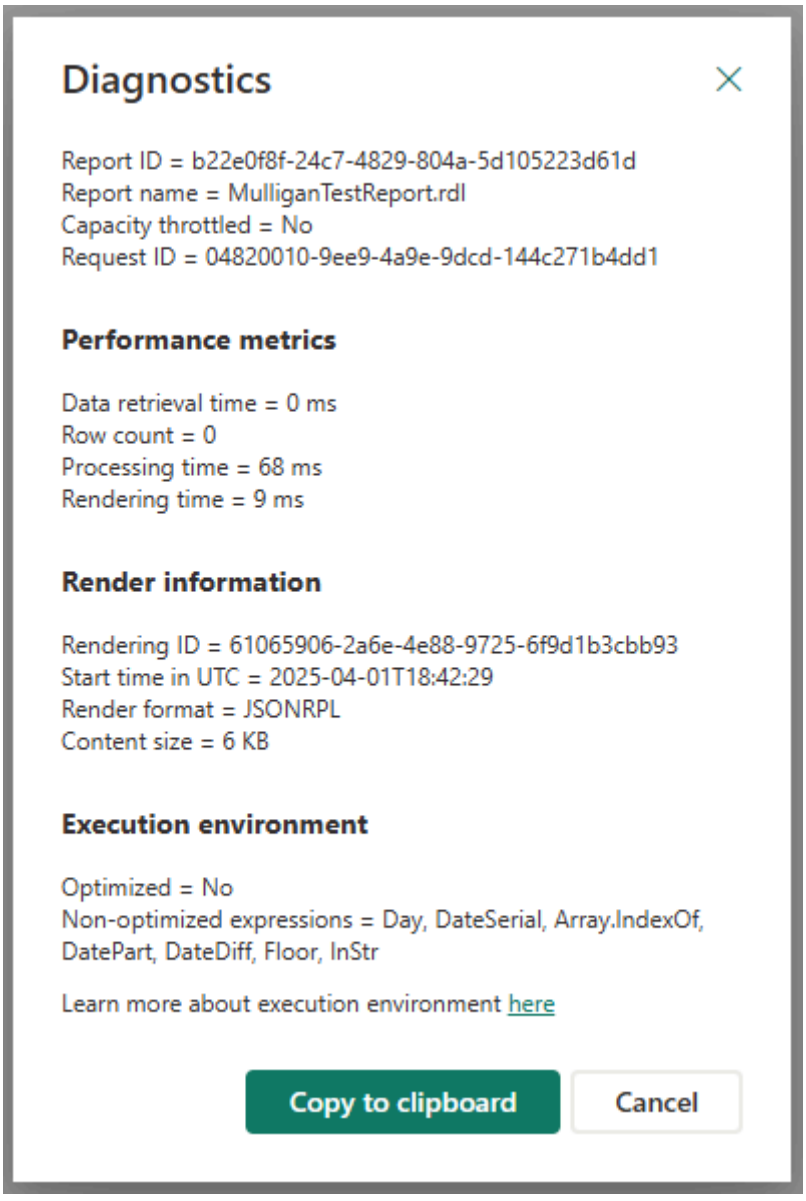
A paginated report can run in two different execution environments: the **standard environment** and the **optimized environment**. The primary difference between these environments is that optimized environment can handle higher data volumes compared to the standard execution environment. The expressions used in the report determine the execution environment, and users are unable to modify it through any settings or configurations. The only way to ensure a report runs in an optimized environment is to remove any nonsupported expressions because the optimized environment only supports a specific subset of report expressions.

If your report includes nonsupported expressions, consider removing or updating those expressions to ensure the report runs in an optimized environment. One approach is to move calculations into the dataset query. Additionally, another advantage of using calculated fields in the semantic model data sources is that other reports can also use them. Power Query is another option for performing advanced calculations and data processing operations outside of the paginated report. More information can be found here.

> **Note**
>
> One example of unsupported expression is the usage of calculated fields in the RDL, such as this: `If(Weekday(Fields!SalesDate.Value) > 5, "Relax", "Work")`. Weekday is a function that is not yet optimized. Rather than using a report expression this could be

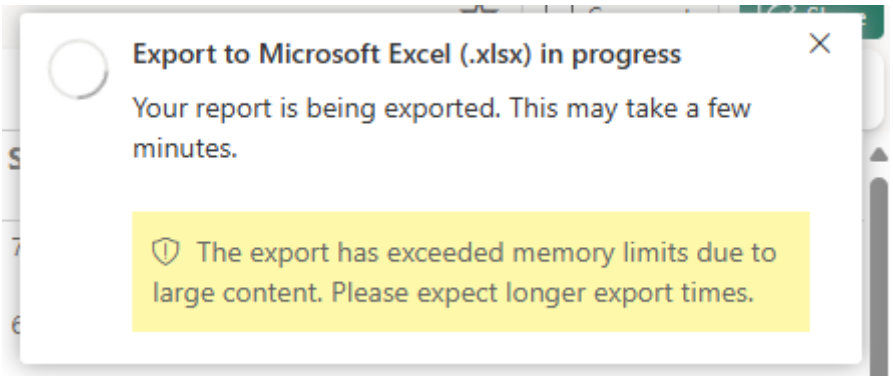
calculated as a part of a SQL query. For SQL Server/ Azure SQL that could be done using the Transact SQL functions DATEPART and IF..ELSE.

To check if a report was run in the optimized environment, click on the Diagnostics button in the paginated report and check the 'Execution environment' section. If the report is running in the optimized environment, then 'Optimized' shows 'Yes.' If the report is running in the standard, nonoptimized environment, the Execution environment section shows a list of nonsupported report expressions. More details on the performance metrics displayed on the Diagnostics card can be found here.



Large data volumes

Both the standard and optimized execution environments for paginated reports have specific data volume limits that, when exceeded, significantly slow down report processing. When performance slows down during the export report scenario, the user sees the warning shown in this image:



This issue typically occurs in two scenarios:

- Data dumps of tables with no groupings and many columns/rows: these data dumps require significant resources when exporting the paginated report to Excel. Alternatively, consider exporting the report to CSV format, which is less intensive.
- Aggregation of large volumes of data during report processing: instead of aggregating large data volumes inside paginated reports, perform those aggregations in the query used by a paginated reports dataset. This approach reduces the amount of data processed by paginated reports and typically enhances overall performance because data sources such as SQL Server or Power BI Semantic Model are highly optimized for data aggregation.

The recommended threshold for data volume in a standard execution environment before report processing slows significantly is **1,000,000 rows with 15-20 columns**, consisting of a mixture of numeric, date, and small to medium length string data types. **This limit doubles for optimized execution environments**. Adding image data (`byte[]`) to data dumps significantly increases data volumes. Similarly, selecting more or fewer columns than specified would also affect data volumes, as would the mix of data types.

Performance considerations

Even for medium to low-sized data volumes, there are more factors that can impact the performance of paginated reports in the service for these scenarios:

- Processing report *parameters* each time a report is opened in the Power BI service.
- Rendering the report whenever a user clicks on the "View report" button.
- Retrieving data from various data sources

Best practices to improve report parameters processing

Report parameters can be backed by datasets with available and default values, which is often the case for multiple parameters. Parameters can also depend on other parameters such as

cascading parameters. As a result, parameter queries are executed sequentially. When a report has cascading parameters, each parameterized query runs each time subsequent values are selected.

Given that parameter values are usually static and don't change frequently, SQL Server Reporting Services ('SSRS') uses this advantage to offer robust support for dataset caches (as explained here), which improves report parameter processing in SSRS. However, paginated reports in the service don't support dataset caches. Therefore, users should follow these best practices to improve execution time for parameter queries for paginated reports in the service:

- Avoid using parameter queries that run against on-premises data sources accessed via a Power BI Gateway. Instead, use a Power BI semantic model as a cache.
- Reduce the amount of data fetched by parameter queries to ensure efficiency. Fetching more than 1,000 values can be time-consuming and may not be user-friendly in the UI.
- Use the EnterData data extension for a static set of parameter values in order to embed them in the RDL. Besides providing parameter values, these datasets can also be utilized in your report without running a query.

Best practices to improve paginated report rendering

Similar to the guidelines provided in the scalability section, avoid using live queries for on-premises data sources that are accessed via a Power BI Gateway. To improve report rendering in paginated reports, use a Power BI semantic model as a cache instead. Rather than aggregating data within the paginated report, perform aggregations inside the dataset query to reduce data volume and enhance performance. Use data sources such as Azure SQL or Power BI semantic models which are highly optimized for processing aggregates.

Additionally, avoid using dataset or table/matrix filters, as paginated reports fetch all data first, then apply filters internally, affecting performance. Instead, use filters in dataset queries whenever possible.

Best practices for efficient data retrieval

Multi-geo environments, where the report and the data source like a semantic model are in different regions, slow down data retrieval. More guidance for efficient data retrieval for paginated reports can be found here.

Related content

For more information related to this article, check out the following resources:

- What are paginated reports in Power BI?

- Power BI paginated reports diagnostics
- Paginated reports data retrieval guidance
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Image use guidance for paginated reports

Article • 12/30/2024

This article targets you as a report author designing Power BI paginated reports. It provides suggestions when working with images. Commonly, images in report layouts can display a graphic like a company logo, or pictures.

Images can be stored in three different locations:

- Within the report (embedded)
- On a web server (external)
- In a database, which can be retrieved by a dataset

They can then be used in a variety of scenarios in your report layouts:

- Free-standing logo, or picture
- Pictures associated with rows of data
- Background for certain report items:
  - Report body
  - Textbox
  - Rectangle
  - Tablix data region (table, matrix, or list)

Suggestions

Consider the following suggestions to deliver professional report layouts, ease of maintenance, and optimized report performance:

- **Use smallest possible size**: We recommend you prepare images that are small in size, yet still look sharp, and crisp. It's all about a balance between quality and size. Consider using a graphics editor (like MS Paint) to reduce the image file size.

- **Avoid embedded images**: First, embedded images can bloat the report file size, which can contribute to slower report rendering. Second, embedded images can quickly become a maintenance nightmare if you need to update many report images (as might be the case should your company logo change).

- **Use web server storage**: Storing images on a web server is a good option, especially for the company logo, which may be sourced from the company website. However, take care if your report users will access reports outside your

network. In this case, be sure that the images are available over the Internet and do not require authentication or additional sign-in to access the image. Images stored on a web server must not exceed 4 MB in size or they will not load in the Power BI service.

When images relate to your data (like pictures of your salespeople), name image files so a report expression can dynamically produce the image URL path. For example, you could name the salespeople pictures using each salesperson's employee number. Providing the report dataset retrieves the employee number, you can write an expression to produce the full image URL path.

- **Use database storage**: When a relational database stores image data, it makes sense to source the image data directly from the database tables—especially when the images are not too large.
- **Appropriate background images**: If you decide to use background images, take care not to distract the report user from your report data.

  Also, be sure to use *watermark styled images*. Generally, watermark styled images have a transparent background (or have the same background color used by the report). They also use faint colors. Common examples of watermark styled images include the company logo, or sensitivity labels like "Draft" or "Confidential".

Related content

For more information related to this article, check out the following resources:

- What are paginated reports in Power BI?
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Use cascading parameters in paginated reports

Article • 12/30/2024

This article targets you as a report author designing Power BI paginated reports. It provides scenarios for designing cascading parameters. Cascading parameters are report parameters with dependencies. When a report user selects a parameter value (or values), it's used to set available values for another parameter.

> **Note**
>
> An introduction to cascading parameters, and how to configure them, isn't covered in this article. If you're not completely familiar with cascading parameters, we recommend you first read Add Cascading Parameters to a Report in Power BI Report Builder.

Design scenarios

There are two design scenarios for using cascading parameters. They can be effectively used to:

- Filter *large sets* of items
- Present *relevant* items

Example database

The examples presented in this article are based on an Azure SQL Database. The database records sales operations, and contains various tables storing resellers, products, and sales orders.

A table named `Reseller` stores one record for each reseller, and it contains many thousands of records. The `Reseller` table has these columns:

- ResellerCode (integer)
- ResellerName
- Country-Region
- State-Province
- City
- PostalCode

There's a table named `Sales`, too. It stores sales order records, and has a foreign key relationship to the `Reseller` table, on the `ResellerCode` column.

Example requirement

There's a requirement to develop a Reseller Profile report. The report must be designed to display information for a single reseller. It's not appropriate to have the report user enter a reseller code, as they rarely memorize them.

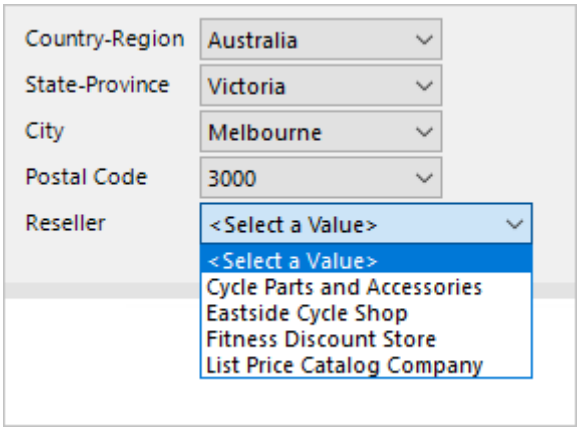
Filter large sets of items

Let's take a look at three examples to help you limit large sets of available items, like resellers. They are:

- Filter by related columns
- Filter by a grouping column
- Filter by search pattern

Filter by related columns

In this example, the report user interacts with five report parameters. They must select country-region, state-province, city, and then postal code. A final parameter then lists resellers that reside in that geographic location.



Here's how you can develop the cascading parameters:

1. Create the five report parameters, ordered in the correct sequence.

2. Create the **CountryRegion** dataset that retrieves distinct country-region values, using the following query statement:

   ```
   SQL
   ```

```
SELECT DISTINCT
  [Country-Region]
FROM
  [Reseller]
ORDER BY
  [Country-Region]
```

3. Create the **StateProvince** dataset that retrieves distinct state-province values for the selected country-region, using the following query statement:

```
SQL

SELECT DISTINCT
  [State-Province]
FROM
  [Reseller]
WHERE
  [Country-Region] = @CountryRegion
ORDER BY
  [State-Province]
```

4. Create the **City** dataset that retrieves distinct city values for the selected country-region and state-province, using the following query statement:

```
SQL

SELECT DISTINCT
  [City]
FROM
  [Reseller]
WHERE
  [Country-Region] = @CountryRegion
  AND [State-Province] = @StateProvince
ORDER BY
  [City]
```

5. Continue this pattern to create the **PostalCode** dataset.

6. Create the **Reseller** dataset to retrieve all resellers for the selected geographic values, using the following query statement:

```
SQL

SELECT
  [ResellerCode],
  [ResellerName]
FROM
  [Reseller]
```

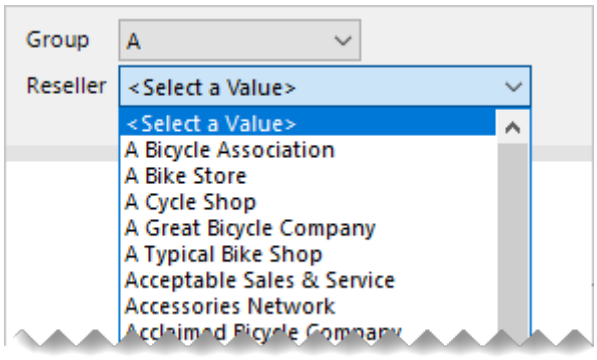
```
WHERE
  [Country-Region] = @CountryRegion
  AND [State-Province] = @StateProvince
  AND [City] = @City
  AND [PostalCode] = @PostalCode
ORDER BY
  [ResellerName]
```

7. For each dataset except the first, map the query parameters to the corresponding report parameters.

> ⓘ Note
>
> All query parameters (prefixed with the @ symbol) shown in these examples could be embedded within `SELECT` statements, or passed to stored procedures.
>
> Generally, stored procedures are a better design approach. It's because their query plans are cached for quicker execution, and they allow you develop more sophisticated logic, when needed. However, they aren't currently supported for gateway relational data sources, which means SQL Server, Oracle, and Teradata.
>
> Lastly, you should always ensure suitable indexes exist to support efficient data retrieval. Otherwise, your report parameters could be slow to populate, and the database could become overburdened. For more information about SQL Server indexing, see SQL Server Index Architecture and Design Guide.

Filter by a grouping column

In this example, the report user interacts with a report parameter to select the first letter of the reseller. A second parameter then lists resellers when the name commences with the selected letter.



Here's how you can develop the cascading parameters:

1. Create the **ReportGroup** and **Reseller** report parameters, ordered in the correct sequence.

2. Create the **ReportGroup** dataset to retrieve the first letters used by all resellers, using the following query statement:

   SQL

   ```
   SELECT DISTINCT
     LEFT([ResellerName], 1) AS [ReportGroup]
   FROM
     [Reseller]
   ORDER BY
     [ReportGroup]
   ```

3. Create the **Reseller** dataset to retrieve all resellers that commence with the selected letter, using the following query statement:

   SQL

   ```
   SELECT
     [ResellerCode],
     [ResellerName]
   FROM
     [Reseller]
   WHERE
     LEFT([ResellerName], 1) = @ReportGroup
   ORDER BY
     [ResellerName]
   ```

4. Map the query parameter of the **Reseller** dataset to the corresponding report parameter.

It's more efficient to add the grouping column to the `Reseller` table. When persisted and indexed, it delivers the best result. For more information, see Specify Computed Columns in a Table.

SQL

```
ALTER TABLE [Reseller]
ADD [ReportGroup] AS LEFT([ResellerName], 1) PERSISTED
```

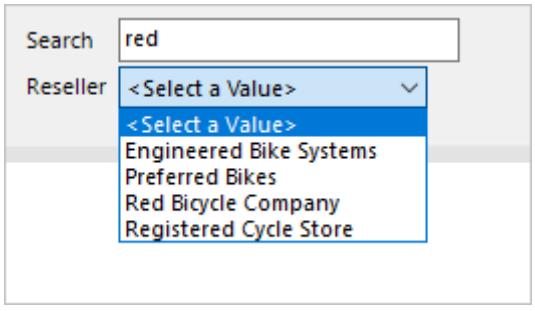
This technique can deliver even greater potential. Consider the following script that adds a new grouping column to filter resellers by *pre-defined bands of letters*. It also creates an index to efficiently retrieve the data required by the report parameters.

SQL

```
ALTER TABLE [Reseller]
ADD [ReportGroup2] AS CASE
  WHEN [ResellerName] LIKE '[A-C]%' THEN 'A-C'
  WHEN [ResellerName] LIKE '[D-H]%' THEN 'D-H'
  WHEN [ResellerName] LIKE '[I-M]%' THEN 'I-M'
  WHEN [ResellerName] LIKE '[N-S]%' THEN 'N-S'
  WHEN [ResellerName] LIKE '[T-Z]%' THEN 'T-Z'
  ELSE '[Other]'
END PERSISTED
GO

CREATE NONCLUSTERED INDEX [Reseller_ReportGroup2]
ON [Reseller] ([ReportGroup2]) INCLUDE ([ResellerCode], [ResellerName])
GO
```

Filter by search pattern

In this example, the report user interacts with a report parameter to enter a search pattern. A second parameter then lists resellers when the name contains the pattern.



Here's how you can develop the cascading parameters:

1. Create the **Search** and **Reseller** report parameters, ordered in the correct sequence.
2. Create the **Reseller** dataset to retrieve all resellers that contain the search text, using the following query statement:

   SQL

   ```
   SELECT
     [ResellerCode],
     [ResellerName]
   FROM
     [Reseller]
   WHERE
     [ResellerName] LIKE '%' + @Search + '%'
   ORDER BY
     [ResellerName]
   ```

3. Map the query parameter of the **Reseller** dataset to the corresponding report parameter.

> **Tip**
>
> You can improve upon this design to provide more control for your report users. It lets them define their own pattern matching value. For example, the search value "red%" will filter to resellers with names that *commence* with the characters "red".
>
> For more information, see **LIKE (Transact-SQL)**.

Here's how you can let the report users define their own pattern.

SQL

```
WHERE
  [ResellerName] LIKE @Search
```

Many non-database professionals, however, don't know about the percentage (%) wildcard character. Instead, they're familiar with the asterisk (*) character. By modifying the `WHERE` clause, you can let them use this character.

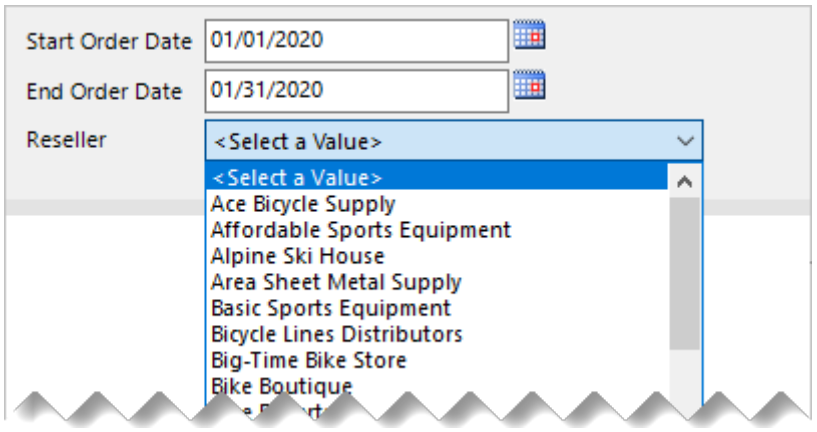
SQL

```
WHERE
  [ResellerName] LIKE SUBSTITUTE(@Search, '%', '*')
```

Present relevant items

In this scenario, you can use fact data to limit available values. Report users will be presented with items where activity has been recorded.

In this example, the report user interacts with three report parameter. The first two set a date range of sales order dates. The third parameter then lists resellers where orders have been created during that time period.



Here's how you can develop the cascading parameters:

1. Create the **OrderDateStart**, **OrderDateEnd**, and **Reseller** report parameters, ordered in the correct sequence.

2. Create the **Reseller** dataset to retrieve all resellers that created orders in the date period, using the following query statement:

   SQL

   ```
   SELECT DISTINCT
     [r].[ResellerCode],
     [r].[ResellerName]
   FROM
     [Reseller] AS [r]
   INNER JOIN [Sales] AS [s]
     ON [s].[ResellerCode] = [r].[ResellerCode]
   WHERE
     [s].[OrderDate] >= @OrderDateStart
     AND [s].[OrderDate] < DATEADD(DAY, 1, @OrderDateEnd)
   ORDER BY
     [r].[ResellerName]
   ```

Recommendations

We recommend you design your reports with cascading parameters, whenever possible. It's because they:

- Provide intuitive and helpful experiences for your report users
- Are efficient, because they retrieve smaller sets of available values

Be sure to optimize your data sources by:

- Using stored procedures, whenever possible
- Adding appropriate indexes for efficient data retrieval

- Materializing column values—and even rows—to avoid expensive query-time evaluations

Related content

For more information related to this article, check out the following resources:

- Report parameters in Power BI Report Builder
- Add Cascading Parameters to a Report (Report Builder)
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Avoid blank pages when printing paginated reports

Article • 12/30/2024

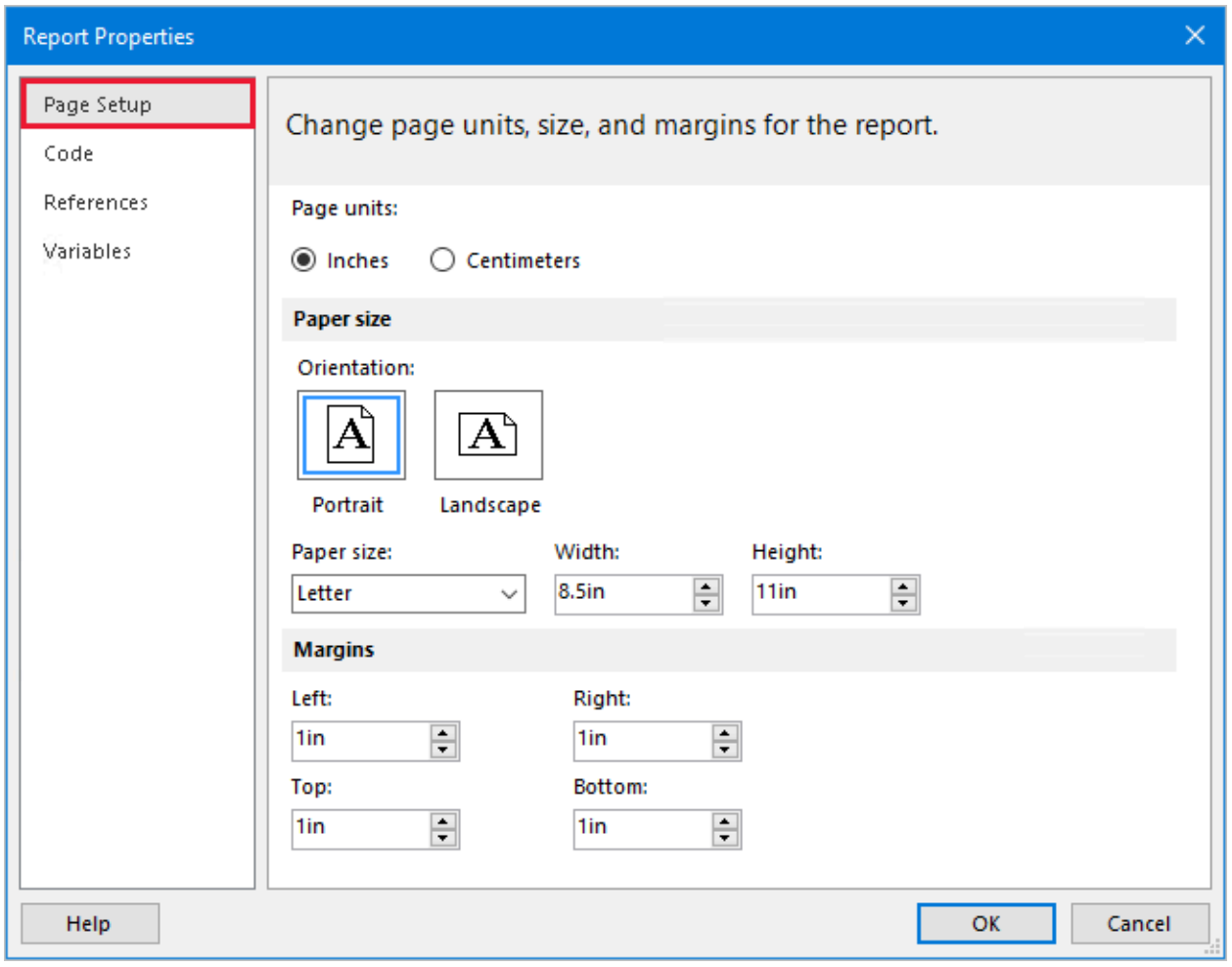
This article targets you as a report author designing Power BI paginated reports. It provides recommendations to help you avoid blank pages when your report is exported to a hard-page format—like PDF or Microsoft Word—or, is printed.

Page setup

Report page size properties determine the page orientation, dimensions, and margins. Access these report properties by:

- Using the report **Property Page**: Right-click the dark gray area outside the report canvas, and then select *Report Properties*.
- Using the **Properties** pane: Click the dark gray area outside the report canvas to select the report object. Ensure the **Properties** pane is open.

The **Page Setup** page of the report **Property Page** provides a friendly interface to view and update the page setup properties.



Ensure all page size properties are correctly configured:

Expand table

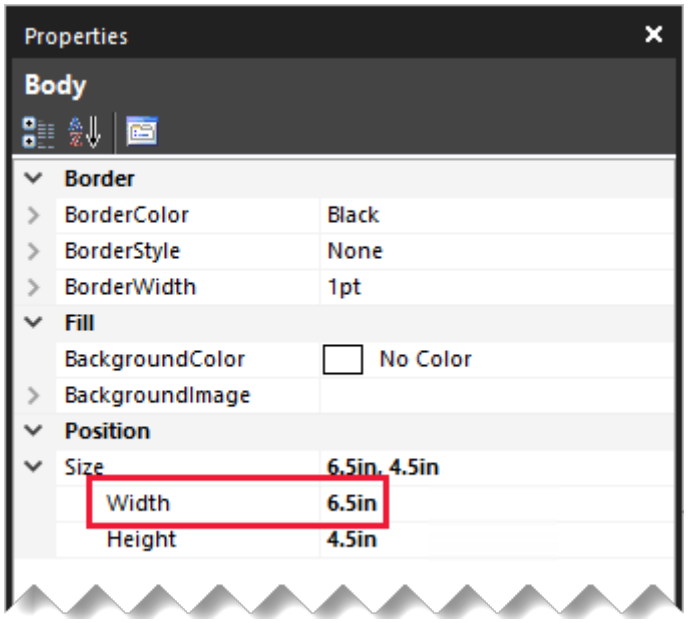
Property	Recommendation
Page units	Select the relevant units—inches or centimeters.
Orientation	Select the correct option—portrait or landscape.
Paper size	Select a paper size, or assign custom width and height values.
Margins	Set appropriate values for the left, right, top, and bottom margins.

Report body width

The page size properties determine the available space available for report objects. Report objects can be data regions, data visualizations, or other report items.

A common reason why blank pages are output, is the report body width *exceeds the available page space*.

You can only view and set the report body width using the **Properties** pane. First, click anywhere in an empty area of the report body.



Ensure the width value doesn't exceed available page width. Be guided by the following formula:

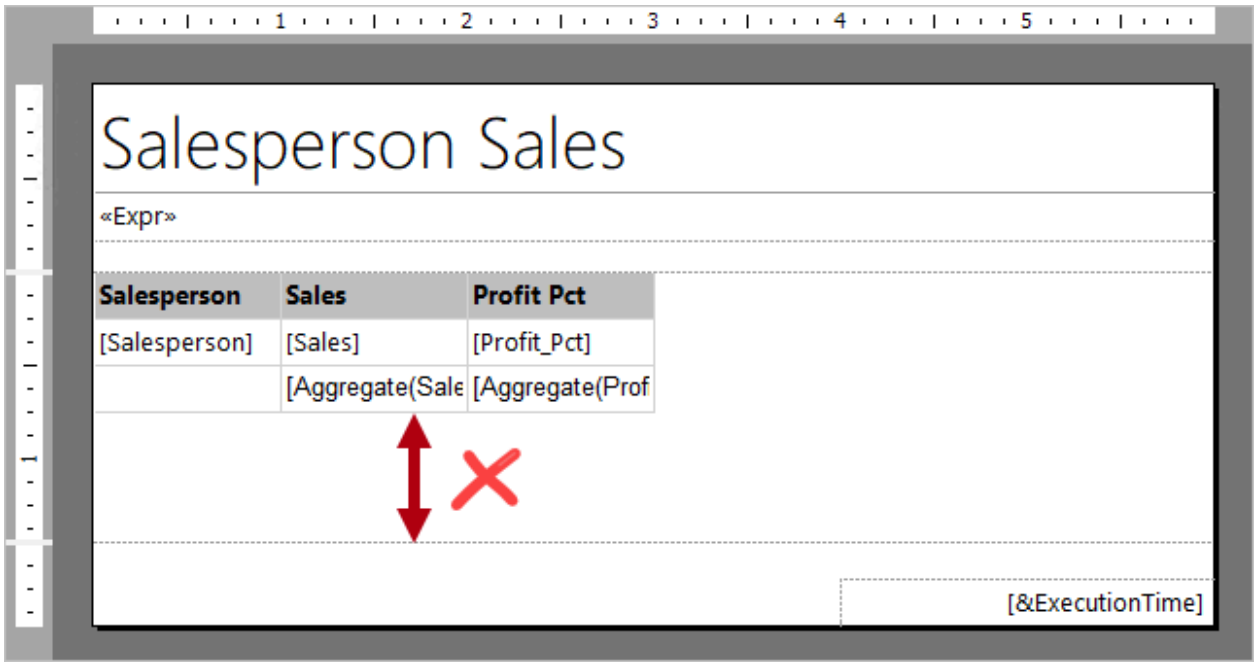
```
Report body width <= Report page width - (Left margin + Right margin)
```

> ⓘ Note
>
> It's not possible to reduce the report body width when there are report objects already in the space you want to remove. You must first reposition or resize them before reducing the width.
>
> Also, the report body width can increase automatically when you add new objects, or resize or reposition existing objects. The report designer always widens the body to accommodate the position and size of its contained objects.

Report body height

Another reason why a blank page is output, is there's excess space in the report body, after the last object.

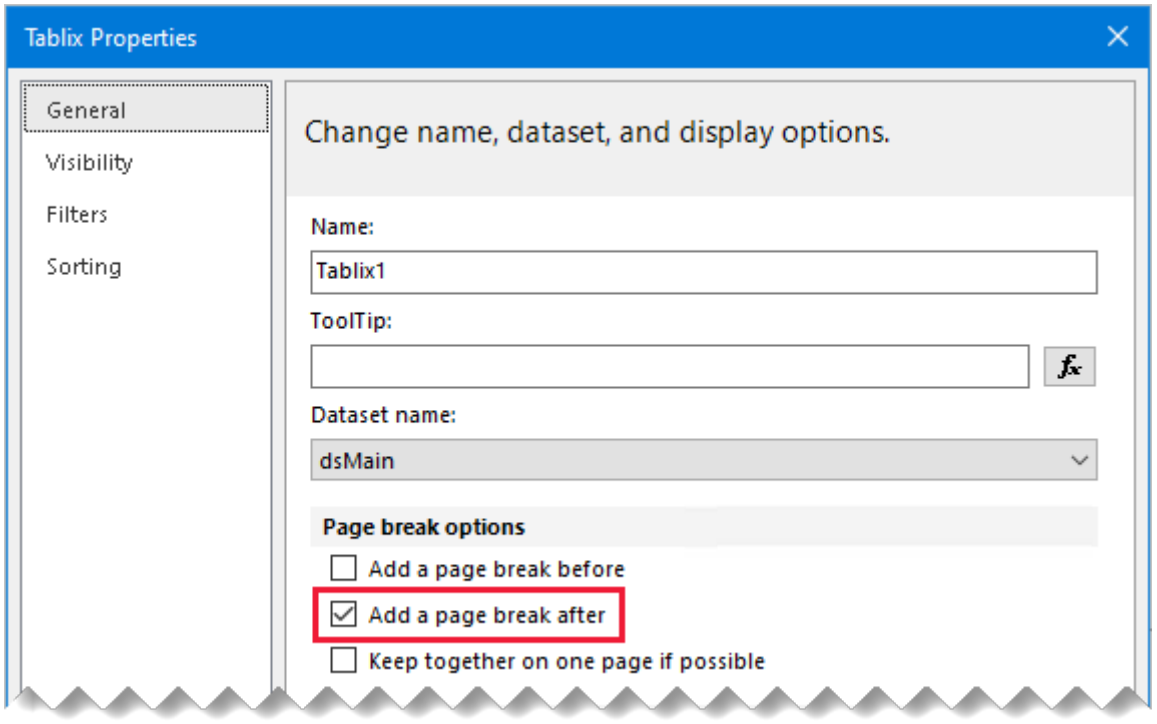
We recommend you always reduce the height of the body to remove any trailing space.



Page break options

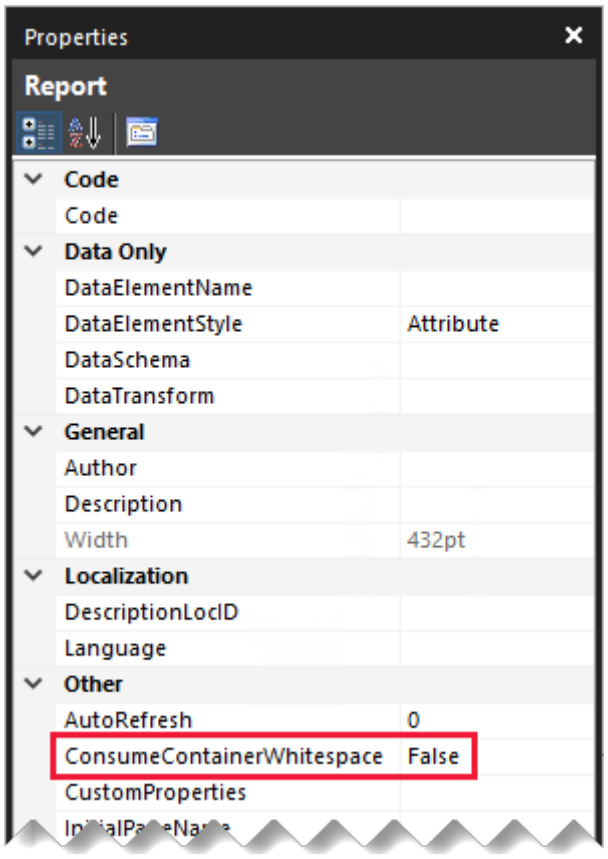
Each data region and data visualization has page break options. You can access these options in its property page, or in the **Properties** pane.

Ensure the **Add a page break after** property isn't unnecessarily enabled.



Consume Container Whitespace

If the blank page issue persists, you can also try disabling the report **ConsumeContainerWhitespace** property. It can only be set in the **Properties** pane.



By default, it's enabled. It directs whether minimum whitespace in containers, such as the report body or a rectangle, should be consumed. Only whitespace to the right of, and below, the contents is affected.

Printer paper size

Lastly, if you're printing the report to paper, ensure the printer has the correct paper loaded. The physical paper size should correspond to the report paper size.

Related content

For more information related to this article, check out the following resources:

- What are paginated reports in Power BI?
- Pagination in Power BI paginated reports
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Plan to migrate .rdl reports to Power BI

Article • 12/30/2024

APPLIES TO: ✔ Power BI Report Builder ⊗ Power BI Desktop ✔
Power BI 2022 Report Server ✔ SQL Server 2022 Reporting Services

This article targets Power BI Report Server and SQL Server Reporting Services (SSRS) report authors and Fabric administrators. It provides you with guidance to help you migrate your Report Definition Language (.rdl) reports to Power BI.

> ⓘ **Note**
>
> In Power BI, .rdl reports are called *paginated reports*.

Guidance is divided into four stages. We recommend that you first read the entire article prior to migrating your reports.

1. Before you start
2. Pre-migration stage
3. Migration stage
4. Post-migration stage

You can achieve migration without downtime to your report servers, or disruption to your report users. It's important to understand that you don't need to remove any data or reports. So, it means you can keep your current environment in place until you're ready for it to be retired.

Before you start

Before you start the migration, verify that your environment meets certain prerequisites. We describe these prerequisites, and also introduce you to a helpful migration tool.

Prepare for migration

As you prepare to migrate your reports to Power BI, first verify that you have a Power BI Pro or Premium Per User license to upload content to the target workspace.

Supported versions

You can migrate report server instances running on premises, or on virtual machines hosted by cloud providers like Azure.

The following list describes the SQL Server Reporting Services versions supported for migration to Power BI:

- ✓ SQL Server Reporting Services 2012
- ✓ SQL Server Reporting Services 2014
- ✓ SQL Server Reporting Services 2016
- ✓ SQL Server Reporting Services 2017
- ✓ SQL Server Reporting Services 2019
- ✓ SQL Server Reporting Services 2022

You can migrate .rdl files from Power BI Report Server as well.

Migration tool for Power BI Report Server and SQL Server Reporting Services 2017+

If you're using Power BI Report Server or SQL Server Reporting Services after SQL Server 2016, there's a built-in tool to publish its reports to Power BI. For more information, see Publish .rdl files to Power BI.

Migration tool for previous versions of SQL Server

For earlier versions of SQL Server Reporting Services, we recommend that you use the RDL Migration Tool to help prepare and migrate your reports. This tool was developed by Microsoft to help customers migrate .rdl reports from their SSRS servers to Power BI. It's available on GitHub, and it documents an end-to-end walkthrough of the migration scenario.

The tool automates the following tasks:

- Checks for unsupported data sources and unsupported report features.
- Converts any *shared* resources to *embedded* resources:
  - Shared **data sources** become embedded data sources.
  - Shared **datasets** become embedded datasets.
- Publishes reports that pass checks as *paginated reports*, to a specified Power BI workspace.

It doesn't modify or remove your existing reports. On completion, the tool outputs a summary of all actions completed, successful or unsuccessful.

Over time, Microsoft might improve the tool. The community is encouraged to contribute and help enhance it, too.

Pre-migration stage

After verifying that your organization meets the pre-requisites, you're ready to start the *Pre-migration* stage. This stage has three phases:

1. Discover
2. Assess
3. Prepare

Discover

The goal of the *Discover* phase is to identify your existing report server instances. This process involves scanning the network to identify all report server instances in your organization.

Start your migration by discovering and assessing on-premises resources using the free Azure Migrate tool. It can discover and report on your report server instances, versions, and installed features. It's a powerful inventory, assessment, and reporting tool that can help with your migration planning process.

Organizations might have hundreds of SQL Server Reporting Services (SSRS) reports. Some of those reports might become obsolete due to lack of use. The article Find and retire unused reports can help you discover unused reports and how to create a cadence for cleanup.

Assess

Having discovered your report server instances, the goal of the *Assess* phase is to understand any .rdl reports, or server items, that can't be migrated.

Your .rdl reports can be migrated from your report servers to Power BI. Each migrated .rdl report becomes a Power BI paginated report.

The following report server item types, however, can't be migrated to Power BI:

- **Shared data sources** and **shared datasets**: The RDL Migration Tool automatically converts shared data sources and shared datasets into embedded data sources and datasets, provided that they're using supported data sources.
- **Resources** such as image files.

- **Linked reports** migrate, whether the *parent* report that links to them is selected for migration or not. In the Power BI service, they're regular .rdl reports.
- **KPIs**: Power BI Report Server, or Reporting Services 2016 or later—Enterprise Edition only.
- **Mobile reports**: Power BI Report Server, or Reporting Services 2016 or later—Enterprise Edition only.
- **Report models**: deprecated.
- **Report parts**: deprecated.

If your .rdl reports rely on features not yet supported by Power BI paginated reports, you can plan to redevelop them as Power BI reports, when it makes sense.

For more information about supported data sources for paginated reports in the Power BI service, see Supported data sources for Power BI paginated reports.

Generally, Power BI paginated reports are optimized for **printing**, or **PDF generation**. Power BI reports are optimized for **exploration and interactivity**. For more information, see When to use paginated reports in Power BI.

Referencing custom code DLL files within a report isn't supported.

Differences in PDF output occur most often when a font that doesn't support non-Latin characters is used in a report and then non-Latin characters are added to the report. Test the PDF rendering output on both the report server and the client computers to verify that the report renders correctly.

Prepare

The goal of the *Prepare* phase involves getting everything ready. It covers setting up the Power BI environment, planning how you'll secure and publish your reports, and ideas for redeveloping report server items that won't migrate.

1. Verify support for your report data sources, and set up a Power BI gateway to allow connectivity with any on-premises data sources.
2. Become familiar with Power BI security, and plan how you'll reproduce your report server folders and permissions with Power BI workspaces.
3. Become familiar with Power BI sharing, and plan how you'll distribute content by publishing Power BI apps.
4. Consider using shared Power BI semantic models in place of your report server shared data sources.
5. Use Power BI Desktop to develop mobile-optimized reports, possibly using the Power KPI custom visual in place of your report server mobile reports and KPIs.

6. Reevaluate the use of the **UserID** built-in field in your reports. If you rely on the **UserID** to secure report data, then understand that for paginated reports (when hosted in the Power BI service) it returns the User Principal Name (UPN). So, instead of returning the NT account name, for example *AW\adelev*, the built-in field returns something like *adelev@adventureworks.com*. You need to revise your dataset definitions, and possibly the source data. Once revised and published, we recommend you thoroughly test your reports to ensure data permissions work as expected.
7. Reevaluate the use of the **ExecutionTime** built-in field in your reports. For paginated reports (when hosted in the Power BI service), the built-in field returns the date/time *in Coordinated Universal Time (or UTC)*. It could impact on report parameter default values, and report execution time labels (typically added to report footers).
8. If your data source is SQL Server (on premises), verify that reports aren't using map visualizations. The map visualization depends on SQL Server spatial data types, and these aren't supported by the gateway. For more information, see Data retrieval guidance for paginated reports (SQL Server complex data types).
9. For cascading parameters, be mindful that parameters are evaluated sequentially. Try preaggregating report data first. For more information, see Use cascading parameters in paginated reports.
10. Ensure your report authors have Power BI Report Builder installed, and that you can easily distribute later releases throughout your organization.
11. Utilize capacity planning documentation for paginated reports.

Migration stage

After preparing your Power BI environment and reports, you're ready for the *Migration* stage.

There are two migration options: *manual* and *automated*. Manual migration is suited to a small number of reports, or reports requiring modification before migration. Automated migration is suited to the migration of a large number of reports.

Manual migration

Anyone with permission to access to the report server instance and the Power BI workspace can manually migrate reports to Power BI. Here are the steps to follow:

1. Open the report server portal that contains the reports you want to migrate.
2. Download each report definition, saving the .rdl files locally.

3. Open *the latest version* of Power BI Report Builder, and connect to the Power BI service using your Microsoft Entra ID credentials.
4. Open each report in Power BI Report Builder, and then:
   a. Verify all data sources and datasets are embedded in the report definition, and that they're supported data sources.
   b. Preview the report to ensure it renders correctly.
   c. Select **Publish**, then select **Power BI service**.
   d. Select the workspace where you want to save the report.
   e. Verify that the report saves. If certain features in your report design aren't yet supported, the save action fails. You're notified of the reasons. You then need to revise your report design, and try saving again.

Automated migration

There are three options for automated migration. You can use:

- For Power BI Report Server and SQL Server 2022, see Publish .rdl files to Power BI.
- For previous versions of Reporting Services, use the RDL Migration Tool in GitHub.
- The publicly available APIs for Power BI Report Server, Reporting Services, and Power BI

You can also use the publicly available Power BI Report Server, Reporting Services, and Power BI APIs to automate the migration of your content. While the RDL Migration Tool already uses these APIs, you can develop a custom tool suited to your exact requirements.

For more information about the APIs, see:

- Power BI REST APIs.
- SQL Server Reporting Services REST APIs.

Post-migration stage

After you've successfully completed the migration, you're ready for the *Post-migration* stage. This stage involves working through a series of post-migration tasks to ensure everything is functioning correctly and efficiently.

Setting query time-out for embedded datasets

You specify query time-out values during report authoring when you define an embedded dataset. The time-out value is stored with the report, in the Timeout element of the report definition.

Configure data sources

Once reports have been migrated to Power BI, you need to ensure their data sources are correctly set up. It can involve assigning to gateway data sources, and securely storing data source credentials. These actions aren't done by the RDL Migration Tool.

Review report performance

We highly recommended you complete the following actions to ensure the best possible report user experience:

1. Test the reports in each browser supported by Power BI to confirm the report renders correctly.
2. Run tests to compare report rending times on the report server and in the Power BI service. Check that Power BI reports render in an acceptable time.
3. For long-rendering reports, consider having Power BI deliver them to your report users as email subscriptions with report attachments.
4. For Power BI reports based on Power BI semantic models, review model designs to ensure they're fully optimized.

Reconcile issues

The Post-migration phase is crucial for reconciling any issues, and that you address any performance concerns. Adding the paginated reports workload to a capacity can contribute to slow performance for paginated reports *and other content* stored in the capacity.

Related content

For more information about this article, check out the following resources:

- Publish .rdl files to Power BI from Power BI Report Server and SQL Server Reporting Services
- RDL Migration Tool for older versions of Reporting Services
- Power BI Report Builder
- Data retrieval guidance for paginated reports
- When to use paginated reports in Power BI
- Paginated reports in Power BI: FAQ
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Power BI partners are available to help your organization succeed with the migration process. To engage a partner, visit the Microsoft Power BI partner portal.

Feedback

Was this page helpful? Yes No

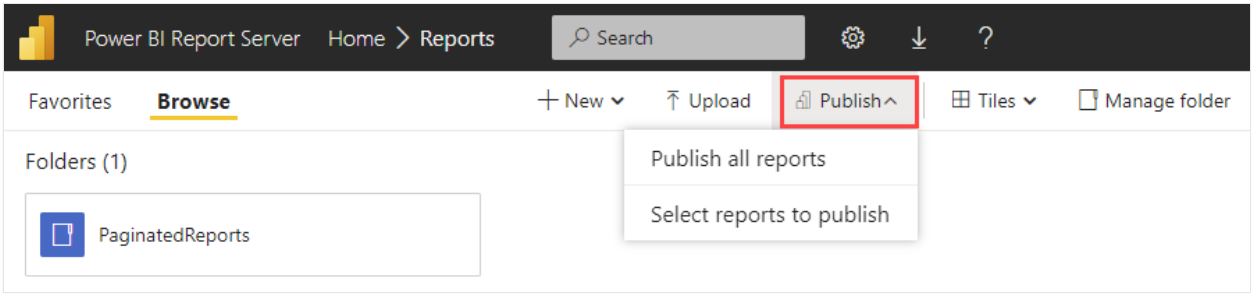
Provide product feedback | Ask the community

Publish .rdl files to Power BI from Power BI Report Server and Reporting Services

Article • 04/12/2023

APPLIES TO: ✔ Power BI Report Builder ✘ Power BI Desktop ✔ Power BI 2022 Report Server ✔ SQL Server 2022 Reporting Services

Do you have Report Definition Language (.rdl) paginated reports in Power BI Report Server or SQL Server 2022 Reporting Services (SSRS) that you want to migrate to the Power BI service? This article provides step-by-step instructions for migrating .rdl files *and Power BI reports (.pbix files)* from Power BI Report Server and SQL Server 2022 Reporting Services to the Power BI service.



> **Note**
>
> If you're using a previous version of Reporting Services, continue to use the **RDL Migration Tool** for now.

You can migrate reports without downtime to your report servers or disruption to your report users. It's important to understand that you don't need to remove any data or reports. You can keep your current environment in place until you're ready for it to be retired.

Prerequisites

My Workspace

You can publish and share paginated reports to your My Workspace with a Power BI free license.

Other workspaces

To publish to other workspaces, you need to meet these prerequisites:

- You have a Power BI Pro or Premium Per User license.
- You have write access to the workspace.

Read more about Power BI licenses.

Supported versions

You can migrate reports from SSRS instances running on-premises or on virtual machines hosted by cloud providers like Azure.

This publishing tool is designed to help customers migrate SSRS paginated reports (.rdl files) from their local servers to a Power BI workspace in their tenant. As part of the migration process, the tool also:

- Checks for unsupported data sources or report components when uploading to Power BI.
- Saves the converted files that pass these checks to a Power BI workspace that you specify.
- Provides a summary of the successful and unsuccessful assets migrated.

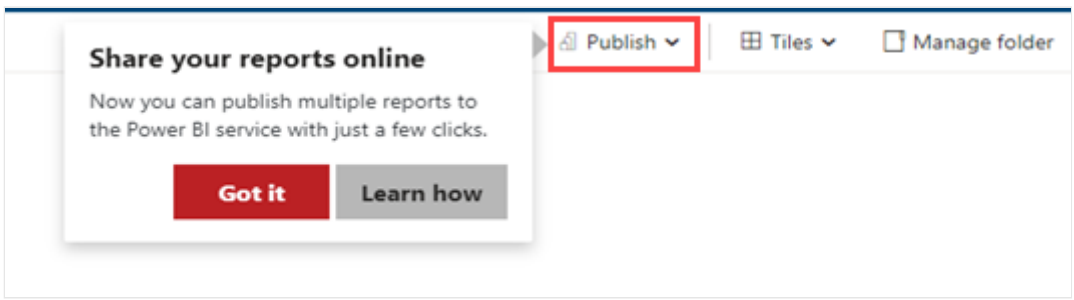
You can only migrate .rdl reports from your SSRS servers to Power BI. Each migrated .rdl report becomes a Power BI paginated report.

You can publish individual .rdl reports or the contents of entire folders from the SSRS web portal to the Power BI service. Read on to learn how to publish .rdl reports to Power BI.

Step 1: Browse to reports

SQL Server Reporting Services

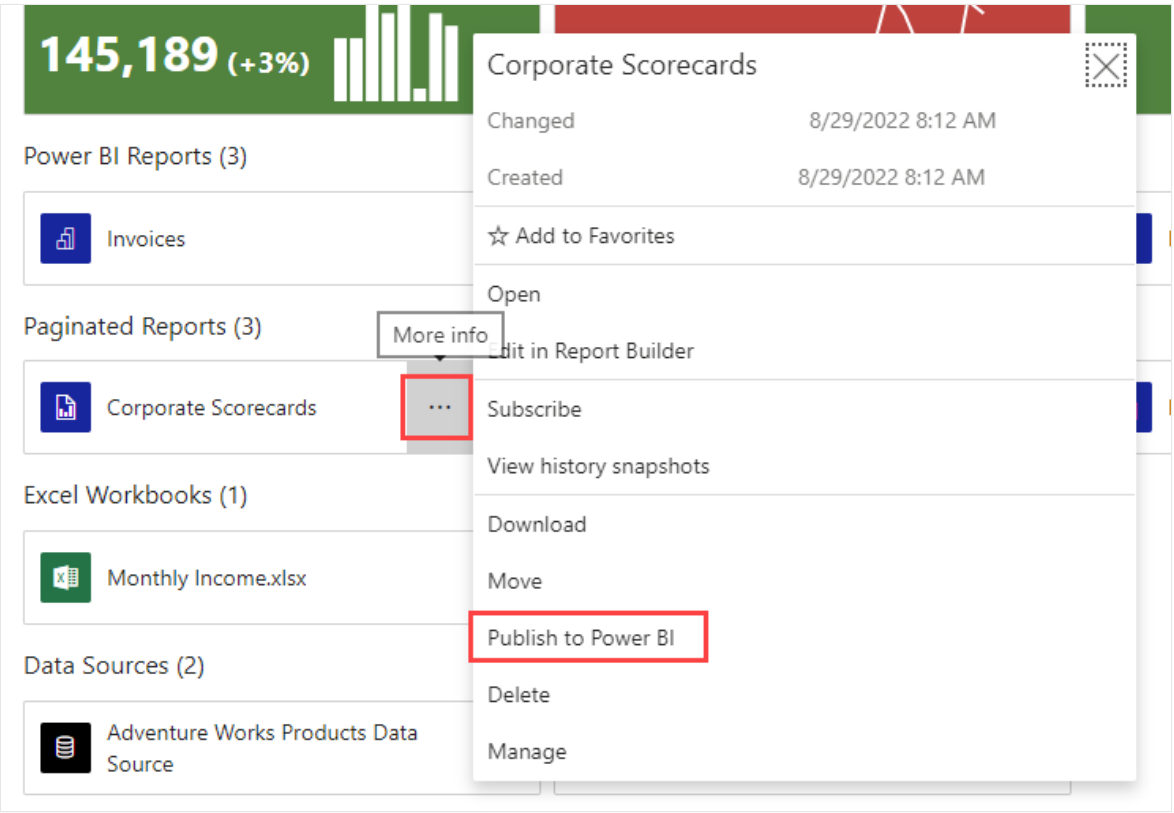
Select **Publish** to find the .rdl files you want to publish from SSRS to the Power BI service.



- Select **Publish all reports** to select all the .rdl files in the current folder and start the migration.
- Select **Select reports to publish** to open a list view of all .rdl files in the current folder. Select the reports and folders you want to migrate.

You can also publish individual articles.

- On the **More info** menu next to an .rdl report, select **Publish**.



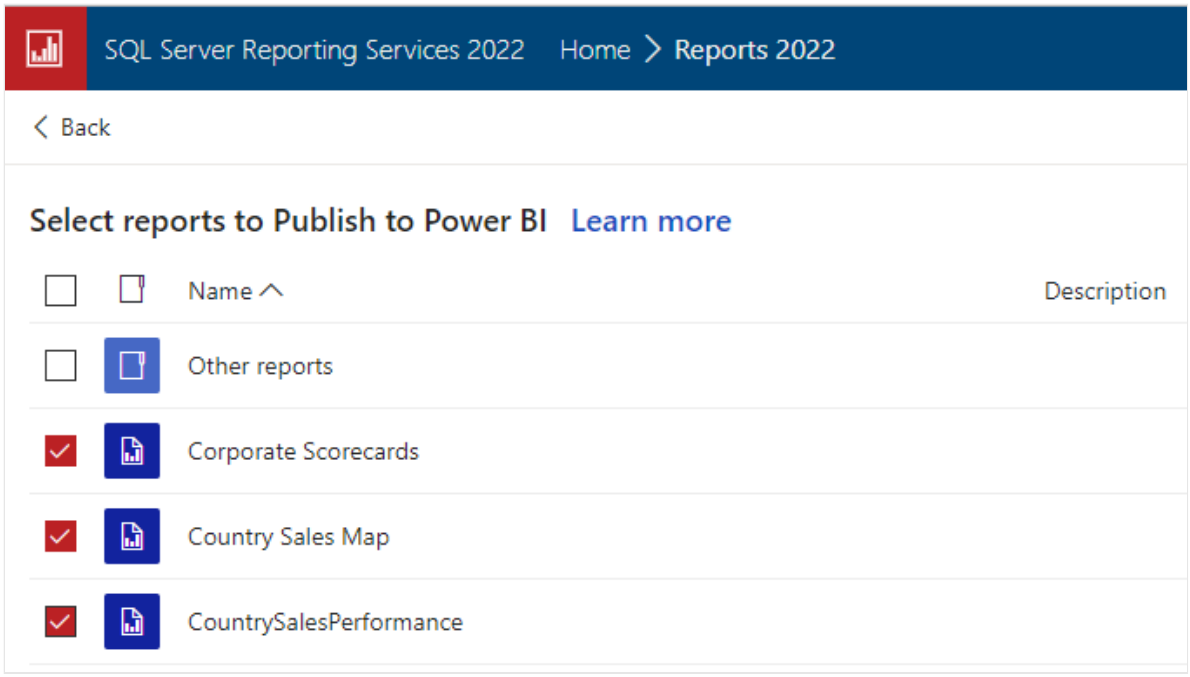
Step 1b: Select reports

Items that you can migrate now:

- The .rdl files
- Linked reports (.rdl files)
- Folders (all .rdl reports from the folder are migrated)

SQL Server Reporting Services

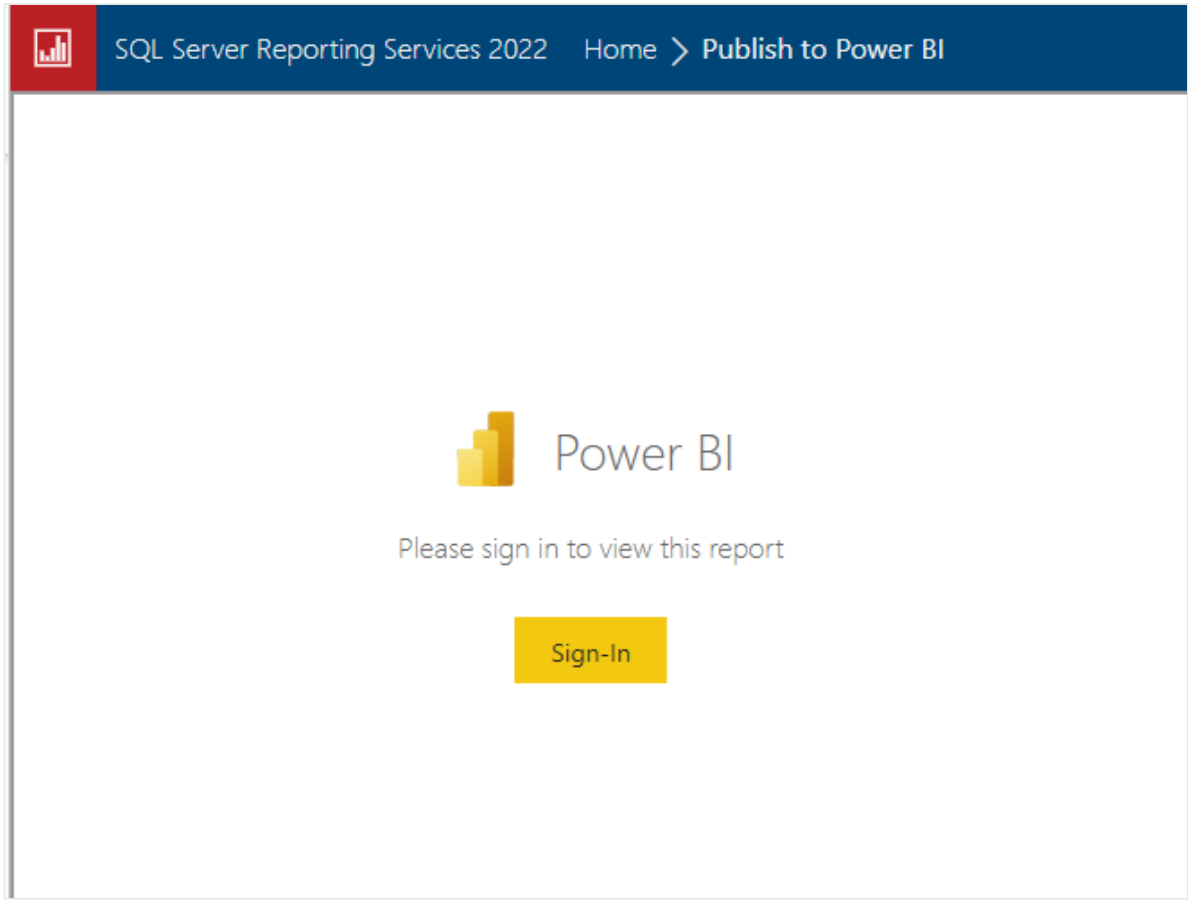
If you chose **Select reports to publish**, the next step is to **Select reports to publish to Power BI**.



Step 2: Sign in/Sign up

SQL Server Reporting Services

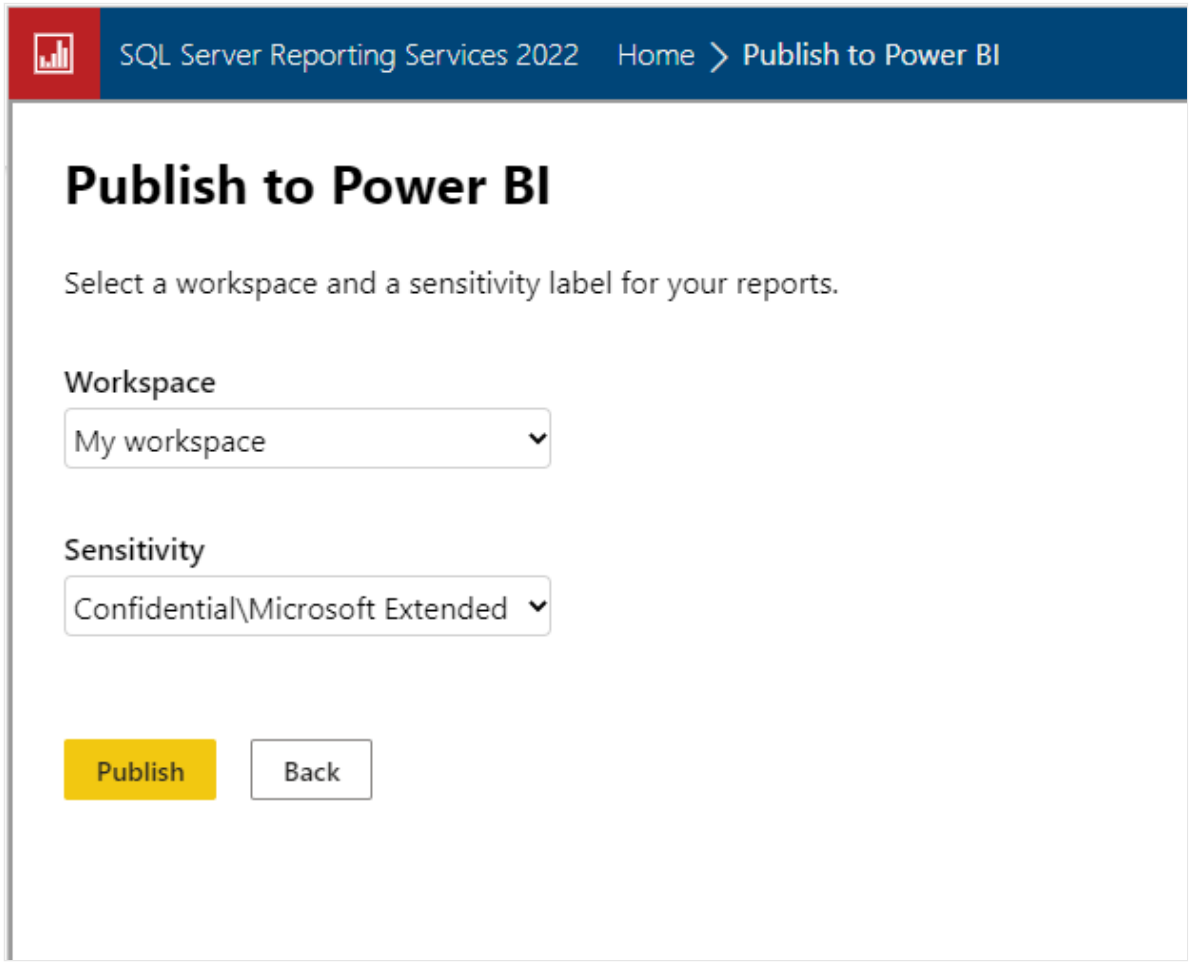
After you've selected the reports you want to publish, it's time to **Sign in** to the Power BI service.



Step 3: Choose a workspace

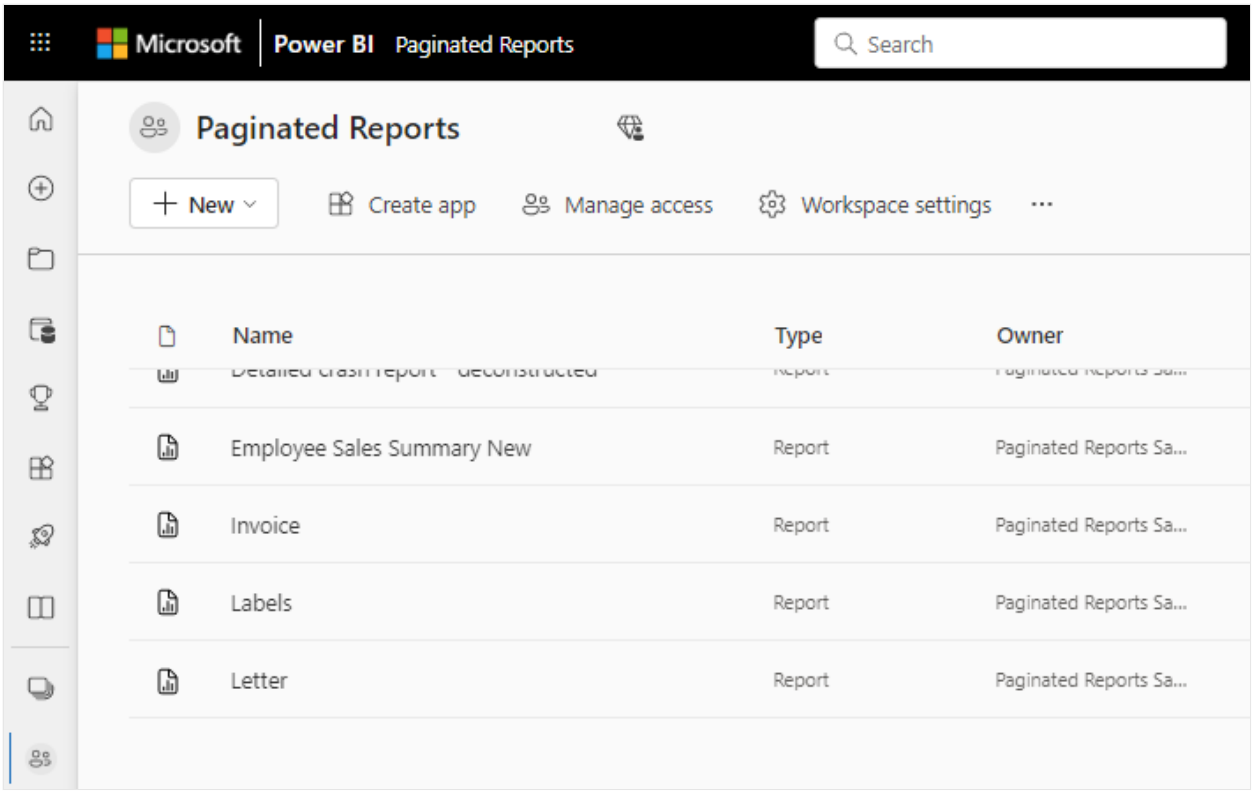
SQL Server Reporting Services

Now that you're signed in, select the dropdown arrow to find and **Select a workspace**.



Step 4: View reports

In the Power BI service, navigate to the workspace where you saved the reports.



Select a report to view it in the Power BI service.

Site properties

If you'd like to disable the migration setting, you need to update your report server. For more information on server properties, see the article Server Properties Advanced Page - Power BI Report Server & Reporting Services:

- EnablePowerBIReportMigrate
- PowerBIMigrateCountLimit
- PowerBIMigrateUrl

For sovereign clouds, you can update the Power BI endpoints by changing the site settings in the web portal.

Limitations and considerations

You can migrate .rdl reports from your report servers to the Power BI service. Each migrated .rdl report becomes a Power BI paginated report.

Converted report features

Shared data sources and datasets aren't yet supported in the Power BI service. When you migrate .rdl reports, the RDL Migration Tool automatically converts shared

datasets and data sources to embedded datasets and data sources, provided they're using supported datasets and data sources.

Unsupported item types

You can't migrate the following item types to the Power BI service:

- Resources such as image files
- KPIs
- Mobile reports (discontinued)
- Report models (discontinued)
- Report parts (discontinued)

Unsupported report features

See What paginated report features in SSRS aren't yet supported in Power BI? in the Paginated reports in Power BI FAQ for a complete list of unsupported report features.

Next steps

More questions? Try asking the Reporting Services forum

Find and retire unused .rdl reports

Article • 12/30/2024

APPLIES TO: ✅ Power BI Report Builder ⊗ Power BI Desktop ✅
Power BI 2022 Report Server ✅ SQL Server 2022 Reporting Services

Your company may deal with hundreds of paginated reports (.rdl files) in Power BI Report Server and SQL Server Reporting Services (SSRS). Some of those reports may become obsolete and need to be retired. As a report author or administrator, you don't want to migrate unused reports to the Power BI service. As you plan for a migration to the cloud, we suggest doing some housekeeping to get rid of unused .rdl reports. This best practice supports retention governance and allows your organization to make use of a retention schedule and data policy.

There are two processes for checking unused reports. We extend the cleanup to unused objects, as well as to get rid of unused database tables that could have potentially stale data.

Run an audit (optional)

First, we suggest that you create a server audit and database audit specification. Auditing an instance of the SQL Server Database Engine or an individual database involves tracking and logging events that occur on the Database Engine. SQL Server audit lets you create server audits, which can contain server audit specifications for server level events, and database audit specifications for database level events. Audited events can be written to the event logs or to audit files.

Once you've filled your audit log with tables and stored procedures used for reports, you can export those objects to an Excel file and share them with stakeholders. Let them know you're preparing to deprecate unused objects.

> **Note**
>
> Some important reports may run only rarely, so be sure to ask for feedback on database objects that are infrequently used. By deprecating an object, you can alter the object name by placing a *zdel* in front of it, so the object drops to the bottom of the Object Explorer. This way, if you decide later that you need the *zdel* object, you can alter the name back to the original. Once you know you're ready to remove them from your database, you can create a cadence to delete unused objects.

Create a Reports Usage metrics list

Second, you'll want to create an .rdl Reports Usage metrics list by querying Report Server DB. Use the T-SQL below to derive the usage counts. If your report server is configured to store one year of report execution history, you can use a specific date to filter the usage metrics.

Transact-SQL

```
; with UnusedReportsCte
AS
(
SELECT
Cat.Name,Path,COUNT(ExeLog.TimeStart) AS Cnt

FROM (SELECT * FROM Catalog
WHERE type=2 and Hidden=0) AS Cat
LEFT JOIN
ExecutionLog AS ExeLog
ON ExeLog.ReportID = Cat.ItemID
AND ExeLog.TimeStart>'01/01/2021'
GROUP BY Cat.Name,Path)
SELECT * FROM UnusedReportsCte
ORDER BY Cnt ASC,path
```

> **Note**
>
> Subreports and linked reports don't appear in the execution log if the parent report is executed.

From here you can decide whether to delete the unused reports right away or replace the report with a message. You can let your users know the report is no longer being used, so they can contact an administrator for support. Then you can develop a plan to delete them over time.

Related content

- Publish .rdl files to Power BI from Reporting Services
- Migrate SQL Server Reporting Services reports to Power BI

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Develop scalable multitenancy applications with Power BI embedding

Article • 12/30/2024

This article describes how to develop a multitenancy application that embeds Power BI content while achieving the highest levels of scalability, performance, and security. By designing and implementing an application with *service principal profiles*, you can create and manage a multitenancy solution comprising tens of thousands of customer tenants that can deliver reports to audiences of over 100,000 users.

Service principal profiles is a feature that makes it easier for you to manage organizational content in Power BI and use your capacities more efficiently. However, using service principal profiles can add complexity to your application design. Therefore, you should only use them when there's a need to achieve significant scale. We recommend using service principal profiles when you have many workspaces and more than 1,000 application users.

> **Note**
>
> The value of using service principal profiles increases as your need to scale increases as well as your need to achieve the highest levels of security and tenant isolation.

You can achieve Power BI embedding by using two different embedding scenarios: *Embed for your organization* and *Embed for your customer*.

The **Embed for your organization** scenario applies when the application audience comprises *internal* users. Internal users have organizational accounts and must authenticate with Microsoft Entra ID. In this scenario, Power BI is software-as-a-service (SaaS). It's sometimes referred to as *User owns data*.

The **Embed for your customer** scenario applies when the application audience comprises *external* users. The application is responsible for authenticating users. To access Power BI content, the application relies on an embedding identity (Microsoft Entra service principal or master user account) to authenticate with Microsoft Entra. In this scenario, Power BI is platform-as-a-service (PaaS). It's sometimes referred to as *App owns data*.

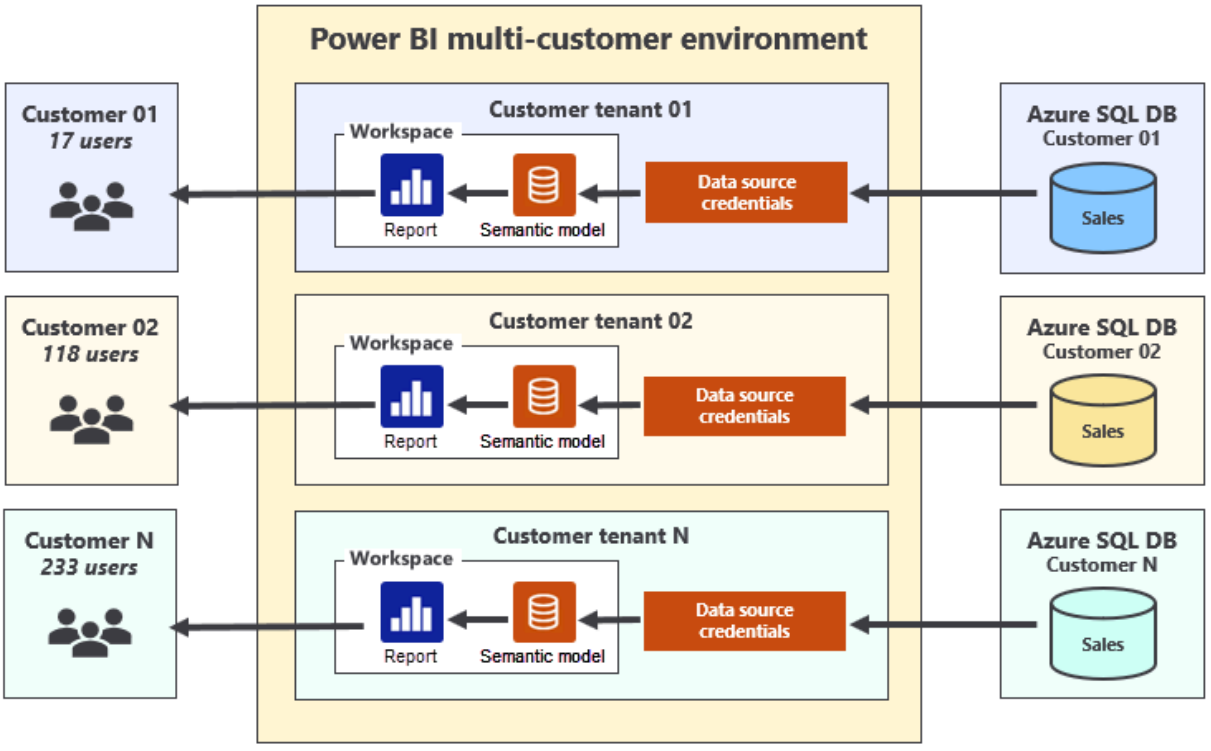
> **Note**

It's important to understand that the service principal profiles feature was designed for use with the *Embed for your customer* scenario. That's because this scenario offers ISVs and enterprise organizations the ability to embed with greater scale to a large number of users and to a large number of customer tenants.

Multitenancy application development

If you're familiar with Microsoft Entra, the word *tenant* might lead you think of a Microsoft Entra tenant. However, the concept of a tenant is different in the context of building a multitenancy solution that embeds Power BI content. In this context, a *customer tenant* is created on behalf of each customer for which the application embeds Power BI content by using the *Embed for your customer* scenario. You typically provision each customer tenant by creating a single Power BI workspace.

To create a scalable multitenancy solution, you must be able to automate the creation of new customer tenants. Provisioning a new customer tenant typically involves writing code that uses the Power BI REST API to create a new Power BI workspace, create semantic models by importing Power BI Desktop (.pbix) files, update data source parameters, set data source credentials, and set up scheduled semantic model refresh. The following diagram shows how you can add Power BI items, such as reports and semantic models, to workspaces to set up customer tenants.



When you develop an application that uses the *Embed for your customer* scenario, it's possible to make Power BI REST API calls by using an embedding identity that's either a

master user account or a service principal. We recommend using a service principal for production applications. It provides the highest security and for this reason it's the approach recommended by Microsoft Entra. Also, it supports better automation and scale and there's less management overhead. However, it requires Fabric admin rights to set up and manage.

By using a service principal, you can avoid common problems associated with master user accounts, such as authentication errors in environments where users are required to sign in by using multifactor authentication (MFA). Using a service principal is also consistent with the idea that the *Embed for your customer* scenario is based on embedding Power BI content by using a PaaS mindset as opposed to a SaaS mindset.

1,000-workspace limitation

When you design a multitenancy environment that implements the *Embed for your customer* scenario, be sure to consider that the embedding identity can't be granted access to more than 1,000 workspaces. The Power BI service imposes this limitation to ensure good performance when making REST API calls. The reason for this limitation is related to how Power BI maintains security-related metadata for each identity.

Power BI uses metadata to track the workspaces and workspace items an identity can access. In effect, Power BI must maintain a separate access control list (ACL) for each identity in its authorization subsystem. When an identity makes a REST API call to access a workspace, Power BI must do a security check against the identity's ACL to ensure it's authorized. The time it takes to determine whether the target workspace is inside the ACL increases exponentially as the number of workspaces increases.

> **Note**
>
> Power BI doesn't enforce the 1,000 workspace limitation through code. If you try, you add an embedding identity to more than 1,000 workspaces, and REST API calls will still execute successfully. However, your application will move into an *unsupported* state, which may have implications should you try to request help from Microsoft support.

Consider a scenario where two multi-tenant applications have each been set up to use a single service principal. Now consider that the first application has created 990 workspaces while the second application has created 1,010 workspaces. From a support perspective, the first application is within the supported boundaries while the second application isn't.

Now compare these two applications purely from a performance perspective. There's not that much difference because the ACLs for both service principals have let the metadata for their ACLs grow to a point where it will degrade performance to some degree.

Here's the key observation: The number of workspaces created by a service principal has a direct impact on performance and scalability. A service principal that's a member of 100 workspaces will execute REST API calls faster than a service principal that's a member of 1,000 workspaces. Likewise, a service principal that's a member of only 10 workspaces will execute REST API calls faster than a service principal that's a member of 100 workspaces.

> **Important**
>
> From the perspective of performance and scalability, the optimal number of workspaces for which a service principal is a member is *exactly one*.

Manage isolation for semantic models and data source credentials

Another important aspect when designing a multitenancy application is to isolate customer tenants. It's critical that users from one customer tenant don't see data that belongs to another customer tenant. Therefore, you must understand how to manage semantic model ownership and data source credentials.

Semantic model ownership

Each Power BI semantic model has a single owner, which can be either a user account or a service principal. Semantic model ownership is required to set up scheduled refresh and set semantic model parameters.

> **Tip**
>
> In the Power BI service, you can determine who the semantic model owner is by opening the semantic model settings.

If necessary, you can transfer ownership of the semantic model to another user account or service principal. You can do that in the Power BI service, or by using the REST API TakeOver operation. When you import a Power BI Desktop file to create a new semantic

model by using a service principal, the service principal is automatically set as the semantic model owner.

Data source credentials

To connect a semantic model to its underlying data source, the semantic model owner must set data source credentials. Data source credentials are encrypted and cached by Power BI. From that point, Power BI uses those credentials to authenticate with the underlying data source when refreshing the data (for import storage tables) or executing passthrough queries (for DirectQuery storage tables).

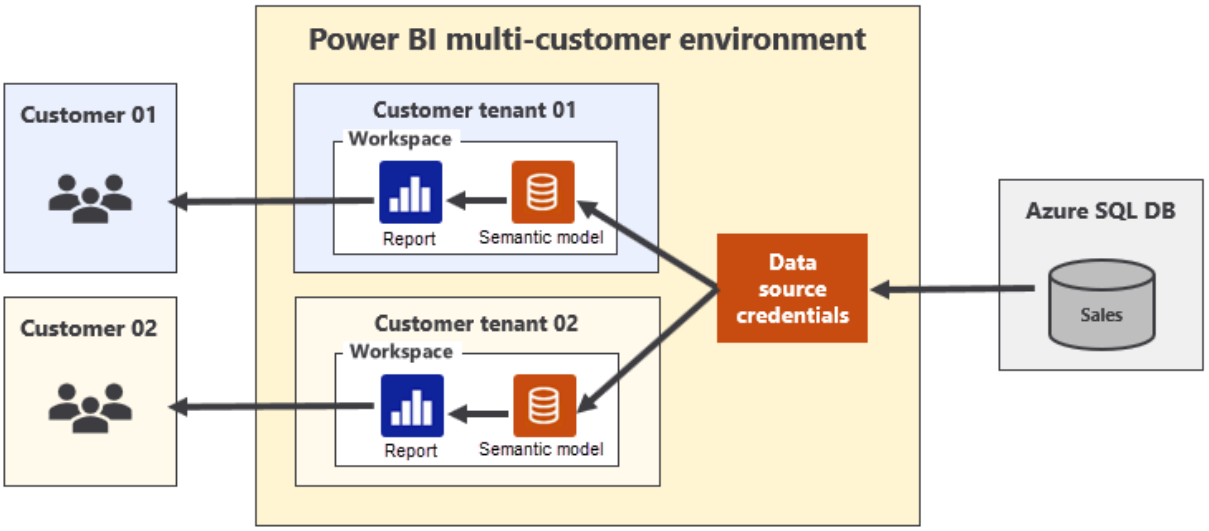
We recommend that you apply a common design pattern when provisioning a new customer tenant. You can execute a series of REST API calls by using the identity of the service principal:

1. Create a new workspace.
2. Associate the new workspace with a dedicated capacity.
3. Import a Power BI Desktop file to create a semantic model.
4. Set the semantic model source credentials for that semantic model.

On completion of these REST API calls, the service principal will be an admin of the new workspace and the owner of the semantic model and data source credentials.

> **ⓘ Important**
>
> There's a common misconception that semantic model data source credentials are workspace-level scoped. That's not true. Data source credentials are scoped to the service principal (or user account) and that scope extends to all Power BI workspaces in the Microsoft Entra tenant.

It's possible for a service principal to create data source credentials that are shared by semantic models in different workspaces across customer tenants, as shown in the following diagram.



When data source credentials are shared by semantic models that belong to different customer tenants, the customer tenants aren't fully isolated.

Design strategies prior to service principal profiles

Understanding design strategies before the service principal profile feature became available can help you to appreciate the need for the feature. Before that time, developers built multitenancy applications by using one of the following three design strategies:

- Single service principal
- Service principal pooling
- One service principal per workspace

There are strengths and weakness associated with each of these design strategies.

The **single service principal** design strategy requires a once-off creation of a Microsoft Entra app registration. Therefore, it involves less administrative overhead than the other two design strategies because there's no requirement to create more Microsoft Entra app registrations. This strategy is also the most straightforward to set up because it doesn't require writing extra code that switches the calling context between service principals when making REST API calls. However, a problem with this design strategy is that it doesn't scale. It only supports a multitenancy environment that can grow up to 1,000 workspaces. Also, performance is sure to degrade as the service principal is granted access to a larger number of workspaces. There's also a problem with customer tenant isolation because the single service principal becomes the owner of every semantic model and all data credentials across all customer tenants.

The **service principal pooling** design strategy is commonly used to avoid the 1,000-workspace limitation. It allows the application to scale to any number of workspaces by

adding the correct number of service principals to the pool. For example, a pool of five service principals makes it possible to scale up to 5,000 workspaces; a pool of 80 service principals makes it possible to scale up to 80,000 workspaces, and so on. However, while this strategy can scale to a large number of workspaces, it has several disadvantages. First, it requires writing extra code and storing metadata to allow context switching between service principals when making REST API calls. Second, it involves more administrative effort because you must create Microsoft Entra app registrations whenever you need to increase the number of the service principals in the pool.

What's more, the service principal pooling strategy isn't optimized for performance because it allows service principals to become members of hundreds of workspaces. It also isn't ideal from the perspective of customer tenant isolation because the service principals can become owners of semantic model and data credentials shared across customer tenants.

The **one service principal per workspace** design strategy involves creating a service principal for each customer tenant. From a theoretical perspective, this strategy offers the best solution because it optimizes the performance of REST API calls while providing true isolation for semantic models and data source credentials at the workspace level. However, what works best in theory doesn't always work best in practice. That's because the requirement to create a service principal for each customer tenant is impractical for many organizations. That's because some organizations have formal approval processes or they involve excessive bureaucracy to create Microsoft Entra app registrations. These reasons can make it impossible to grant a custom application the authority it needs to create Microsoft Entra app registrations on-demand and in the automated way that your solution requires.

In less common scenarios where a custom application has been granted proper permissions, it can use the Microsoft Graph API to create Microsoft Entra app registrations on demand. However, the custom application is often complex to develop and deploy because it must somehow track authentication credentials for each Microsoft Entra app registration. It must also gain access to those credentials whenever it needs to authenticate and acquire access tokens for individual service principals.

Service principal profiles

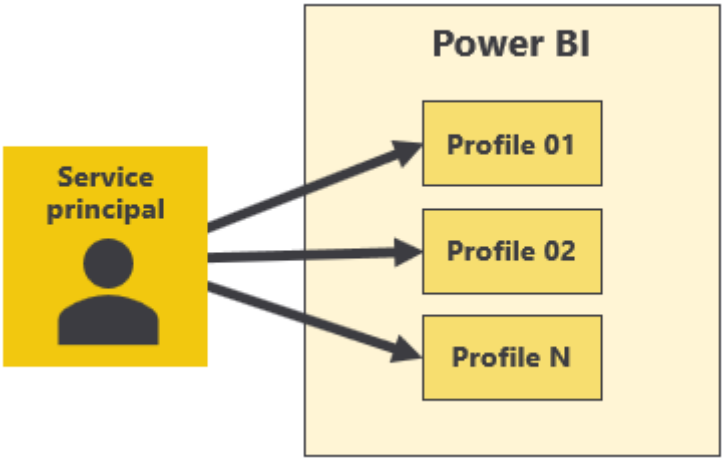
The service principal profiles feature was designed to make it easier for you to manage organizational content in Power BI and use your capacities more efficiently. They help address three specific challenges that involve the lowest amount of developer effort and overhead. These challenges include:

- Scaling to a large number of workspaces.

- Optimizing performance of REST API calls.
- Isolating semantic models and data source credentials at the customer tenant level.

When you design a multitenancy application by using service principal profiles, you can benefit from the strengths of the three design strategies (described in the previous section) while avoiding their associated weaknesses.

Service principal profiles are local accounts that are created within the context of Power BI. A service principal can use the Profiles REST API operation to create new service principal profiles. A service principal can create and manage its own set of service principal profiles for a custom application, as shown in the following diagram.



There's always a parent-child relationship between a service principal and the service principal profiles it creates. You can't create a service principal profile as a stand-alone entity. Instead, you create a service principal profile by using a specific service principal, and that service principal serves as the profile's parent. Furthermore, a service principal profile is never visible to user accounts or other service principals. A service principal profile can only be seen and used by the service principal that created it.

Service principal profiles aren't known to Microsoft Entra

While the service principal itself and its underlying Microsoft Entra app registration are known to Microsoft Entra, Microsoft Entra ID doesn't know anything about service principal profiles. That's because service principal profiles are created by Power BI and they exist only in the Power BI service subsystem that controls Power BI security and authorization.

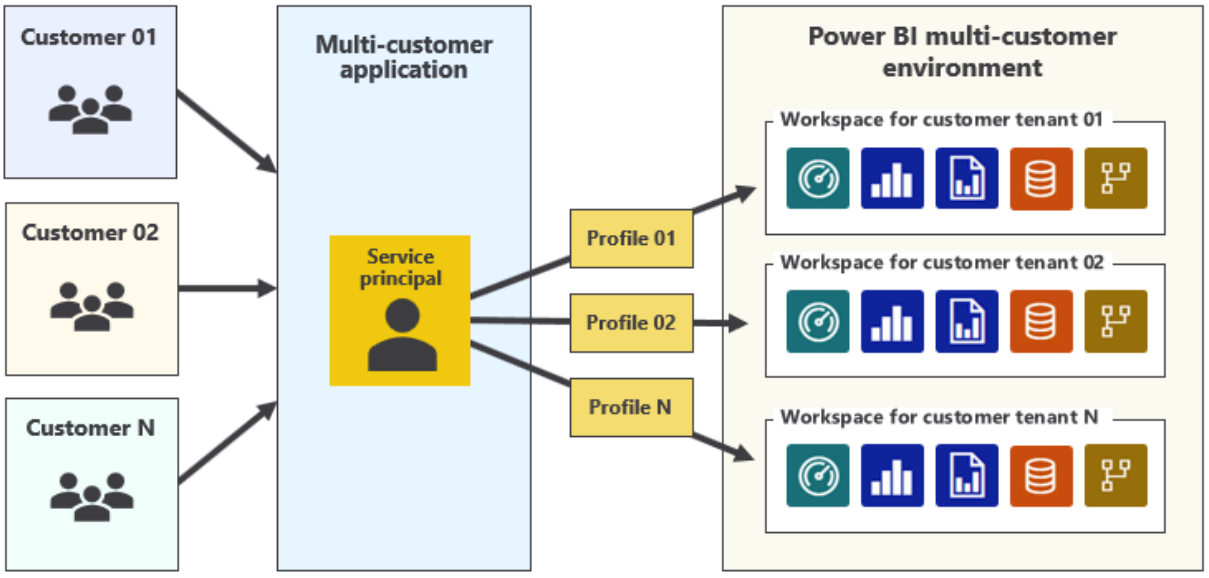
The fact that service principal profiles aren't known to Microsoft Entra ID has both advantages and disadvantages. The primary advantage is that an *Embed for your customer* scenario application doesn't need any special Microsoft Entra permissions to

create service principal profiles. It also means that the application can create and manage a set of local identities that are separate from Microsoft Entra.

However, there are also disadvantages. Because service principal profiles aren't known to Microsoft Entra, you can't add a service principal profile to a Microsoft Entra group to implicitly grant it access to a workspace. Also, external data sources, such as an Azure SQL Database or Azure Synapse Analytics, can't recognize service principal profiles as the identity when connecting to a database. So, the **one service principal per workspace** design strategy (creating a service principal for each customer tenant) might be a better choice when there's a requirement to connect to these data sources by using a separate service principal with unique authentication credentials for each customer tenant.

Service principal profiles are first-class security principals

While service principal profiles aren't known to Microsoft Entra, Power BI recognizes them as first-class security principals. Just like a user account or a service principal, you can add service principal profiles to a workspace role (as an Admin or Member). You can also make it a semantic model owner and the owner of data source credentials. For these reasons, creating a new service principal profile for each new customer tenant is a best practice.



> **Tip**
>
> When you develop an *Embed for your customer* scenario application by using service principal profiles, you only need to create a single Microsoft Entra app registration to provide your application with a single service principal. This approach significantly lowers administrative overhead compared to other

multitenancy design strategies, where it's necessary to create additional Microsoft Entra app registrations on an ongoing basis after the application is deployed to production.

Execute REST API calls as a service principal profile

Your application can execute REST API calls by using the identity of a service principal profile. That means it can execute a sequence of REST API calls to provision and set up a new customer tenant.

1. When a service principal profile creates a new workspace, Power BI automatically adds that profile as a workspace admin.
2. When a service principal profile imports a Power BI Desktop file to create a semantic model, Power BI sets that profile as the semantic model owner.
3. When a service principal profile sets data source credentials, Power BI sets that profile as the owner of the data source credentials.

It's important to understand that a service principal has an identity in Power BI that's separate and distinct from the identities of its profiles. That provides you with choice as a developer. You can execute REST API calls by using the identity of a service principal profile. Alternatively, you can execute REST API calls without a profile, which uses the identity of the parent service principal.

We recommend that you execute REST API calls as the parent service principal when you're creating, viewing, or deleting service principal profiles. You should use the service principal profile to execute all other REST API calls. These other calls can create workspaces, import Power BI Desktop files, update semantic model parameters, and set data source credentials. They can also retrieve workspace item metadata and generate embed tokens.

Consider an example where you need to set up a customer tenant for a customer named *Contoso*. The first step makes a REST API call to create a service principal profile with its display name set to *Contoso*. This call is made by using the identity of the service principal. All remaining set up steps use the service principal profile to complete the following tasks:

1. Create a workspace.
2. Associate the workspace with a capacity.
3. Import a Power BI Desktop file.
4. Set semantic model parameters.
5. Set data source credentials.
6. Set up scheduled data refresh.

It's important to understand that access to the workspace and its content must be done by using the identity of the service principal profile that was used to create the customer tenant. It's also important to understand that the parent service principal doesn't need access to the workspace or its content.

> **Tip**
>
> Remember: When making REST API calls, use the service principal to create and manage service principal profiles, and use the service principal profile to create, set up, and access Power BI content.

Use the Profiles REST API operations

The **Profiles** REST API operation group comprises operations that create and manage service principal profiles:

- `Create Profile`
- `Delete Profile`
- `Get Profile`
- `Get Profiles`
- `Update Profile`

Create a service principal profile

Use the Create Profile REST API operation to create a service principal profile. You must set the `displayName` property in the request body to provide a display name for the new tenant. The value must be unique across all the profiles owned by the service principal. The call will fail if another profile with that display name already exists for the service principal.

A successful call returns the `id` property, which is a GUID that represents the profile. When you develop applications that use service principal profiles, we recommend that you store profile display names and their ID values in a custom database. That way, it's straightforward for your application to retrieve the IDs.

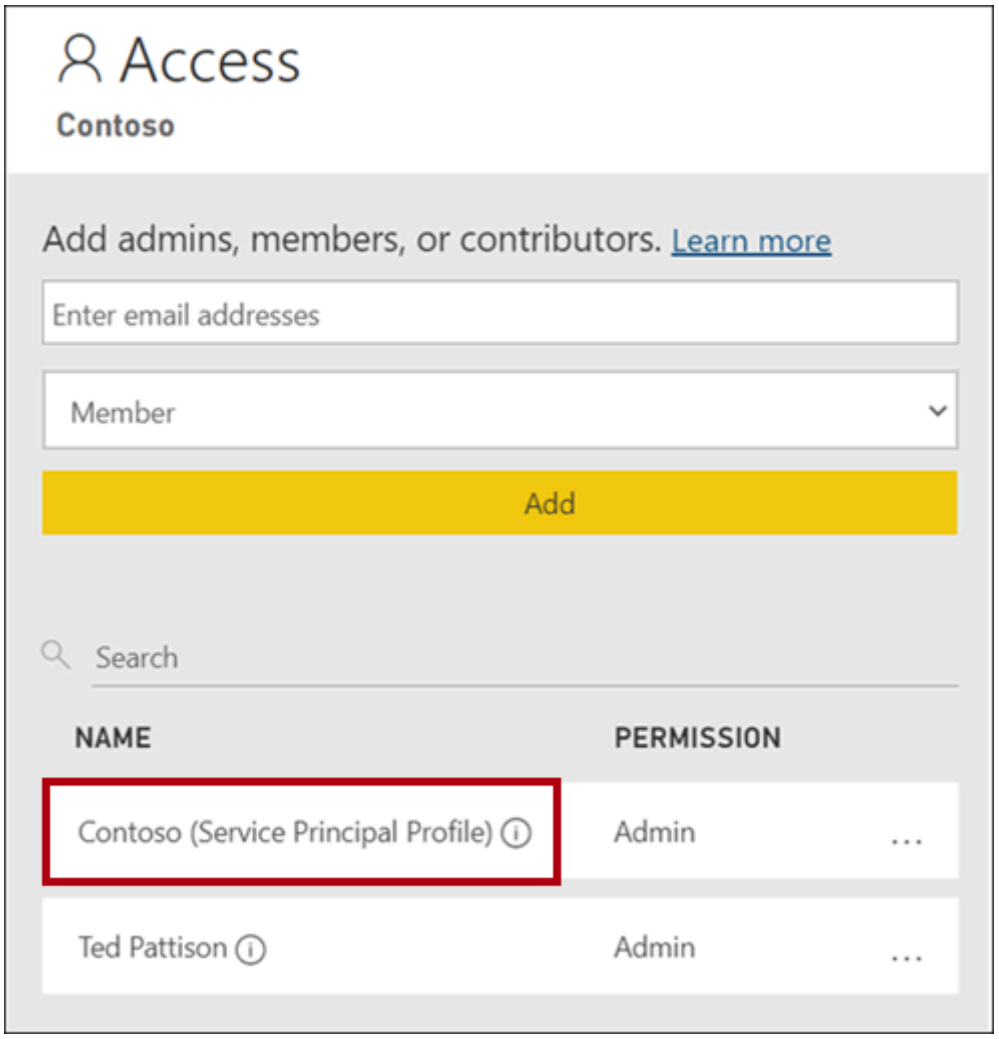
If you're programming with the Power BI .NET SDK, you can use the `Profiles.CreateProfile` method, which returns a `ServicePrincipalProfile` object representing the new profile. It makes it straightforward to determine the `id` property value.

Here's an example of creating a service principal profile and granting it workspace access.

C#

```
// Create a service principal profile
string profileName = "Contoso";

var createRequest = new CreateOrUpdateProfileRequest(profileName);
var profile = pbiClient.Profiles.CreateProfile(createRequest);

// Retrieve the ID of the new profile
Guid profileId = profile.Id;

// Grant workspace access
var groupUser = new GroupUser {
    GroupUserAccessRight = "Admin",
    PrincipalType = "App",
    Identifier = ServicePrincipalId,
    Profile = new ServicePrincipalProfile {
        Id = profileId
    }
};

pbiClient.Groups.AddGroupUser(workspaceId, groupUser);
```

In the Power BI service, in the workspace **Access** pane, you can determine which identities, including security principals, have access.



Delete a service principal profile

Use the Delete Profile REST API operation to delete a service principal profile. This operation can only be called by the parent service principal.

If you're programming with the Power BI .NET SDK, you can use the `Profiles.DeleteProfile` method.

Retrieve all service principal profiles

Use the Get Profiles REST API operation to retrieve a list of service principal profiles that belong to the calling service principal. This operation returns a JSON payload that contains the `id` and `displayName` properties of each service principal profile.

If you're programming with the Power BI .NET SDK, you can use the **Profiles.GetProfiles** method.

Execute REST API calls by using a service principal profile

There are two requirements for making REST API calls by using a service principal profile:

- You must pass the access token for the parent service principal in the **Authorization** header.
- You must include a header named **X-PowerBI-profile-id** with the value of the service principal profile's ID.

If you're using the Power BI .NET SDK, you can set the **X-PowerBI-profile-id** header value explicitly by passing in the service principal profile's ID.

C#

```
// Create the Power BI client
var tokenCredentials = new TokenCredentials(GetACcessToken(). "Bearer");
var uriPowerBiServiceApiRoot = new Uri(uriPowerBiServiceApiRoot);
var pbiClient = new PowerBIClient(uriPowerBiServiceApiRoot,
tokenCredentials);

// Add X-PowerBI-profile-id header for service principal profile
string profileId = "11111111-1111-1111-1111-111111111111";
pbiClient.HttpClient.DefaultRequestHeaders.Add("X-PowerBI-profile-id",
profileId);

// Retrieve workspaces by using the identity of service principal profile
var workspaces = pbiClient.Groups.GetGroups();
```

As shown in the above example, once you add the **X-PowerBI-profile-id** header to the `PowerBIClient` object, it's straightforward to invoke methods, such as `Groups.GetGroups`, so they'll be executed by using the service principal profile.

There's a more convenient way to set the **X-PowerBI-profile-id** header for a `PowerBIClient` object. You can initialize the object by passing in the profile's ID to the constructor.

C#

```
// Create the Power BI client
string profileId = "11111111-1111-1111-1111-111111111111";

var tokenCredentials = new TokenCredentials(GetACcessToken(). "Bearer");
var uriPowerBiServiceApiRoot = new Uri(uriPowerBiServiceApiRoot);
var pbiClient = new PowerBiClient(uriPowerBiServiceApiRoot,
tokenCredentials, profileId);
```

As you program a multitenancy application, it's likely that you'll need to switch between executing calls as the parent service principal and executing calls as a service principal

profile. A useful approach to manage context switching is to declare a class-level variable that stores the `PowerBIClient` object. You can then create a helper method that sets the variable with the correct object.

C#

```
// Class-level variable that stores the PowerBIClient object
private PowerBIClient pbiClient;

// Helper method that sets the correct PowerBIClient object
private void SetCallingContext(string profileId = "") {

    if (profileId.Equals("")) {
        pbiClient = GetPowerBIClient();
    }
    else {
        pbiClient = GetPowerBIClientForProfile(new Guid(profileId));
    }
}
```

When you need to create or manage a service principal profile, you can call the `SetCallingContext` method without any parameters. This way, you can create and manage profiles by using the identity of the service principal.

C#

```
// Always create and manage profiles as the service principal
SetCallingContext();

// Create a service principal profile
string profileName = "Contoso";

var createRequest = new CreateOrUpdateProfileRequest(profileName);
var profile = pbiClient.Profiles.CreateProfile(createRequest);
```

When you need to create and set up a workspace for a new customer tenant, you want to execute that code as a service principal profile. Therefore, you should call the `SetCallingContext` method by passing in the profile's ID. This way, you can create the workspace by using the identity of the service principal profile.

C#

```
// Always create and set up workspaces as a service principal profile
string profileId = "11111111-1111-1111-1111-111111111111";
SetCallingContext(profileId);

// Create a workspace
```

```
GroupCreationRequest request = new GroupCreationRequest(workspaceName);
Group workspace = pbiClient.Groups.CreateGroup(request);
```

After you've used a specific service principal profile to create and configure a workspace, you should continue to use that same profile to create and set up the workspace content. There's no need to invoke the `SetCallingContext` method to complete the setup.

Developer sample

We encourage you to download a sample application named **AppOwnsDataMultiTenant**.

This sample application was developed by using .NET 6 and ASP.NET, and it demonstrates how to apply the guidance and recommendations described in this article. You can review the code to learn how to develop a multitenancy application that implements the *Embed for your customer* scenario by using service principal profiles.

Related content

For more information about this article, check out the following resources:

- Service principal profiles for multitenancy apps in Power BI Embedded
- Migrate multi-customer applications to the service principal profiles model
- Profiles Power BI REST API operation group
- AppOwnsDataMultiTenant sample application
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Plan translation for multiple-language reports in Power BI

Article • 08/23/2024

When it comes to localizing Power BI items, such as semantic models and reports, there are three types of translations.

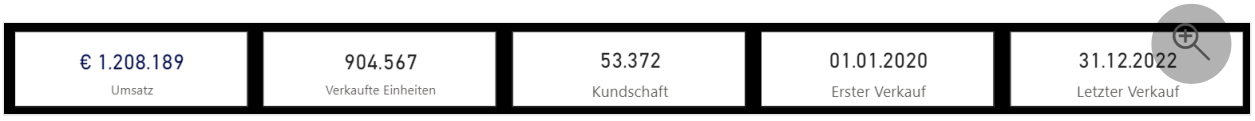
- Metadata translation
- Report label translation
- Data translation

In this article, learn about these types.

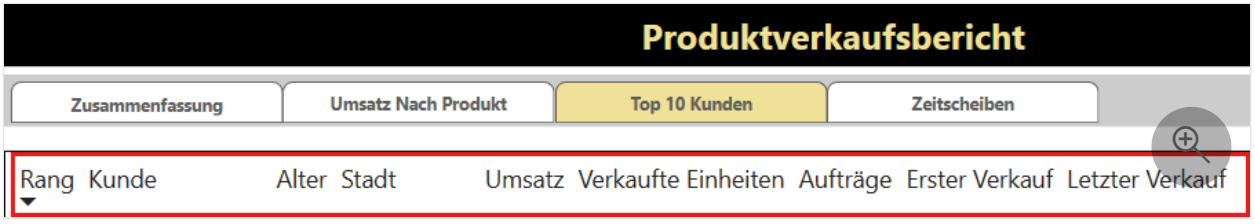
Metadata translation

Metadata translation provides localized values for semantic model object properties. The object types that support metadata translation include tables, columns, measures, hierarchies, and hierarchy levels. Metadata translation rarely provides a complete solution by itself.

The following screenshot shows how metadata translations provide German names for the measures displayed in Card visuals.



Metadata translation is also used to display column names and measure names in tables and matrices.



Metadata translations are the easiest to create, manage, and integrate into a Power BI report. By applying the features of Translations Builder to generate machine translations, you can add the metadata translations you need to build and test a Power BI report. Adding metadata translations to your semantic model is an essential first step. For more information, see Create multiple-language reports with Translations Builder.

Power BI support for metadata translation

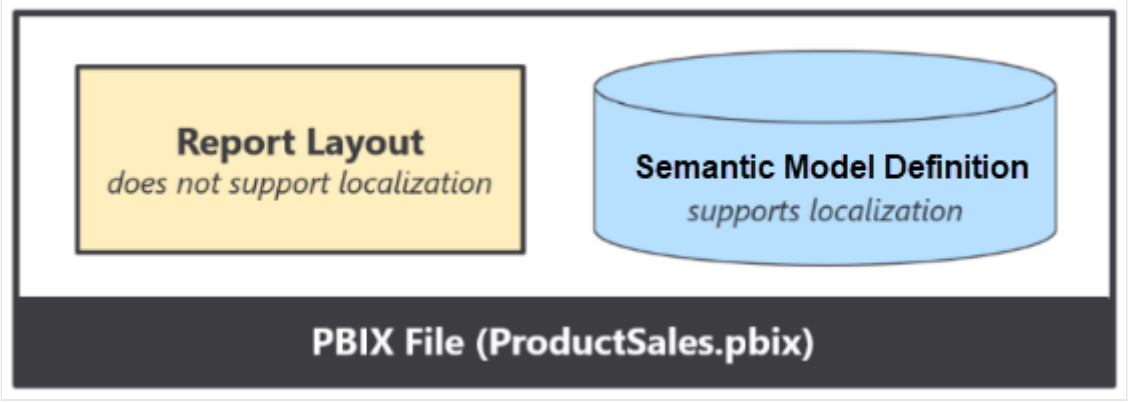
Metadata translation is the main localization feature in Power BI to build multiple-language reports. In Power BI, metadata translation support is integrated at the semantic model level.

A metadata translation represents the property for a semantic model object that's been translated for a specific language. If your semantic model contains a table with an English name of *Products*, you can add translations for the **Caption** property of this table object to provide alternative names. These names appear when the report is rendered in a different language.

In addition to the **Caption** property, which tracks an object's display name, semantic model objects also support adding metadata translations for two other properties, which are **Description** and **DisplayFolder**.

When you begin designing a semantic model that uses metadata translation, you can assume that you always need translations for the **Caption** property. If you require support for metadata translation for report authors who create and edit reports in the Power BI service, you also need to provide metadata translations for the **Description** and **DisplayFolder** properties.

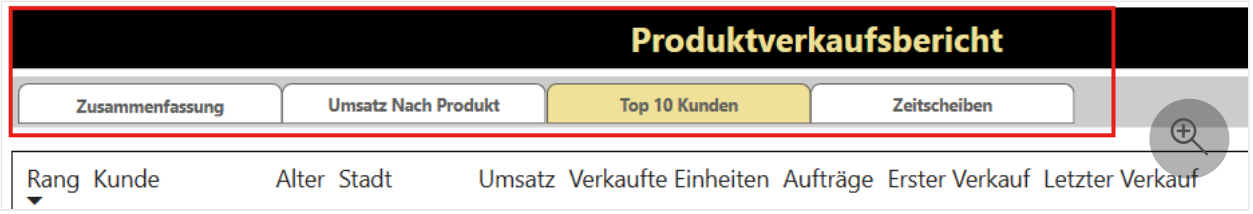
Power BI support for metadata translations only applies to semantic models. Power BI Desktop and the Power BI service don't support storing or loading translations for text values stored as part of the report layout.



If you add a textbox or button to a Power BI report and then add a hard-coded text value for a string displayed to the user, that text value is stored in the report layout. It can't be localized. Avoid using hard-coded text values. Page tabs display names can't be localized. You can design multiple-language reports so that page tabs are hidden and never displayed to the user.

Report label translation

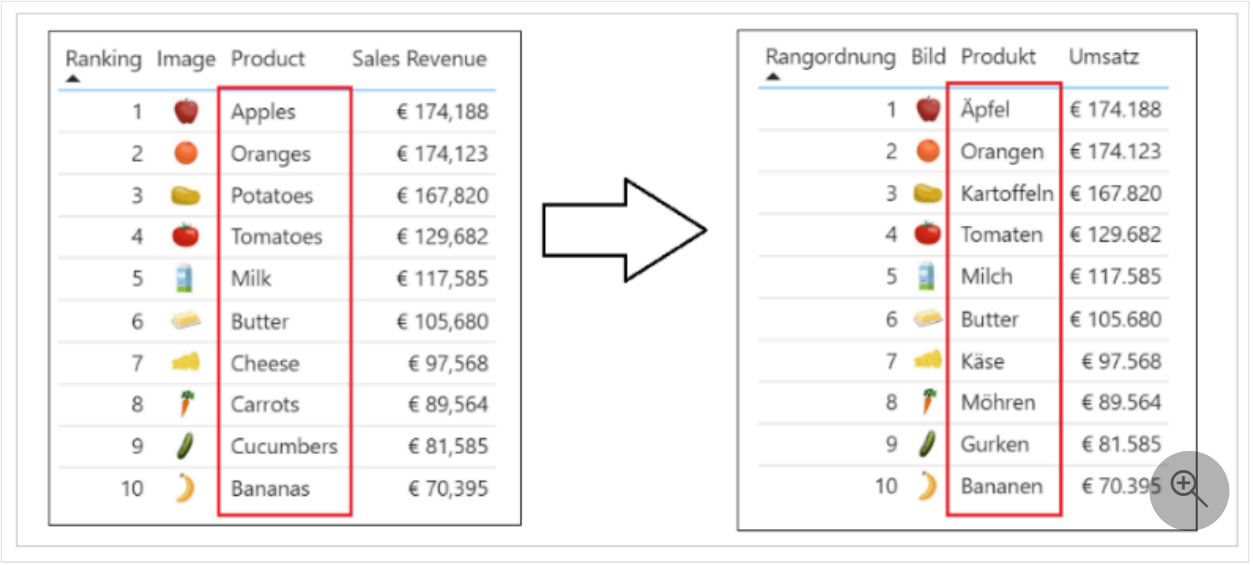
Report label translation provides localized values for text elements on a report that aren't directly associated with a semantic model object. Examples of report labels include the report title, section headings, and button captions. Here are examples of report label translations with the report title and the captions of navigation buttons.



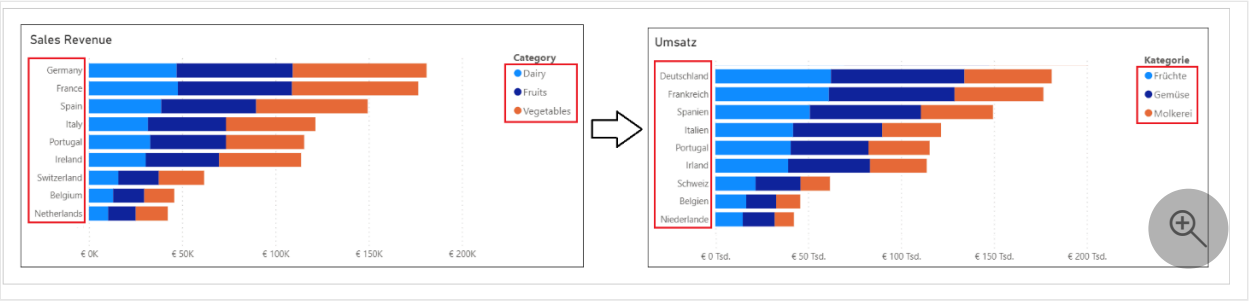
Report label translations are harder to create and manage than metadata translations because Power BI provides no built-in feature to track or integrate them. Translations Builder solves this problem using a *Localized Labels* table, which is a hidden table in the semantic model of a report. Add measures that track the required translations for each report label.

Data translation

Data translation provides translated values for text-based columns in the underlying data itself. Suppose a Power BI report displays product names imported from the rows of the **Products** table in an underlying database. Data translation is used to display product names differently for users who speak different languages. For example, some users see products names in English while other users see product names in other languages.



Data translations also appear in the axes of cartesian visuals and in legends.



Data translation is harder to design and implement than the other two types of translation. You must redesign the underlying data source with extra text columns for secondary language translations. Once the underlying data source has been extended with extra text columns, you can then use a powerful feature in Power BI Desktop called *Field Parameters*. This feature uses filters to control loading the data translations for a specific language.

A multiple-language report typically requires both metadata translations and report label translations. Some multiple-language projects require data translations, but others don't.

Related content

- Use locale values in multiple-language Power BI reports

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Use locale values in multiple-language Power BI reports

Article • 11/10/2023

Every report that loads in the Power BI service initializes with a user context that identifies a *language* and a geographical region known as a *locale*. In most cases, a locale identifies a country/region. The Power BI service tracks the combination of the user's language and locale using a *culture name*.

A culture name is usually a lower-case language identifier and an upper-case locale identifier separated by a hyphen. The culture name `en-US` identifies a user in the United States who speaks English. A culture name of `es-ES` identifies a user in Spain who speaks Spanish. A culture name of `fr-FR` identifies a user in France who speaks French. A culture name of `de-DE` identifies a user in Germany who speaks German.

Expand table

USERCULTURE	Language	Locale
en-US	English	United States
es-ES	Spanish	Spain
fr-FR	French	France
de-DE	German	Germany

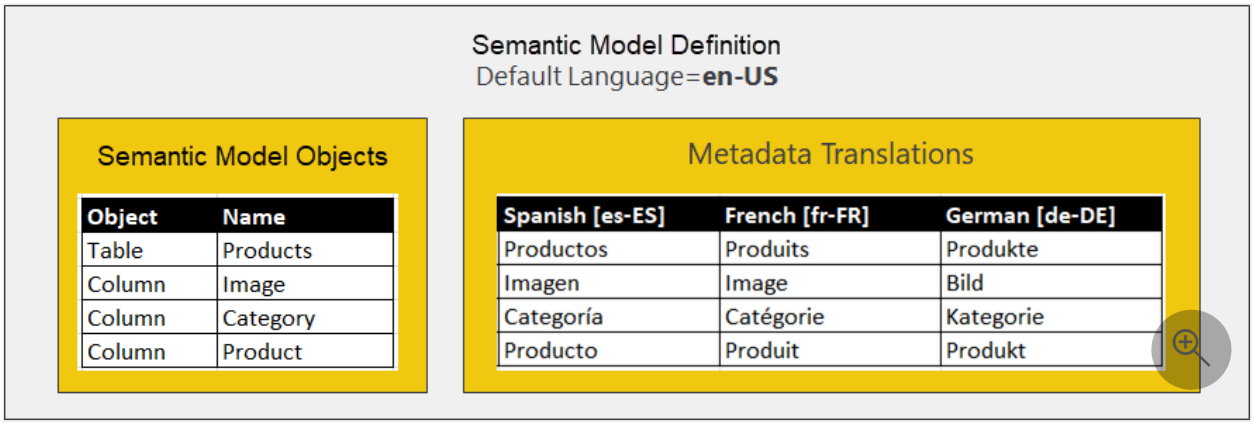
> **Note**
>
> In some cases, a culture name also includes other information. For example, there are two different culture names for the language Serbian in Serbia, which are `sr-Cyrl-RS` and `sr-Latn-RS`. The part in the middle known as the script (*Cyrl* and *Latn*) indicates whether to use the Cyrillic alphabet or the Latin alphabet. For more information, see RFC 4646.

For a list of culture name values, see ISO 639 Language codes and Online Browsing Platform.

Organize project for metadata translation

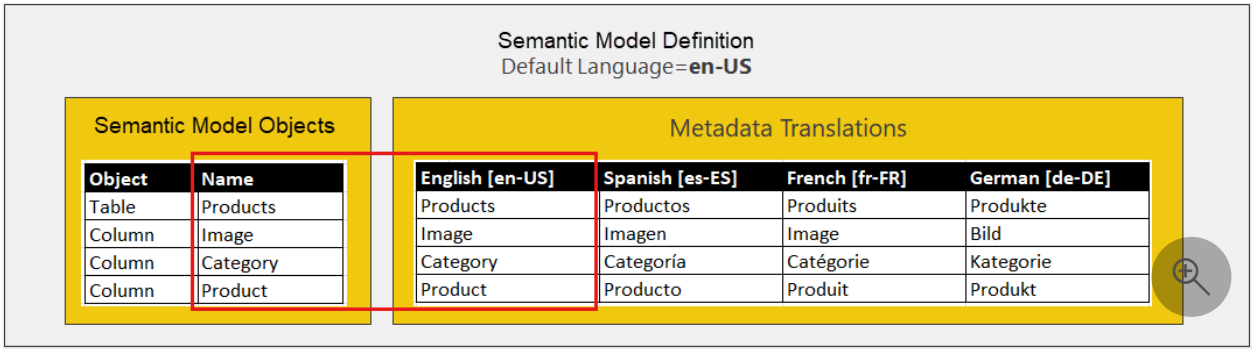
At the start of a project that involves creating a new Power BI semantic model with metadata translation, list the culture names that you plan to support. Next, extend the semantic model by adding metadata translations for each culture name.

The following diagram shows a semantic model that has a default language setting of `en-US`. The semantic model has been extended with metadata translations for three other culture names: `es-ES`, `fr-FR`, and `de-DE`.



Every metadata translation is associated with a specific culture name. Cultures names act as lookup keys that are used to add and retrieve metadata translations within the context of a Power BI semantic model.

You don't need to supply metadata translations for the default language of the semantic model. Power BI can just use the semantic model object names directly for that culture name. One way to think about this is that the semantic model object names act as a virtual set of metadata translations for the default language.



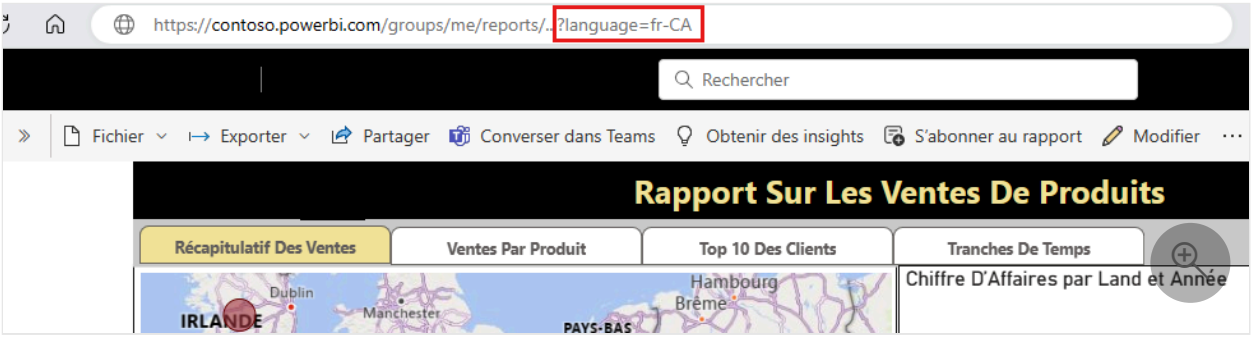
It's possible to explicitly add metadata translation for the default language. Use this approach sparingly. Power BI Desktop doesn't support loading metadata translations in its report designer. Instead, Power BI Desktop only loads semantic model object names. If you explicitly add metadata translations for the default language, Power BI reports look different in Power BI Desktop than they do in the Power BI service.

Load a report in Power BI

When a user navigates to a Power BI report with an HTTP GET request, the browser transmits an HTTP header named `Accept-Language` with a value set to a valid culture name. The following screenshot shows a GET request that transmits an `Accept-Language` header value of `en-US`.

When the Power BI service loads a report, it reads the culture name passed in the `Accept-Language` header and uses it to initialize the language and locale of the report loading context. On their devices, users can control which culture name is passed in the `Accept-Language` header value by configuring regional settings.

When you open a Power BI report in the Power BI service, you can override the `Accept-Language` header value by adding the `language` parameter at the end of the report URL and setting its value to a valid culture name. For example, you can test loading a report for a user in Canada who speaks French by setting the `language` parameter value to `fr-CA`.



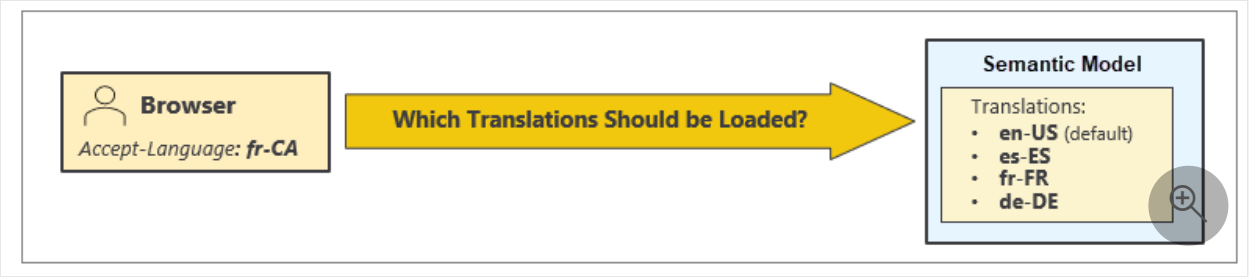
> **Note**
>
> Adding the `language` parameter to report URLs provides a convenient way to test metadata translations in the Power BI service. This technique doesn't require you to reconfigure any settings on your local machine or in your browser.

Support multiple locales for a single language

You might need to support multiple locales for a single spoken language. Consider a scenario with users who speak French but live in different countries, such as France,

Belgium, and Canada. You publish a semantic model with a default language of `en-US` and metadata translations for three more culture names including `es-ES`, `fr-FR`, and `de-DE`.

What happens when a French-speaking Canadian user opens report with an **Accept-Language** header value of `fr-CA`? Does the Power BI service load translations for French (`fr-FR`) or does it fall back on the English semantic model object names?

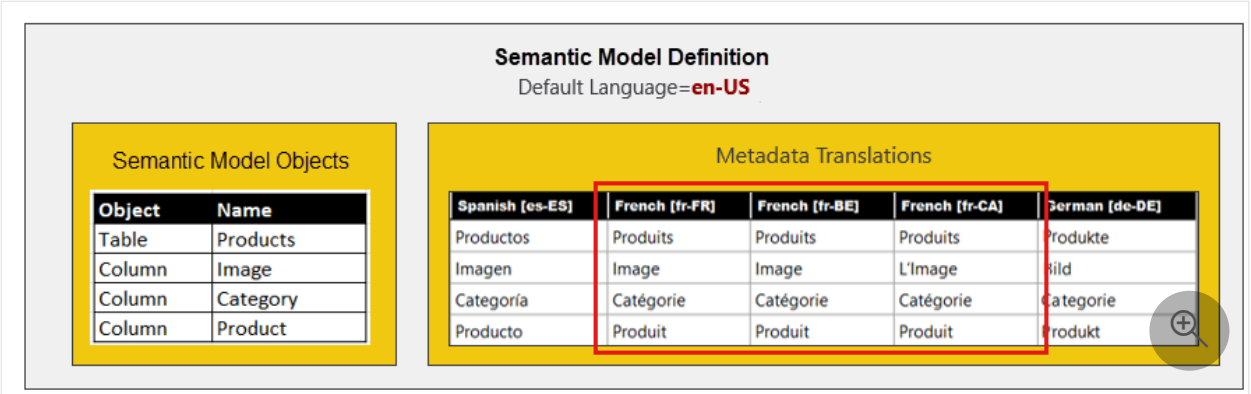


Measures currently act differently than tables and columns in Power BI. With measures, the Power BI service attempts to find the closest match. For the culture name of `fr-CA`, the names of measures would load using the metadata translations for `fr-FR`.

With tables and columns, the Power BI service requires an exact match between the culture name in the request and the supported metadata translations. If there isn't an exact match, the Power BI service falls back to loading semantic model object names. The names of tables and columns in this scenario would load using English semantic model object names.

> ⓘ **Note**
>
> This use of the default language for the names of tables and columns is a known issue for Power BI.

We recommend that you add metadata translation for any culture name you want to support. In this example, add three sets of French translations to support the culture names of `fr-FR`, `fr-BE` and `fr-CA`. The approach handles the scenario where the French translations for users in France are different from French translations for users in Canada.

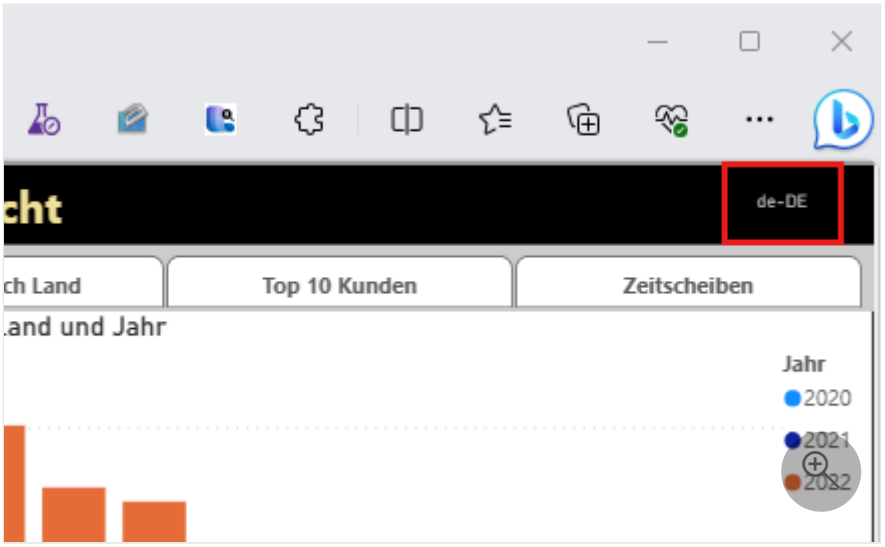


Implement translations using measures and USERCULTURE

Another feature in Power BI that helps with building multiple-language reports is the Data Analysis Expressions (DAX) `USERCULTURE` function. When called inside a measure, the `USERCULTURE` function returns the culture name of the current report loading context. This approach makes it possible to write DAX logic in measures that implement translations dynamically.

You can implement translations dynamically by calling `USERCULTURE` in a measure, but you can't achieve the same result with calculated tables or calculated columns. The DAX expressions for calculated tables and calculated columns get evaluated at semantic model load time. If you call the `USERCULTURE` function in the DAX expression for a calculated table or calculated column, it returns the culture name of the semantic model's default language. Calling `USERCULTURE` in a measure returns the culture name for the current user.

The example report displays the `USERCULTURE` return value in the upper right corner of the report banner. You don't typically display a report element like this in a real application.



This code is a simple example of writing a DAX expression for a measure that implements dynamic translations. You can use a `SWITCH` statement that calls `USERCULTURE` to form a basic pattern for implementing dynamic translations.

DAX

```
Product Sales Report Label = SWITCH( USERCULTURE() ),
  "es-ES", "Informe De Ventas De Productos",
  "fr-FR", "Rapport Sur Les Ventes De Produits",
  "fr-BE", "Rapport Sur Les Ventes De Produits",
  "fr-CA", "Rapport Sur Les Ventes De Produits",
  "de-DE", "Produktverkaufsbericht",
  "Product Sales Report"
)
```

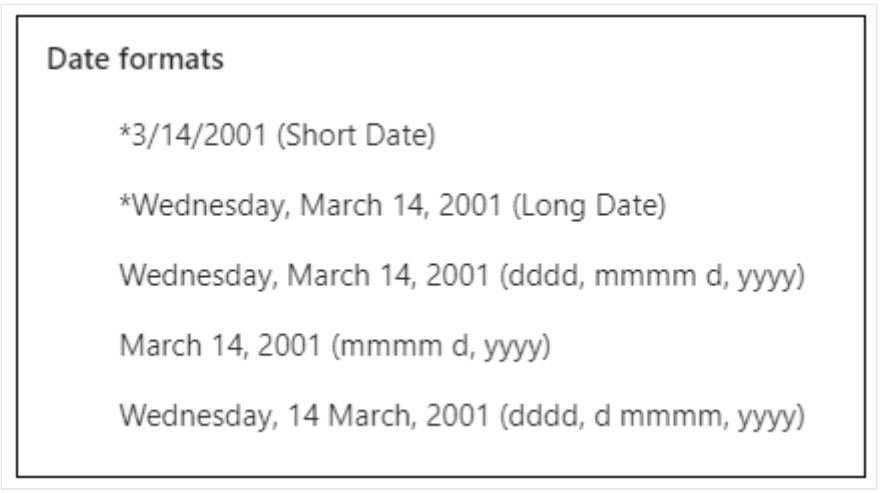
For more information, see Learn DAX basics in Power BI Desktop.

Format dates and numbers with current user locale

You can translate dynamically by writing a DAX expression in a measure with conditional logic based on the user's culture name. In most cases, you aren't required to write conditional DAX logic based on the user's locale because Power BI visuals automatically handle locale-specific formatting behind the scenes.

In a simple scenario, you build a report for an audience of report consumers that live in both New York (`en-US`) and in London (`en-GB`). All users speak English (`en`), but some live in different regions (`US` and `GB`) where dates and numbers are formatted differently. For example, a user from New York wants to see dates in a `mm/dd/yyyy` format while a user from London wants to see dates in a `dd/mm/yyyy` format.

Everything thing works as expected if you configure columns and measures using format strings that support regional formatting. If you're formatting a date, we recommend that you use a format string such as **Short Date** or **Long Date** because they support regional formatting.



Here are a few examples of how a date value formatted with **Short Date** appears when loaded under different locales.

Expand table

Locale	Format
en-US	12/31/2022
en-GB	31/12/2022
pt-PT	31-12-2022
de-DE	31.12.2022
ja-JP	2022/12/31

Related content

- Create multiple-language reports with Translations Builder

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Use best practices to localize Power BI reports

Article • 11/10/2023

When it comes to localizing software, there are some universal principles to keep in mind. The first is to plan for localization from the start of any project. It's harder to add localization support to an existing semantic model or report that was initially built without any regard for internationalization or localization.

This fact is especially true with Power BI reports because there are so many popular design techniques that don't support localization. Much of the work for adding localization support to existing Power BI reports involves undoing things that don't support localization. Only after that work can you move forward with design techniques that do support localization.

Package a semantic model and reports in project files

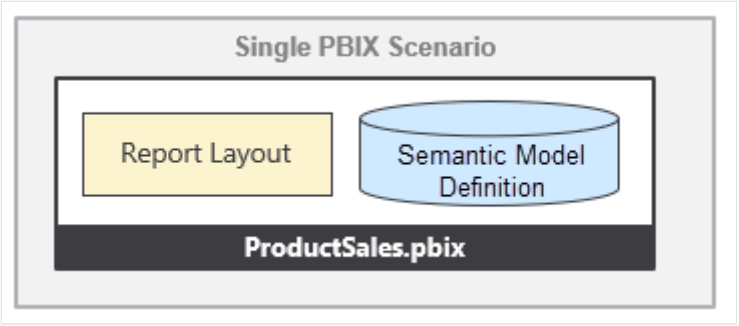
Before you proceed, you need to decide how to package your semantic model definitions and report layouts for distribution. There are two popular approaches used by content creators who work with Power BI Desktop.

- Single .pbix project file
- Multiple project files with a shared semantic model

For adding multiple-language support to a Power BI solution, choose either of these approaches.

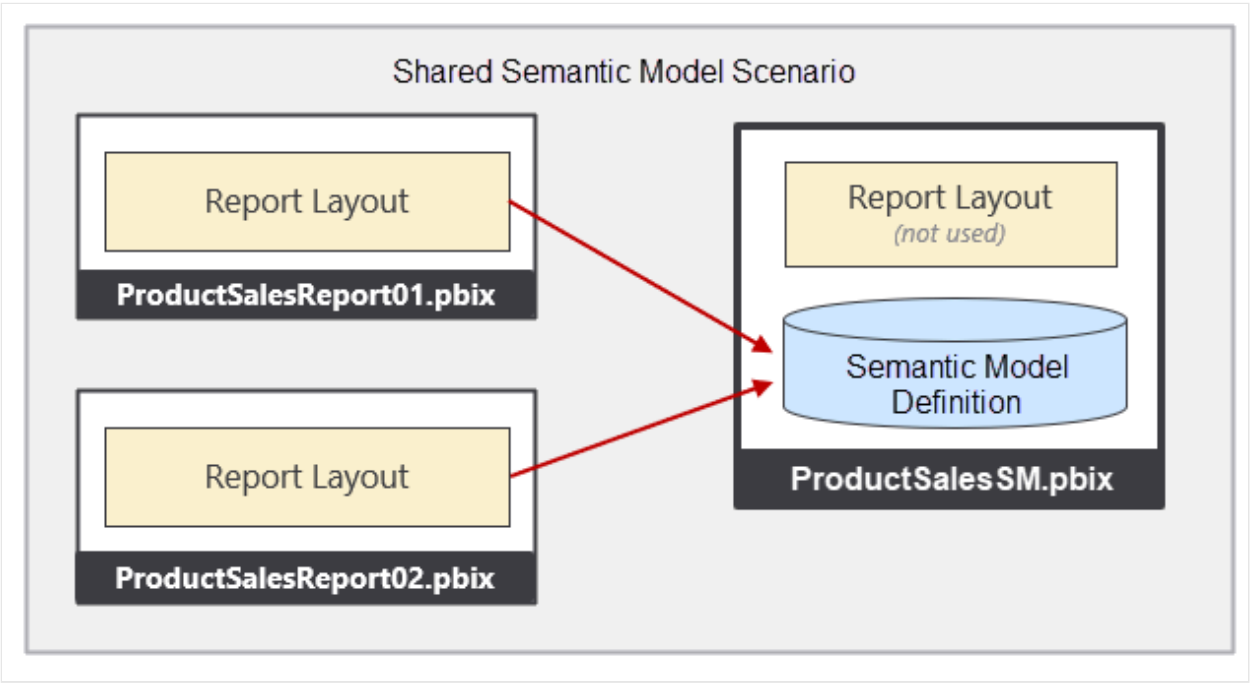
Single project file

You can package both a report layout and its underlying semantic model definition together. Deploy a reporting solution like this by publishing the project into a Power BI service workspace. If you need to update either the report layout or the semantic model definition, upgrade by publishing an updated version of the .pbix project file.



Shared semantic model

The single project file approach doesn't always provide the flexibility you need. Maybe one team is responsible for creating and updating semantic models while other teams are responsible for building reports. It might make sense to share a semantic model with reports in separate .pbix project files.



To use the shared semantic model approach, create one .pbix project file with a semantic model and an empty report, which remains unused. After this semantic model has been deployed to the Power BI service, report builders can connect to it using Power BI Desktop to create report-only .pbix files.

This approach makes it possible for the teams building reports to build .pbix project files with report layouts that can be deployed and updated independently of the underlying semantic model. For more information, see Connect to semantic models.

Account for text size

Another important concept in localization is to plan for growth. A label that's 400 pixels wide when displayed in English could require a greater width when translated into another language. If you optimize the width of your labels for text in English, you might find that translations in other languages introduce unexpected line breaks or get cut off. These effects compromise the user experience.

Adding a healthy degree of padding to localized labels is the norm when developing internationalized software. It's essential that you test your reports with each language you plan to support. You need to be sure that your report layouts look the way you expect with any language you choose to support.

Related content

- Create multiple-language reports with Translations Builder

Feedback

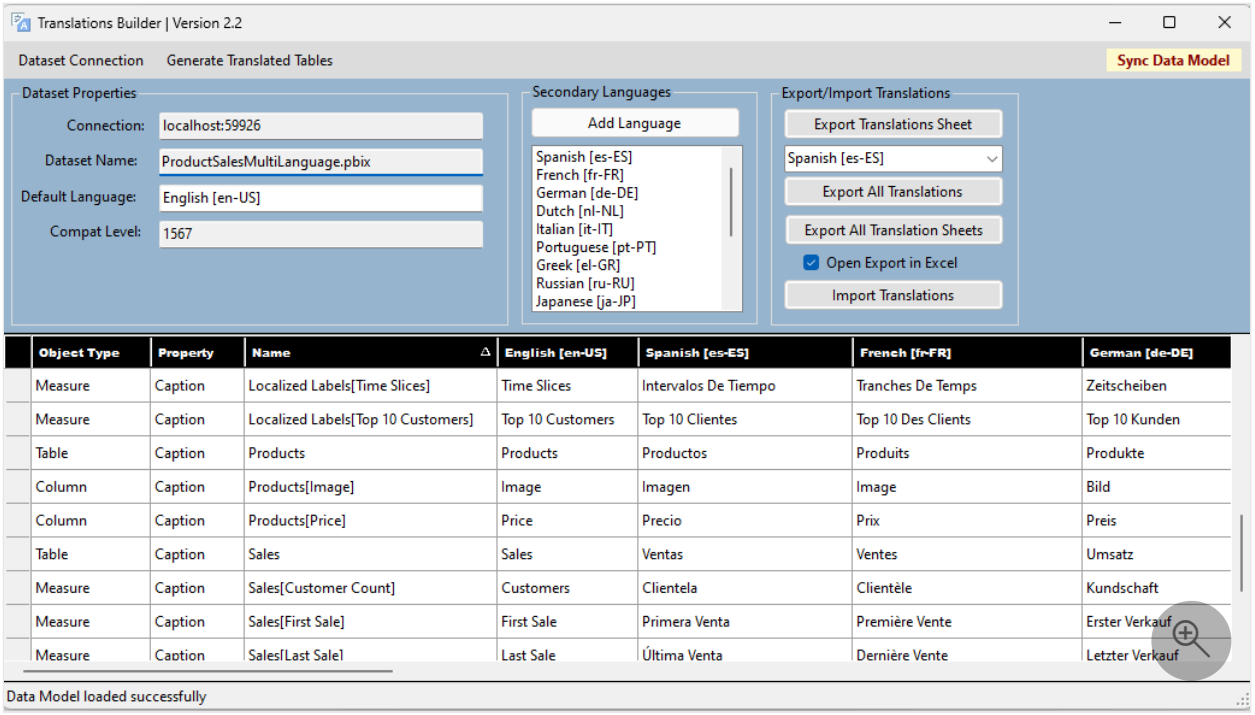
Was this page helpful? Yes No

Provide product feedback | Ask the community

Create multiple-language reports with Translations Builder

Article • 11/10/2023

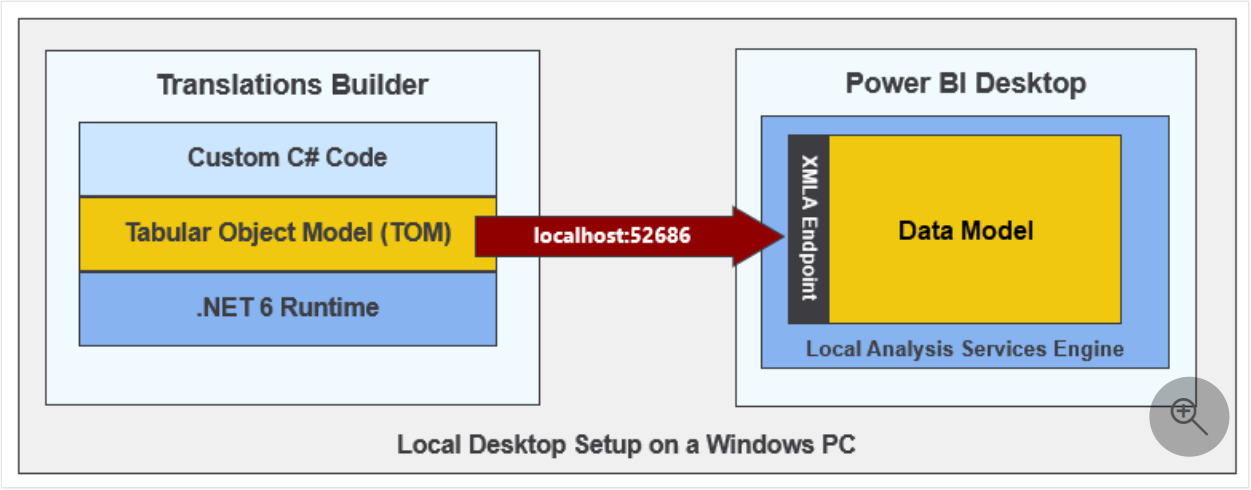
Content creators can use Translations Builder to add multiple-language support to .pbix project files in Power BI Desktop. The following screenshot shows what Translations Builder looks like when working with a simple .pbix project that supports a few secondary languages.



Translations Builder is an external tool developed for Power BI Desktop using C#, .NET 6, and Windows Forms. Translations Builder uses an API called *Tabular Object Model* (TOM) to update data models that are loaded into memory and run in a session of Power BI Desktop.

Translations Builder does most of its work by adding and updating the metadata translations associated with data model objects including tables, columns, and measures. There are also cases in which Translations Builder creates new tables in a data model to implement strategies to handle aspects of building multiple-language reports.

When you open a .pbix project in Power BI Desktop, the data model defined inside the .pbix file is loaded into memory in a local session of the Analysis Services engine. Translations Builder uses TOM to establish a direct connection to the data model of the current .pbix project.

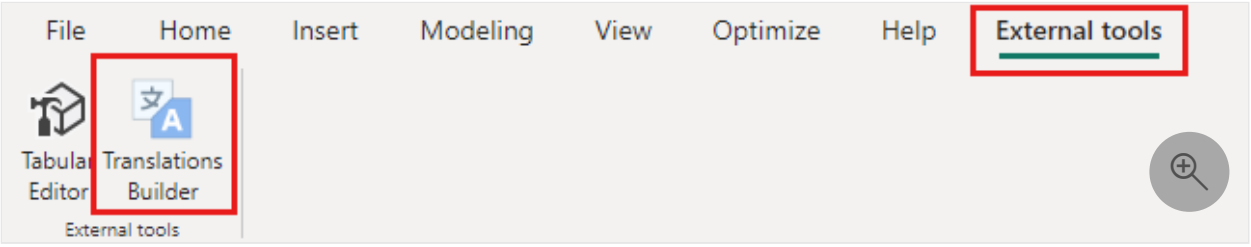


Open Translations Builder

If you don't already have Power BI Desktop installed, see Get Power BI Desktop.

On the same computer where you run Power BI Desktop, download and install Translations Builder by using the Translations Builder Installation Guide.

After you install Translations Builder, you can open it directly from Power BI Desktop in the **External Tools** ribbon. The Translations Builder project uses external tools integration support. For more information, see External tools in Power BI Desktop.



When you launch an external tool like Translations Builder, Power BI Desktop passes startup parameters to the application, including a connection string. Translations Builder uses the connection string to establish a connection back to the data model that's loaded in Power BI Desktop.

This approach allows Translations Builder to display data model information and to provide commands to automate adding metadata translations. For more information, see Translations Builder Developers Guide.

Translations Builder allows a content creator to view, add, and update metadata translations using a two-dimensional grid. This *translations grid* simplifies the user experience because it abstracts away the low-level details of reading and writing metadata translation associated with a dataset definition. You work with metadata translations in the translation grid similar to working with data in an Excel spreadsheet.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Measure	Caption	Sales[Rank]	Rank	Rango	Rang	Rang
Table	Caption	Products	Products	Productos	Produits	Produkte
Column	Caption	Products[Price]	Price	Precio	Prix	Preis
Column	Caption	Products[Image]	Image	Imagen	Image	Bild
Table	Caption	Calendar	Calendar	Calendario	Calendrier	Kalender
Column	Caption	Calendar[Date]	Date	Fecha	Date	Datum
Column	Caption	Calendar[Year]	Year	Año	Année	Jahr
Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos	Rapport Sur Les Ventes De Produits	Produktverkaufsbericht
Measure	Caption	Localized Labels[Sales Summary]	Sales Summary	Resumen De Ventas	Récapitulatif Des Ventes	Zusammenfassung

Related content

- Add a language to a report in Translations Builder

Feedback

Was this page helpful? Yes No

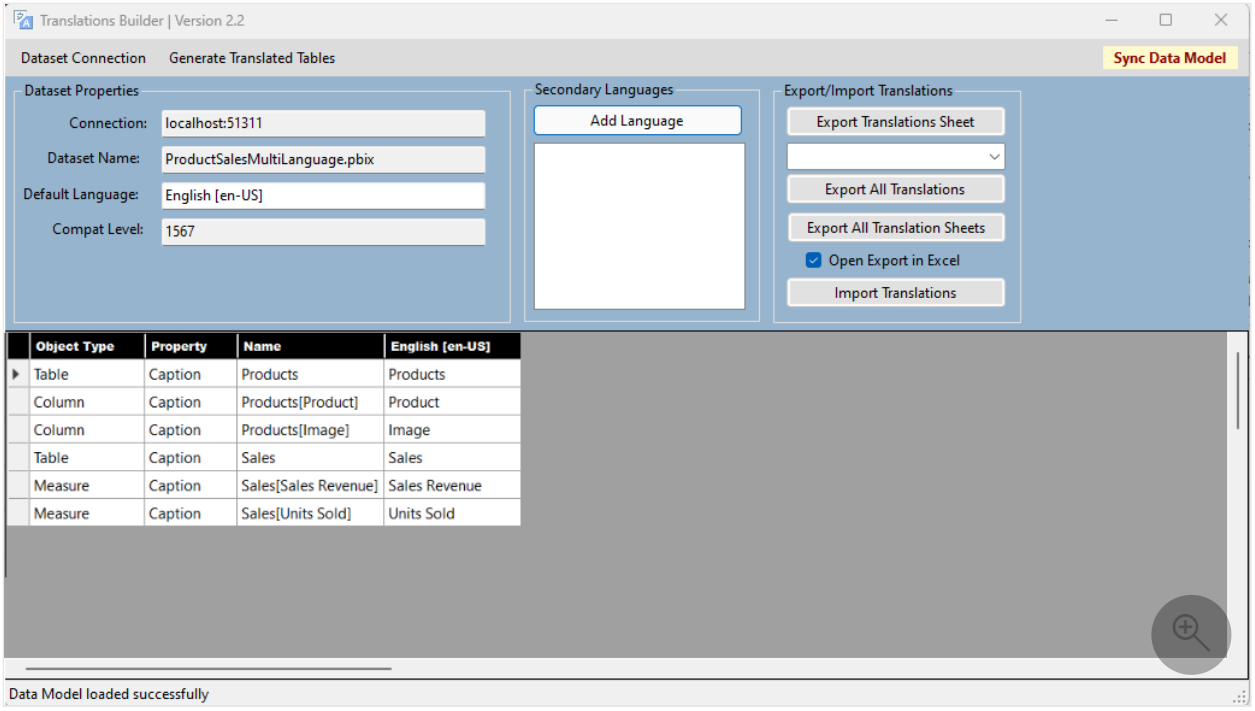
Provide product feedback | Ask the community

Add a language to a report in Translations Builder

Article • 07/31/2023

When you open a .pbix project in Translations Builder for the first time, the translation grid displays a row for each unhidden table, measure, and column in the project's underlying data model. The translation grid doesn't display rows for data model objects in the data model that are hidden from the report view. Hidden objects aren't displayed on a report and don't require translations.

The following screenshot shows the starting point for a simple data model before it's been modified to support secondary languages.



> **Note**
>
> If you haven't installed Translations Builder yet, see Create multiple-language reports with Translations Builder.

If you examine the translation grid for this .pbix project, you can see the first three columns contain read-only columns used to identity each metadata translation. Each metadata translation has an **Object Type**, a **Property**, and a **Name**. Translations for the **Caption** property are always used. You can add translations for the **Description** and **DisplayFolder** properties if necessary.

The fourth column in the translation grid always displays the translations for the data model's default language and locale, which in this case is **English [en-US]**.

	Object Type	Property	Name	English [en-US]
▶	Table	Caption	Products	Products
	Column	Caption	Products[Product]	Product
	Column	Caption	Products[Image]	Image
	Table	Caption	Sales	Sales
	Measure	Caption	Sales[Sales Revenue]	Sales Revenue
	Measure	Caption	Sales[Units Sold]	Units Sold

> ⓘ **Note**
>
> Translations Builder makes it possible to update the translations for the default language. Use this technique sparingly. It can be confusing because translations for the default language don't load in Power BI Desktop.

Add languages

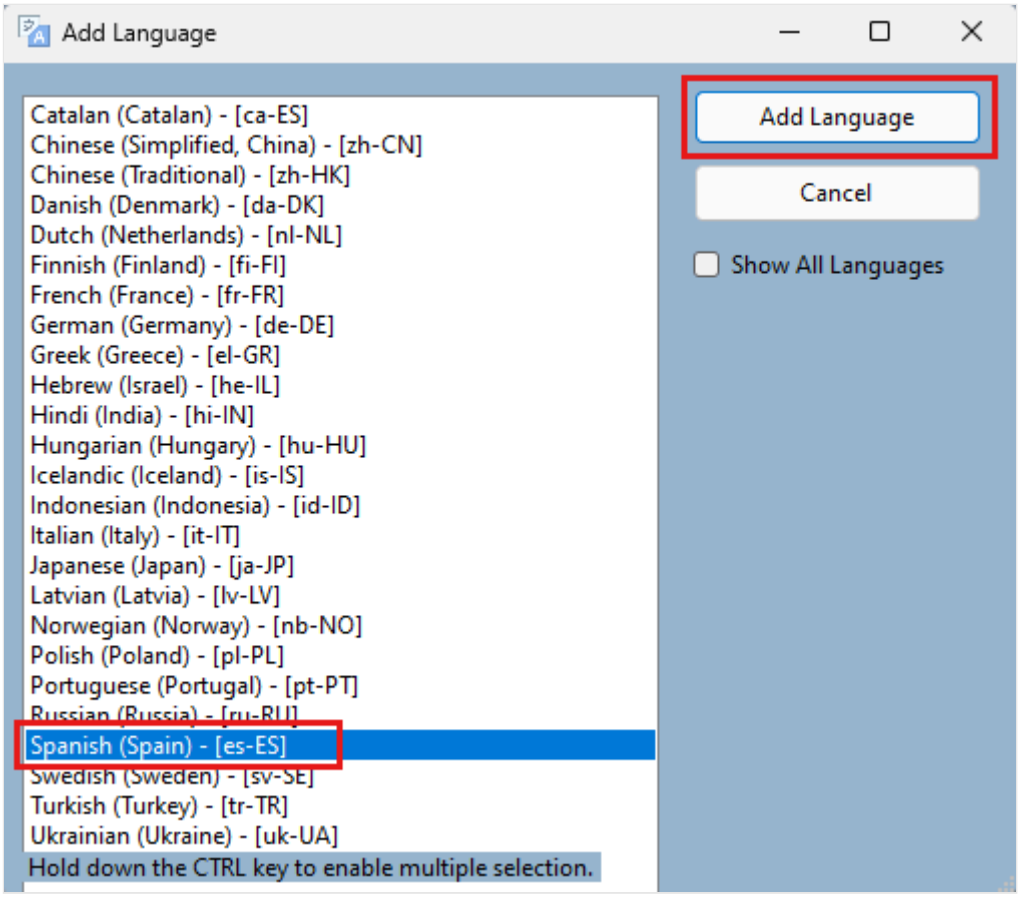
Translations Builder provides an **Add Language** option to add secondary languages to the project's data model.

Translations Builder doesn't add metadata translations for a specific language. Instead, it adds metadata translations for a culture name that identifies both a language and a locale. For more information, see Use locale values in multiple-language Power BI reports.

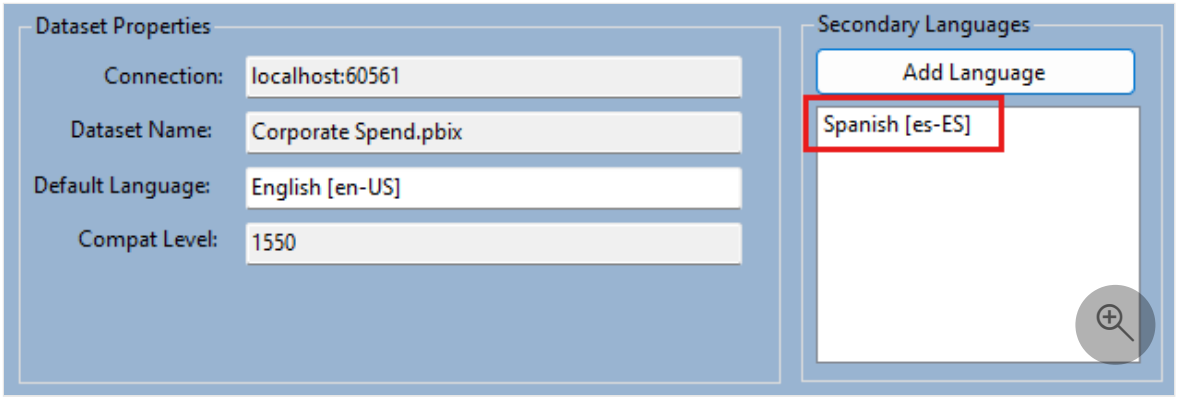
Translations Builder abstracts away the differences between a language and a culture name to simplify the user experience. Content creators can think in terms of languages instead of culture names.

To add one or more secondary languages, follow these steps.

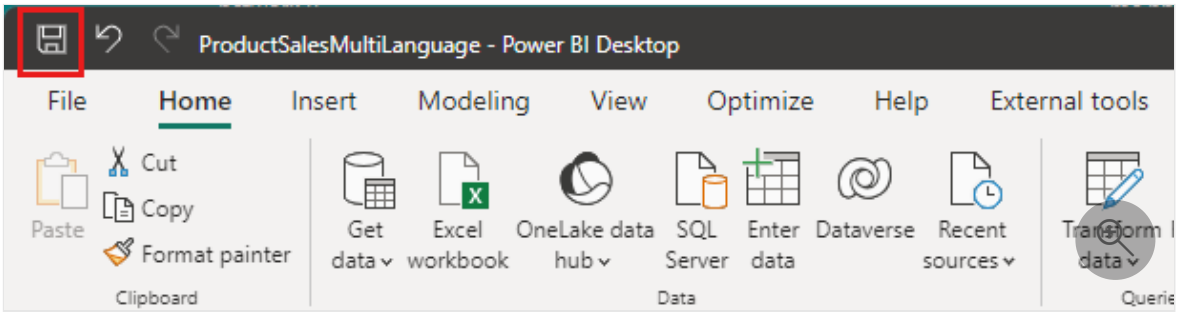
1. Select **Add Language** to display the **Add Language** dialog box.
2. Select a language in the list or use **Ctrl** to select multiple languages.
3. Select **Add Language**.



The added language or languages now appear in the **Secondary Languages** list.



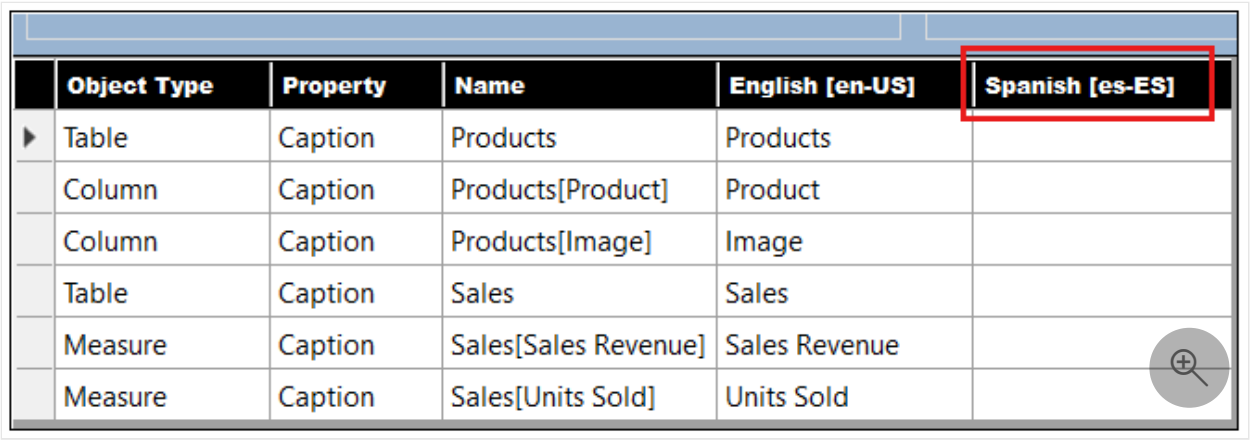
4. In Power BI Desktop, select **Save**.



> **Important**

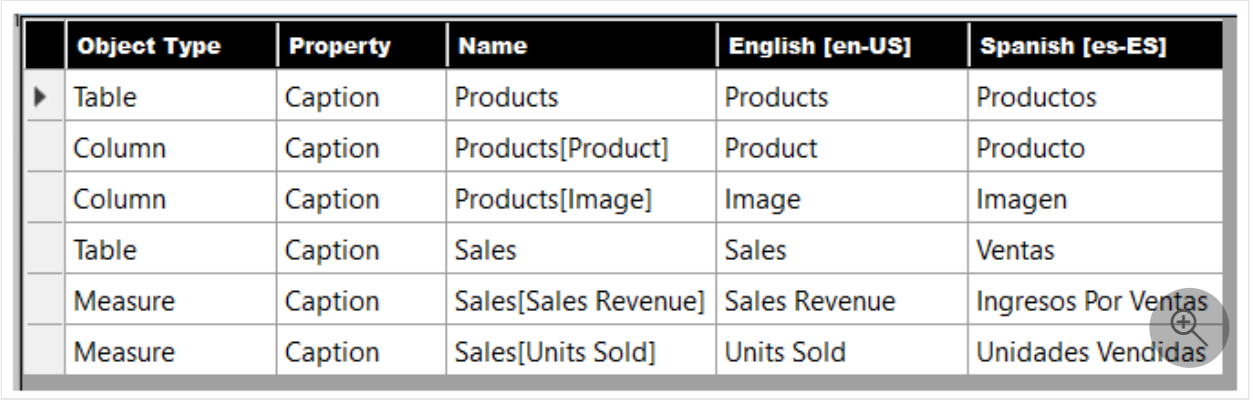
> Translations Builder can modify the data model loaded in memory, but it can't save the in-memory changes back to the underlying .pbix file. Always return to Power BI Desktop and select the **Save** command after you add languages or create or update translations.

Adding a new language adds a new column of editable cells to the translations grid.



Object Type	Property	Name	English [en-US]	Spanish [es-ES]
Table	Caption	Products	Products	
Column	Caption	Products[Product]	Product	
Column	Caption	Products[Image]	Image	
Table	Caption	Sales	Sales	
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	
Measure	Caption	Sales[Units Sold]	Units Sold	

If content creators speak all the languages involved, they can add and update translations for secondary languages directly in the translation grid with an Excel-like editing experience.



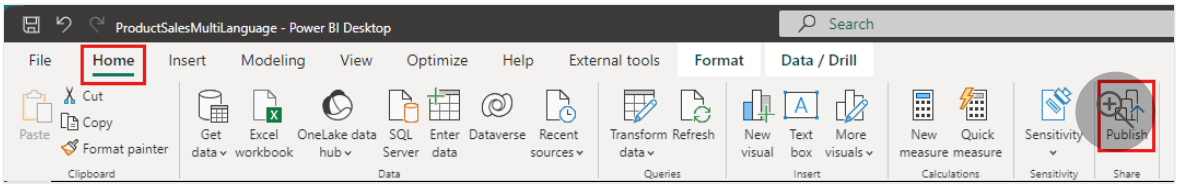
Object Type	Property	Name	English [en-US]	Spanish [es-ES]
Table	Caption	Products	Products	Productos
Column	Caption	Products[Product]	Product	Producto
Column	Caption	Products[Image]	Image	Imagen
Table	Caption	Sales	Sales	Ventas
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas

Test translations in the Power BI service

You can't verify your multiple-language work in Power BI Desktop. Instead, you must test your work in the Power BI service in a workspace associated with a Premium capacity. After you add translation support with Translations Builder, follow these steps:

1. In Power BI Desktop, save changes to the underlying .pbix file.

2. In the **Home** ribbon, select **Publish**.



3. In the **Publish to Power BI** dialog box, highlight a workspace and then choose **Select**.

4. When the publishing finishes, select the link to open the project in the Power BI service.

After the report loads with its default language, select the browser address bar and add the following **language** parameter to the report URL.

```HTTP
?language=es-ES
```

When you add the **language** parameter to the end of the report URL, assign a value that is a valid culture name. After you add the **language** parameter and press **Enter**, you can verify that the parameter has been accepted by the browser as it reloads the report.

If you forget to add the question mark (**?**) or if you don't format the **language** parameter correctly, the browser rejects the parameter and removes it from the URL. After you correctly load a report using a **language** parameter value of **es-ES**, you should see the user experience for the entire Power BI service UI change from English to Spanish.

The report also displays the Spanish translations for the names of columns and measures.

Implement multiple-language workflow

After you test your work and verify that the translations are working properly, you can store the .pbix file in a source control system such as GitHub or Azure Repos. This

approach can be part of an application lifecycle management (ALM) strategy where support for secondary languages and translations evolves over time.

As you begin to work with secondary languages and translations to localize a .pbix project, follow the same set of steps:

1. Make changes in Power BI Desktop.
2. Publish the .pbix project to the Power BI service.
3. Test your work with a browser in the Power BI service using the **language** parameter.
4. Repeat these steps until you complete all the translations.

Embed Power BI reports using a specific language and locale

If you're developing with Power BI embedding, you can use the Power BI JavaScript API to load reports with a specific language and locale. This task is accomplished by extending the `config` object passed to `powerbi.embed` with a `localeSettings` object that contains a `language` property as shown in the following code.

JavaScript

```
let config = {
  type: "report",
  id: reportId,
  embedUrl: embedUrl,
  accessToken: embedToken,
  tokenType: models.TokenType.Embed,
  localeSettings: { language: "de-DE" }
};

let report = powerbi.embed(reportContainer, config);
```

Related content

- Add a Localized Labels table to a Power BI report

Feedback

Was this page helpful? Yes No

Add a Localized Labels table to a Power BI report

Article • 07/31/2023

Report label translations provide localized values for text elements on a report that aren't directly associated with a data model object. Examples of report labels are the text values for report titles, section headings, and button captions. Power BI provides no built-in features to track or integrate report labels. Translations Builder uses *Localized Labels* tables to support this approach.

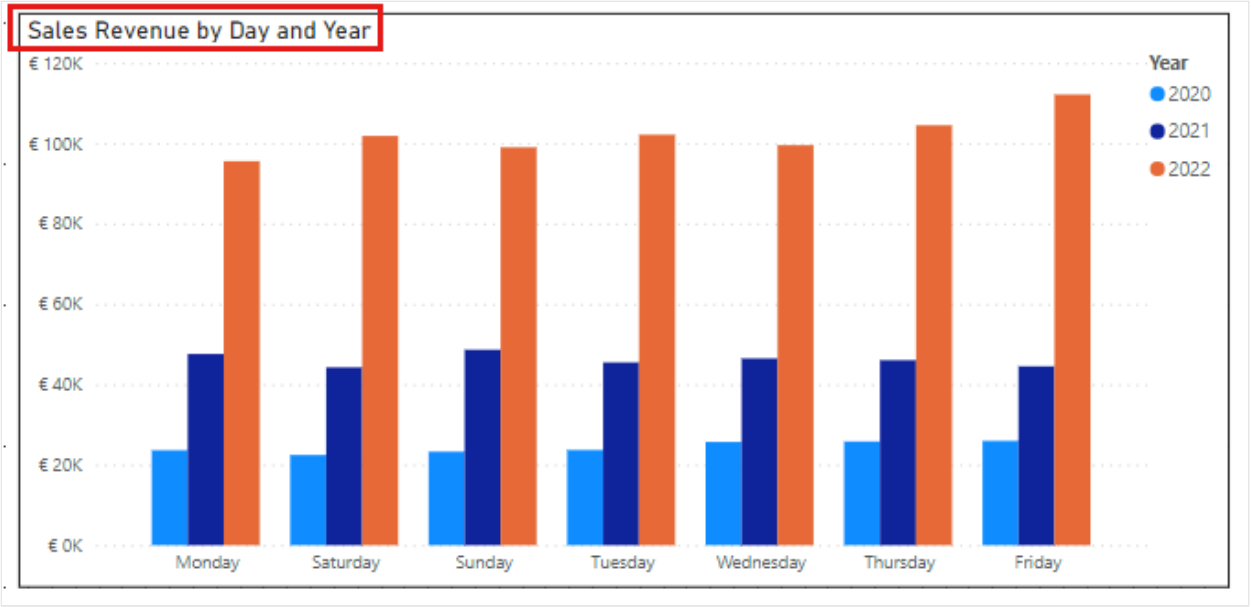
> ⓘ **Note**
>
> If you haven't installed Translations Builder yet, see Create multiple-language reports with Translations Builder.

Compare localized labels and hard-coded text

There are some design techniques for building data models and reports with Power BI Desktop that you should avoid when you build multiple-language reports. These elements cause problems due to a lack of localization support:

- Using text boxes or buttons with hard-coded text values.
- Adding a hard-coded text value for the title of a visual.
- Displaying page tabs to the user.

Any hard-coded text value that you add to the report layout can't be localized. Suppose you add a column chart to your report. By default, a Cartesian visual such as a column chart is assigned a dynamic value to its **Title** property. That value is based on the names of the columns and measures that have been added into the data roles, such as **Axis**, **Legend**, and **Values**.



The default **Title** property for a Cartesian visual is dynamically parsed together in a fashion that supports localization. As long as you supply metadata translations for the names of columns and measures in the underlying data model definition, the **Title** property of the visual uses the translations. So, if you translate **Sales Revenue**, **Day**, and **Year**, the visual creates a localized title.

The following table shows how the default **Title** property of this visual is updated for each of these five languages.

Expand table

Language	Visual Title
English (en-US)	`Sales Revenue by Day and Year`
Spanish (es-ES)	`Ingresos Por Ventas por Día y Año`
French (fr-FR)	`Chiffre D'Affaires par Jour et Année`
German (de-DE)	`Umsatz nach Tag und Jahr`

You might not like the dynamically generated visual **Title**, but don't replace it with hard-coded text. Any hard-coded text for the **Title** property can't be localized. Either leave the visual **Title** property with its default value or use the Localized Labels table strategy to create report labels that support localization.

Use the Localized Labels table strategy

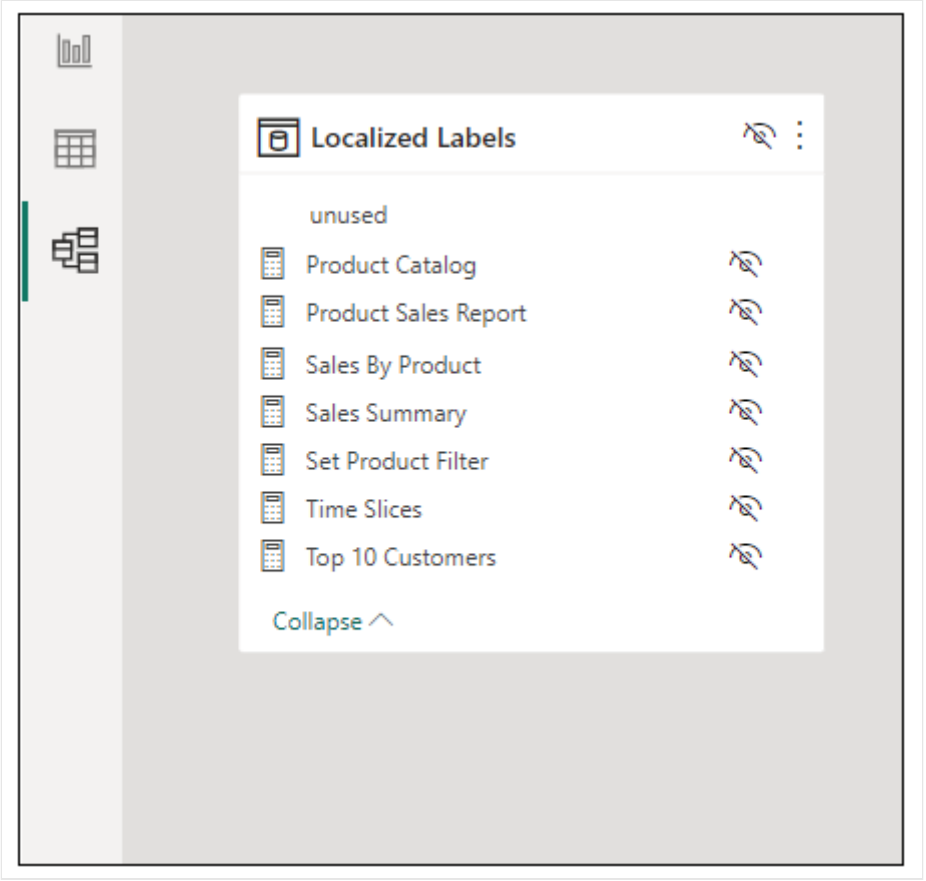
The Power BI localization features are supported at the data model level but not at the report layout level. Using a **Localized Labels** table builds on the fact that Power BI

supports metadata translations for specific types of data model objects, including measures. When you add a report label by using Translations Builder, it automatically adds a new measure to the **Localized Labels** table behind the scenes.

After a measure has been created for each report label, Power BI can store and manage its translations in the same fashion that it does for metadata translations. In fact, the Localized Labels table strategy uses metadata translations to implement report label translations.

Translations Builder creates the **Localized Labels** table and adds a measure each time you need a report label. The **Localized Labels** table is created as a hidden table. You can do all the work to create and manage report labels inside the Translation Builder user experience. There's no need to inspect or modify the **Localized Labels** table using the Power BI Desktop Model or Data views.

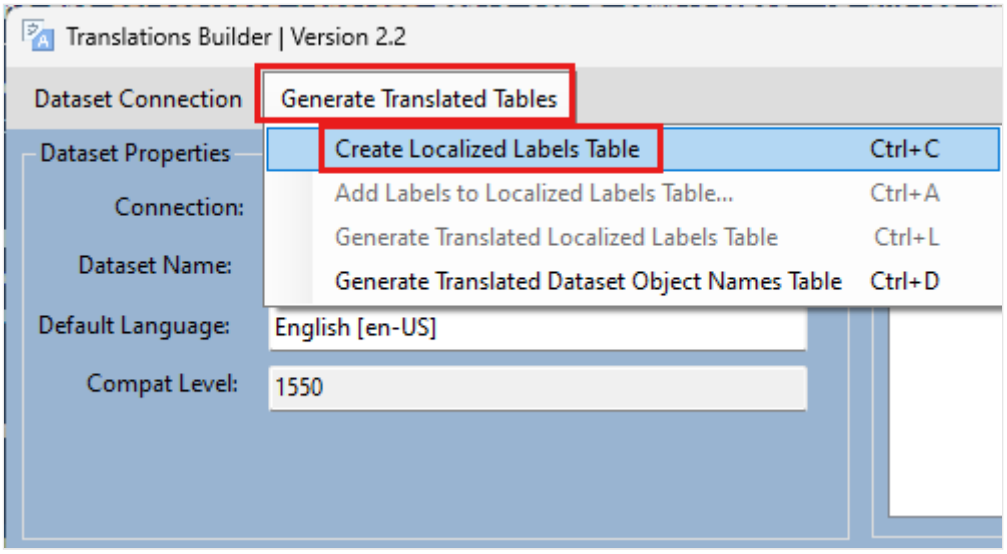
Here's an example of the **Localized Labels** table from the example project. It provides localized report labels for the report title, visual titles, and captions for navigation buttons used throughout the report.



Create the Localized Labels table

You can create the **Localized Labels** table for a .pbix project:

1. From the **Generate Translated Tables** menu, select **Create Localized Labels Table**.



2. An informational dialog box asks if you want more information about the **Localized Labels** table strategy. Select **Yes** to review documentation or **No** to proceed.

After you create the **Localized Labels** table, there are three sample report labels as shown in the following screenshot. In most cases, you want to delete these sample report labels and replace them with the actual report labels required on the current project.

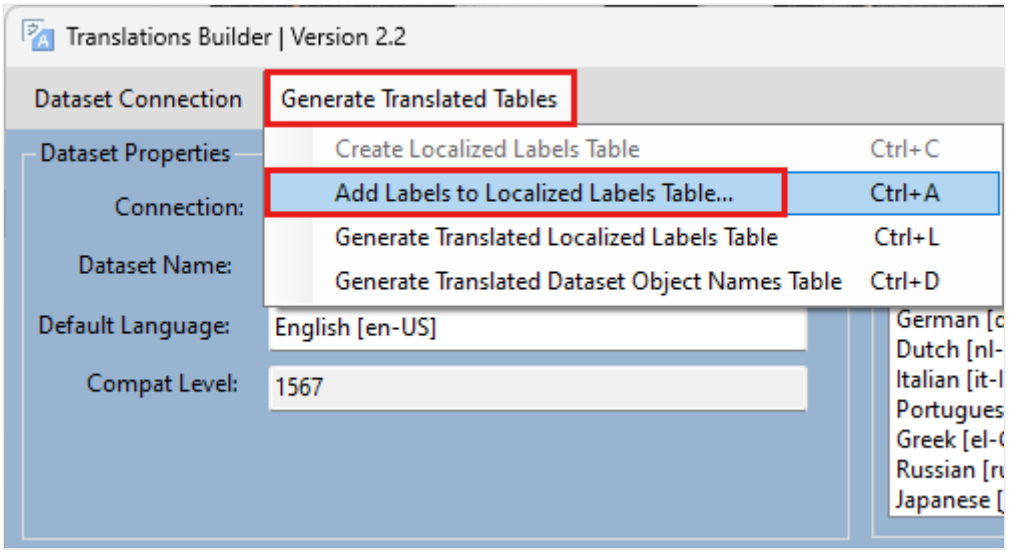
Object Type	Property	Name	English [en-US]	Spanish [es-ES]	Dutch [nl-NL]	French [fr-FR]	German [de-DE]	Italian [it-IT]
Table	Caption	Products	Products	Productos	Producten	Produits	Produkte	Prodotti
Column	Caption	Products[Product]	Product	Producto	Product	Produit	Produkt	Prodotto
Column	Caption	Products[Image]	Image	Imagen	Beeld	Image	Bild	Immagine
Table	Caption	Sales	Sales	Ventas	Verkoop	Ventes	Umsatz	Vendite
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Omzet	Chiffre D'Affaires	Umsatz	Ricavi Delle Vendite
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Verkochte Eenheden	Unités Vendues	Verkaufte Einheiten	Unità Vendute
Measure	Caption	Localized Labels[My Report Title]	My Report Title					
Measure	Caption	Localized Labels[My Button Caption]	My Button Caption					
Measure	Caption	Localized Labels[My Visual Title]	My Visual Title					

There's no need to interact with the **Localized Labels** table in Power BI Desktop. You can add and manage all the report labels you need in Translations Builder.

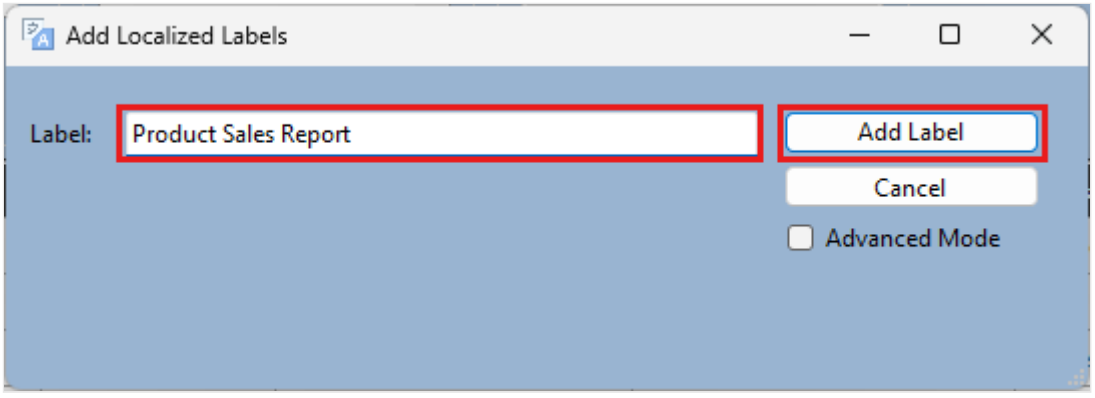
Populate the Localized Labels table

To create your first report label, follow these steps:

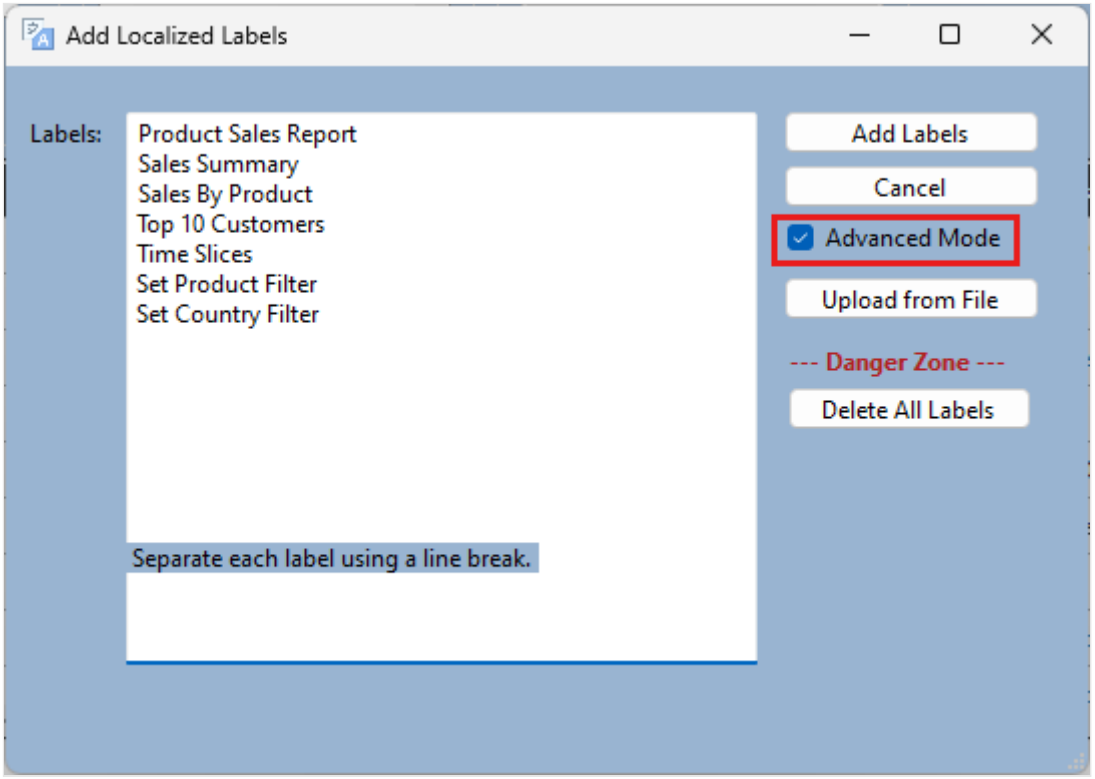
1. From the **Generate Translated Tables** menu, select **Add Labels to the Localized Labels Table**. You can also run the command using the shortcut key **Ctrl**+**A**.



2. Add report labels one at a time by typing the text for the label. Then select **Add Label**.



Alternatively, select **Advanced Mode** to add labels as a batch.



After you add the report labels to your .pbix project, they appear in the translation grid. Now you can add and edit localized label translations just like any other type of translation in the translation grid.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]	Dutch [nl-NL]	Italian [it-IT]
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkaufte Einheiten	Verkochte Eenheden	Unità Vendute
Measure	Caption	Localized Labels[Sales Summary]	Sales Summary	Resumen De Ventas	Récapitulatif Des Ventes	Zusammenfassung	Verkoopoverzicht	Riepilogo Delle Vendite
Measure	Caption	Localized Labels[Top 10 Customers]	Top 10 Customers	Top 10 Clientes	Top 10 Des Clients	Top 10 Kunden	Top 10 Klanten	Top 10 Clienti
Measure	Caption	Localized Labels[Time Slices]	Time Slices	Intervalos De Tiempo	Tranches De Temps	Zeitscheiben	Tijdsegmenten	Intervalli Di Tempo
Measure	Caption	Sales[Customer Count]	Customers	Clientela	Clientèle	Kundschaft	Klanten	Clientela
Measure	Caption	Sales[Last Sale]	Last Sale	Última Venta	Dernière Vente	Letzter Verkauf	Laatste Verkoop	Ultima Vendita
Measure	Caption	Sales[Rank]	Rank	Rango	Rang	Rang	Rang	Rango
Measure	Caption	Sales[First Sale]	First Sale	Primera Venta	Première Vente	Erster Verkauf	Eerste Verkoop	Prima Vendita
Measure	Caption	Localized Labels[Set Product Filter]	Set Product Filter	Establecer Filtro De Producto	Définir Le Filtre De Produit	Produktfilter Einstellen	Productfilter Instellen	Imposta Filtro Prodotto

About the Localized Labels table

Translations Builder only populates the translation grid with data model objects that aren't hidden from **Report View**. The measures in the **Localized Labels** table are hidden from **Report View** and they provide the one exception to the rule that excludes hidden objects from being displayed in the translation grid.

In the Localized Labels table strategy, you can create, manage, and store report labels in the same .pbix project file that holds the metadata translations for the names of tables, columns, and measures. The Localized Labels table strategy can merge metadata translations and report label translations together in a unified experience in the translation grid. There's no need to distinguish between metadata translations and report label translations when it comes to editing translations or when using Translations Builder features to generate machine translations.

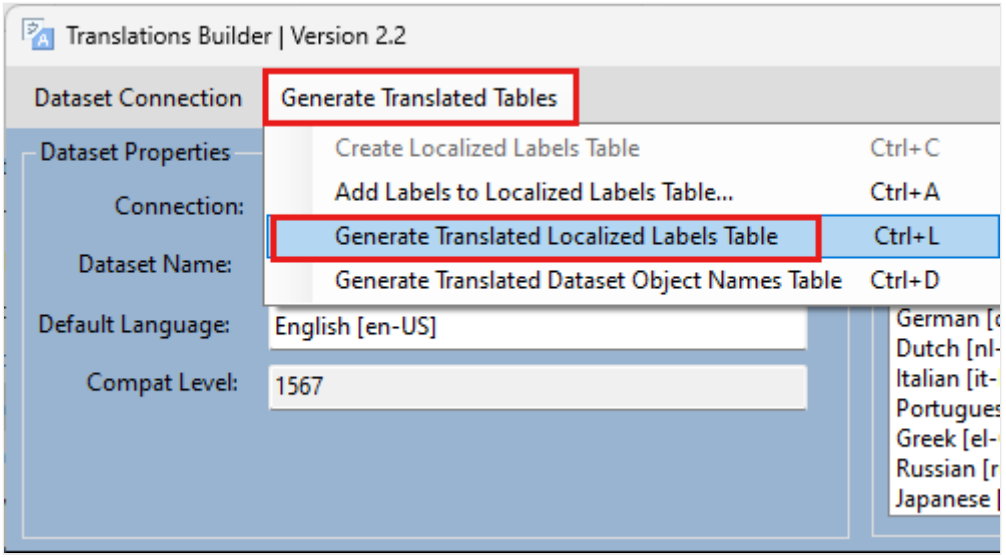
There are other popular localization techniques that track report label translations in a separate CSV file. While these techniques work, they aren't as streamlined. Report label translations must be created separately and managed differently from the metadata translations in a .pbix project. This strategy allows for report label translations and metadata translations to be stored together and managed in the same way.

Generate the translated Localized Labels table

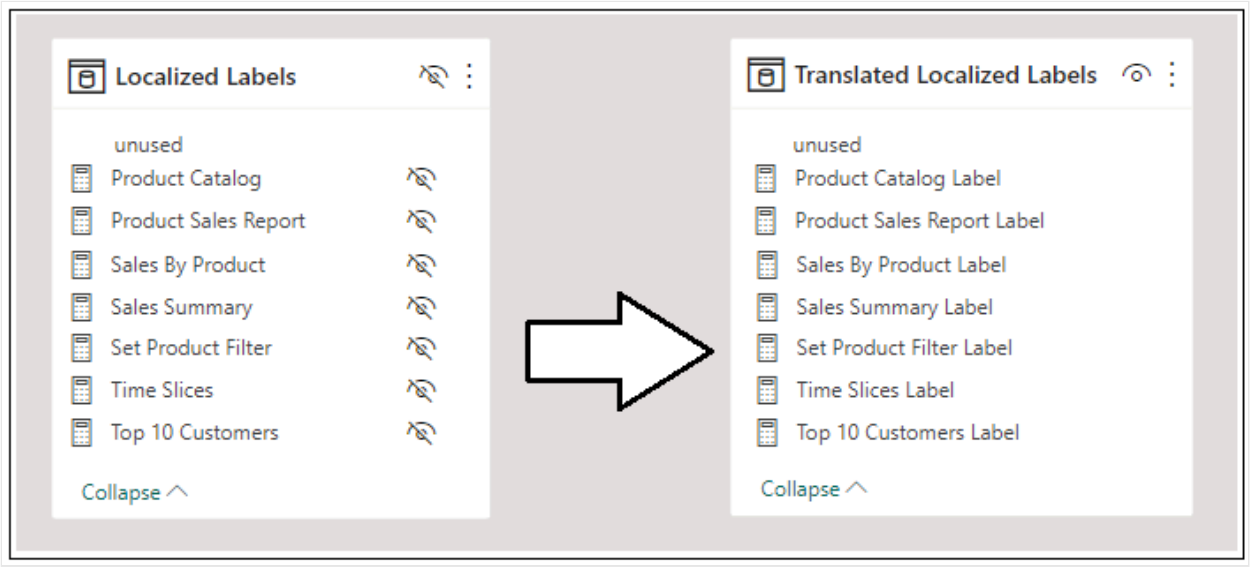
The **Localized Labels** table contains a measure with translations for each report label in a .pbix project. The measures inside the **Localized Labels** table are hidden and aren't intended to be used directly by report authors. Instead, the strategy is based on running code to generate a second table. The **Translated Localized Labels** table has measures that are meant to be used directly on a report page.

To create a **Translated Localized Labels** table, follow these steps.

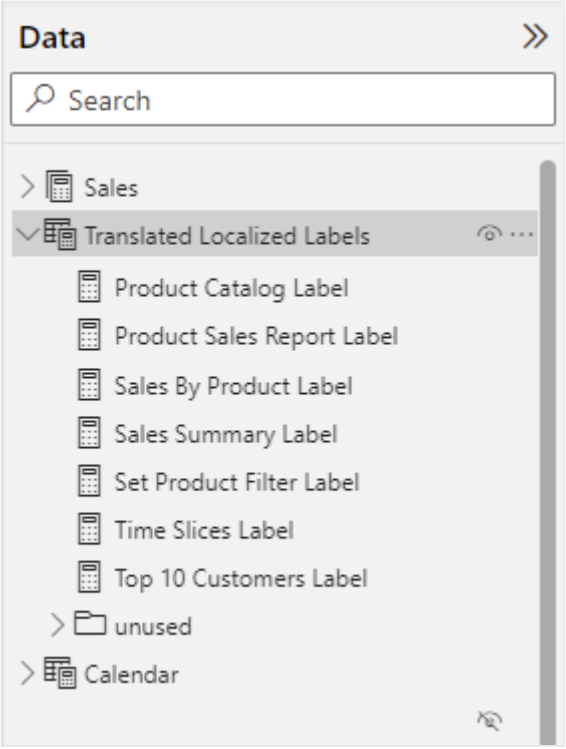
- In Translations Builder, from the **Generate Translated Tables** menu, select **Generate Translated Localized Labels Table**.



The first time you generate the **Translated Localized Labels** table, Translations Builder creates the table and populates it with measures. After that, generating the table deletes all the measures in the **Translated Localized Labels** table and recreates them. This action synchronizes all the report label translations between the **Localized Labels** table and the **Translated Localized Labels** table.



Unlike the **Localized Labels** table, the **Translated Localized Labels** table isn't hidden in the **Report View**. The table provides measures that are intended to be used as report labels in a report. Here's how the **Translated Localized Labels** table appears to a report author in the **Data** pane when the report is in the **Report View** in Power BI Desktop.



Every measure in the **Translated Localized Labels** table has a name that ends with the word **Label**. The reason for this fact is that two measures in the same data model can't have the same name. Measure names must be unique on a project-wide basis. It's not possible to create measures in the **Translated Localized Labels** table that have the same name as the measures in the **Localized Labels** table.

If you examine the machine-generated Data Analysis Expressions (DAX) expressions for measures in the **Translated Localized Labels** table, they're based on the same DAX pattern shown in Implement translations using measures and USERCULTURE. This pattern uses the DAX function `USERCULTURE` together with the `SWITCH` function to return the best translation for the current user. This DAX pattern defaults to the data model's default language if no match is found.

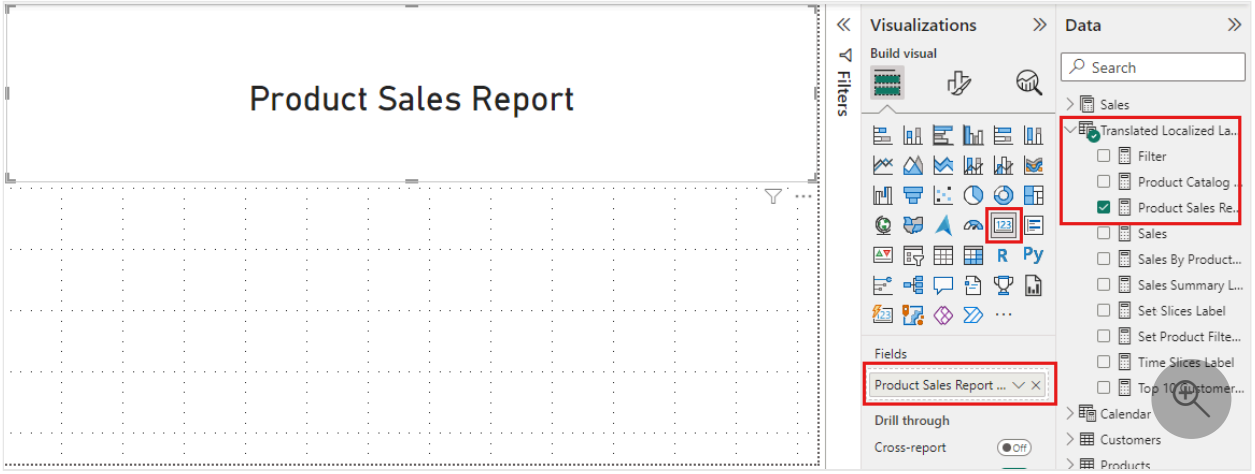
```
Product Sales Report Label = SWITCH(USERCULTURE()),
    "es-ES", "Informe De Ventas De Productos",
    "fr-FR", "Rapport Sur Les Ventes De Produits",
    "de-DE", "Produktverkaufsbericht",
    "nl-NL", "Productverkooprapport",
    "it-IT", "Report Sulle Vendite Dei Prodotti",
    "pt-PT", "Relatório De Vendas De Produtos",
    "el-GR", "Αναφορά Πωλήσεων Προϊόντων",
    "ru-RU", "Отчет О Продажах Продукции",
    "ja-JP", "製品販売レポート",
    "zh-CN", "产品销售报告",
    "hi-IN", "उत्पाद बिक्री रिपोर्ट",
    "he-IL", "דוח מכירות מוצרים",
    "fr-BE", "Rapport Sur Les Ventes De Produits",
    "fr-CA", "Rapport Sur Les Ventes De Produits",
    "fr-CH", "Rapport Sur Les Ventes De Produits",
    "zh-HK", "產品銷售報告",
    "Product Sales Report"
)
```

You must run **Generate Translated Localized Labels Table** anytime you make changes to the **Localized Labels** table.

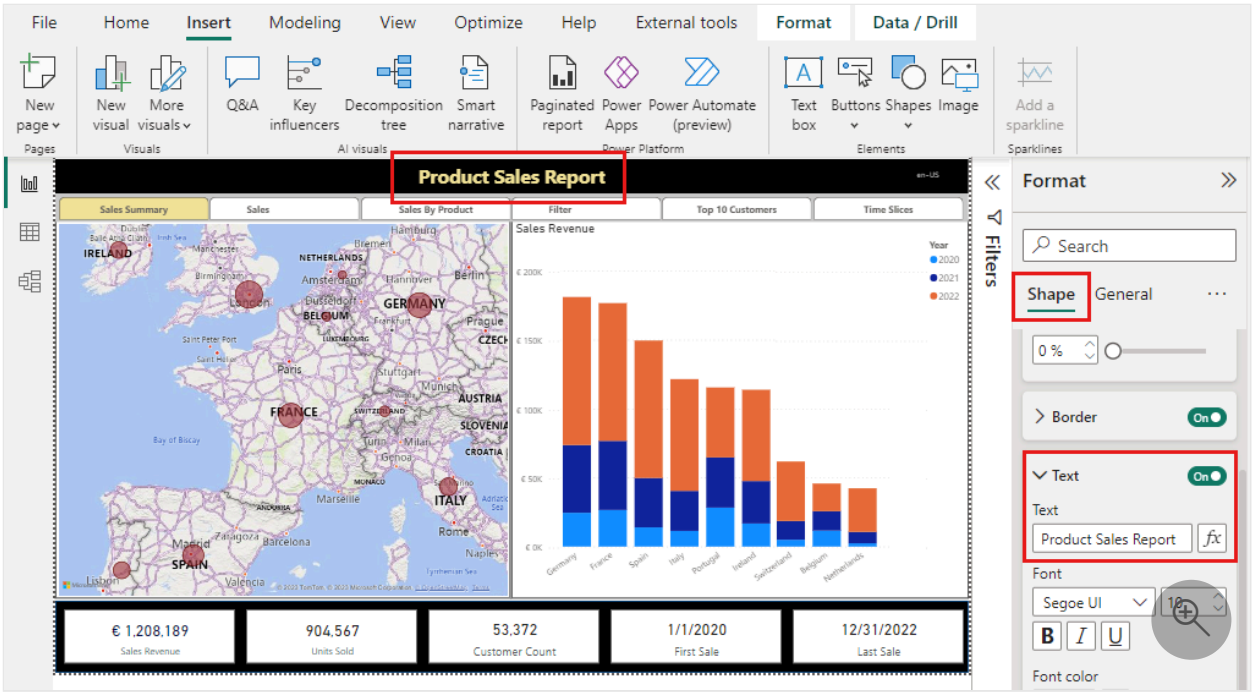
Don't edit the DAX expressions for measures in the **Translated Localized Labels** table. Any edits you make are lost because all the measures in this table are deleted and recreated each time you generate the table.

Surface localized labels on a report page

Report labels are implemented as dynamic measures in the **Translated Localized Labels** table. That fact makes them easy to surface in a Power BI report. For example, you can add a Card visual to a report and then configure its **Data** role in the **Visualizations** pane with a measure from the **Translated Localized Labels** table.



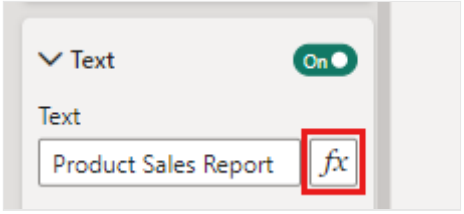
The example multiple-language project uses a *Rectangle* shape to display the localized report label for the report title. The following screenshot shows how to select a Rectangle shape and navigate to configure its **Text** property value in the **Shape** > **Text** section in the **Format** pane.



The **Text** property of a shape can be configured with a hard-coded string. You must avoid hard-coding text values into the report layout when you create multiple-language reports. To use a localized measure, follow these steps.

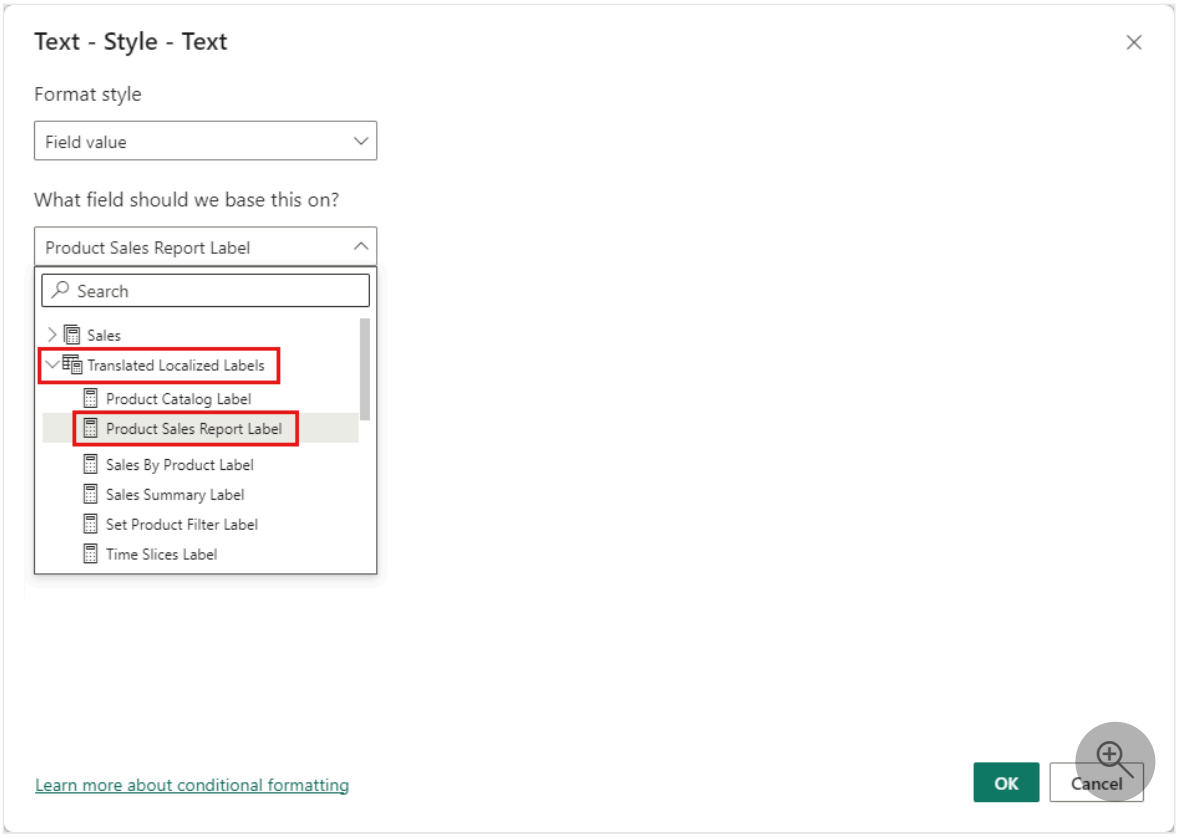
1. In Power BI Desktop, select the shape, a Rectangle, in this example.

2. Under Format, select **Shape** > **Text**. In the **Text** pane, select the ***fx*** button.

   Power BI Desktop displays a dialog box that allows you to configure the **Text** property of the Rectangle shape.

3. In the **Text - Style - Text** dialog box, expand the **Translated Localized Labels** table and select any measure.



4. Select **OK**.

You can use the same technique to localize a visual **Title** using a measure from the **Translated Localized Labels** table.

Related content

- Generate machine translations using Azure Translator Service

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Generate machine translations using Azure Translator Service

Article • 11/10/2023

One of the biggest challenges in building multiple-language reports is managing the language translation process. You must ensure that the quality of translations is high. Be sure that the translated names of tables, columns, measures, and labels don't lose their meaning when translated to another language. In most cases, acquiring quality translations requires human translators to create or at least review translations as part of the multiple-language report development process.

While human translators are typically an essential part of the end-to-end process, it can take a long time to send out translation files to a translation team and then to wait for them to come back. With all the recent industry advances in AI, you can also generate machine translations using a Web API that can be called directly from an external tool such as Translations Builder. If you generate machine translations, you have something to work with while waiting for a translation team to return their high-quality human translations.

While machine translations aren't always guaranteed to be high quality, they do provide value in the multiple-language report development process.

- They can act as translation placeholders so you can begin your testing by loading reports using secondary languages to see if there are layout issues or unexpected line breaks.
- They can provide human translators with a better starting point because they just need to review and correct translations instead of creating every translation from scratch.
- They can be used to quickly add support for languages where there are legal compliance issues and organizations are facing fines or litigation for noncompliance.

Generate machine translations

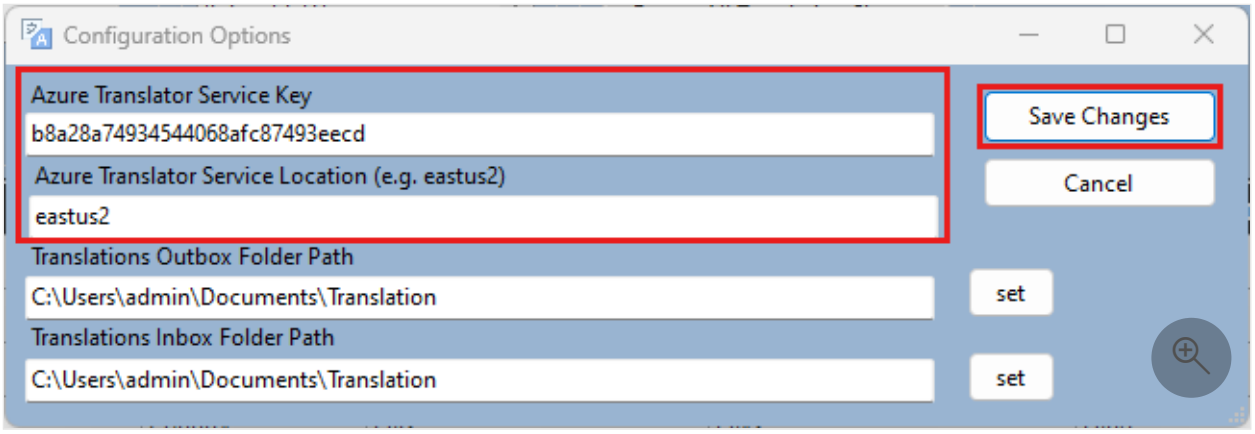
Translations Builder generates machine translations by using Azure AI Translator. This product makes it possible to automate enumerating through data model objects to translate its names from the default language to translations for secondary languages.

To test the support in Translations Builder for generating machine translations, you need a key for an instance of the Azure Translator Service. For more information about

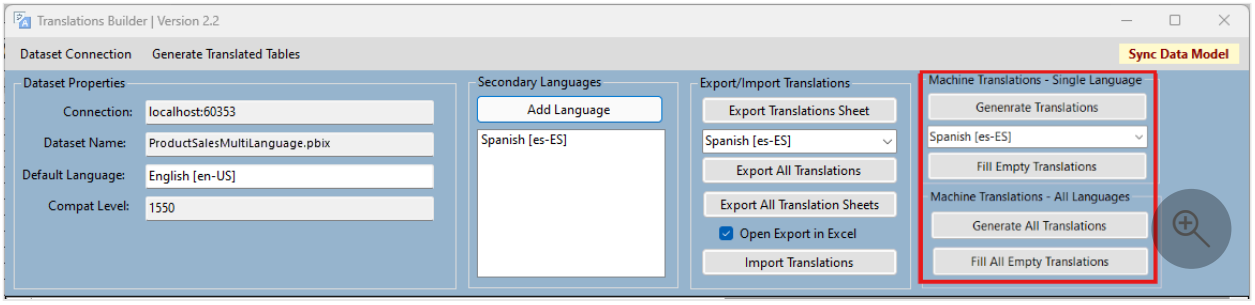
obtaining a key, see What is Azure AI Translator?

> **Note**
>
> If you haven't installed Translations Builder yet, see Create multiple-language reports with Translations Builder.

Translations Builder provides a **Configuration Options** dialog box where you can configure the key and location to access the Azure Translator Service.



After you configure an **Azure Translator Service Key**, Translations Builder displays other command buttons. These buttons generate translations for a single language at a time or for all languages at once. There are also commands to generate machine translations only for the translations that are currently empty.



Related content

- Add support for multiple-language page navigation

Feedback

Was this page helpful? Yes No

Add support for multiple-language page navigation

Article • 05/12/2025

You can't display page tabs to the user in a multiple-language report because page tabs in a Power BI report don't support localization. For localization, you must provide some other means for users to navigate from page to page.

You can use a design technique where you add a navigation menu that uses buttons. When the user selects a button, the button applies a bookmark to navigate to another page. This section describes the process of building a navigation menu that supports localization using measures from the **Localized Labels** table.

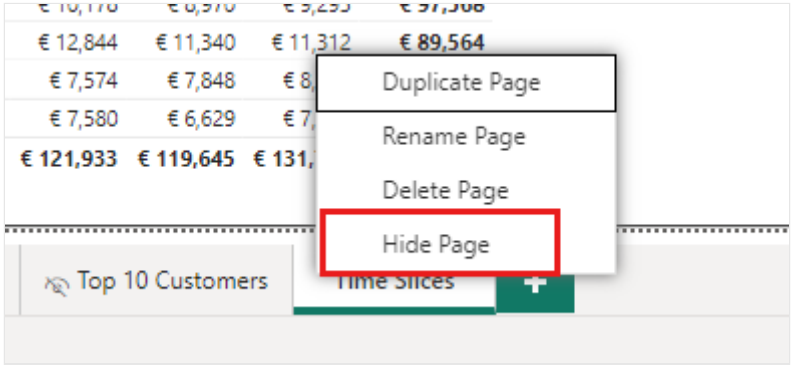
This article uses the multiple-language demo project and Power BI Desktop. You don't need a Power BI license to start developing in Power BI Desktop. If you don't already have Power BI Desktop installed, see Get Power BI Desktop.

Hide tabs

When you hide all but one of the tabs in your report, none of the tabs appear in the published report. The report opens to the page of the unhidden tab. Even that tab isn't displayed.

Start by hiding all but one of the tabs.

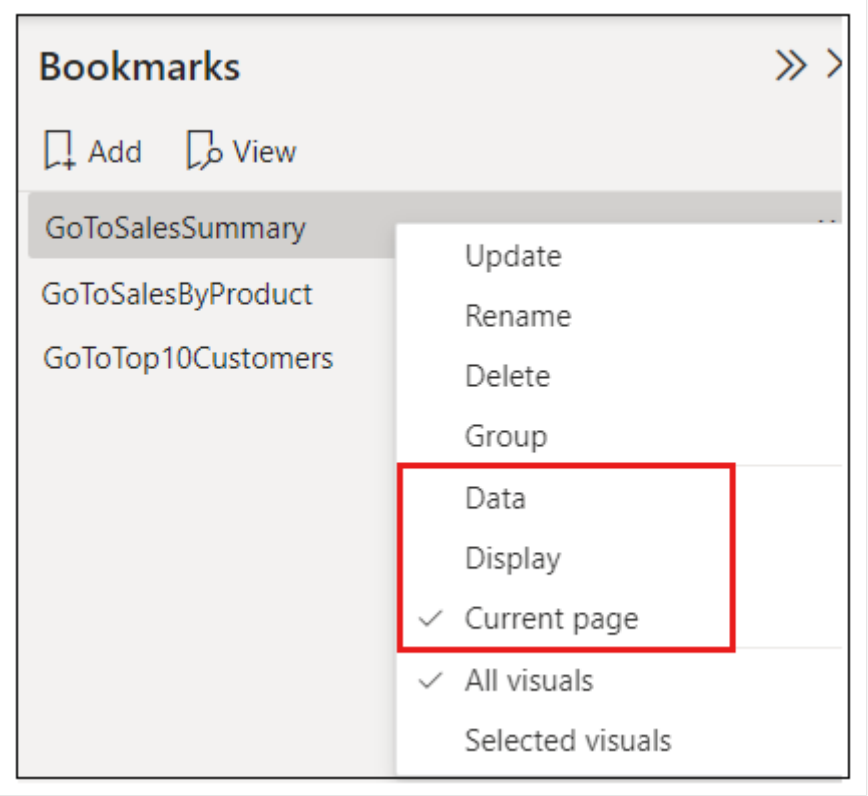
1. Open the report in Power BI Desktop.
2. For each tab that you hide, right-click and select **Hide Page** from the context menu.



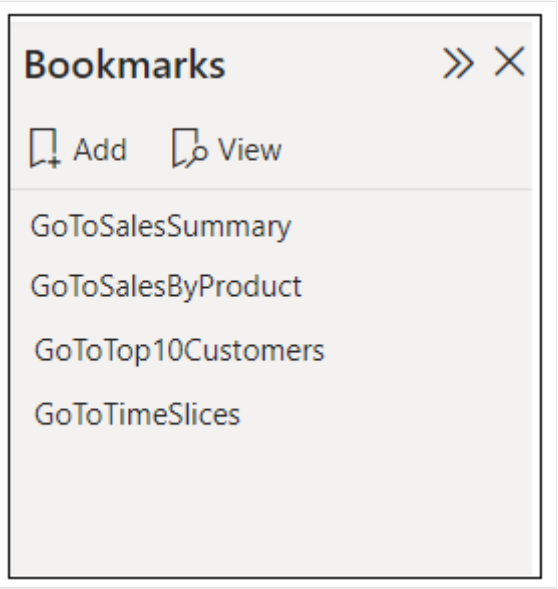
Create bookmarks

Each button uses a bookmark to take the reader to a page. For more information on bookmarks, see Create page and bookmark navigators.

1. From the **View** ribbon, select **Bookmarks** to display the **Bookmarks** pane.
2. In the **Bookmarks** pane, create a set of bookmarks. Each bookmark navigates to a specific page.
    a. Select a tab, starting with **Sales Summary**, which serves as the landing page.
    b. In **Bookmarks**, select **Add**.
    c. Right-click the new bookmark and select **Rename**. Enter a bookmark name, such as **GoToSalesSummary**.
    d. Right-click the bookmark name and disable **Data** and **Display**. Enable **Current Page** behavior.



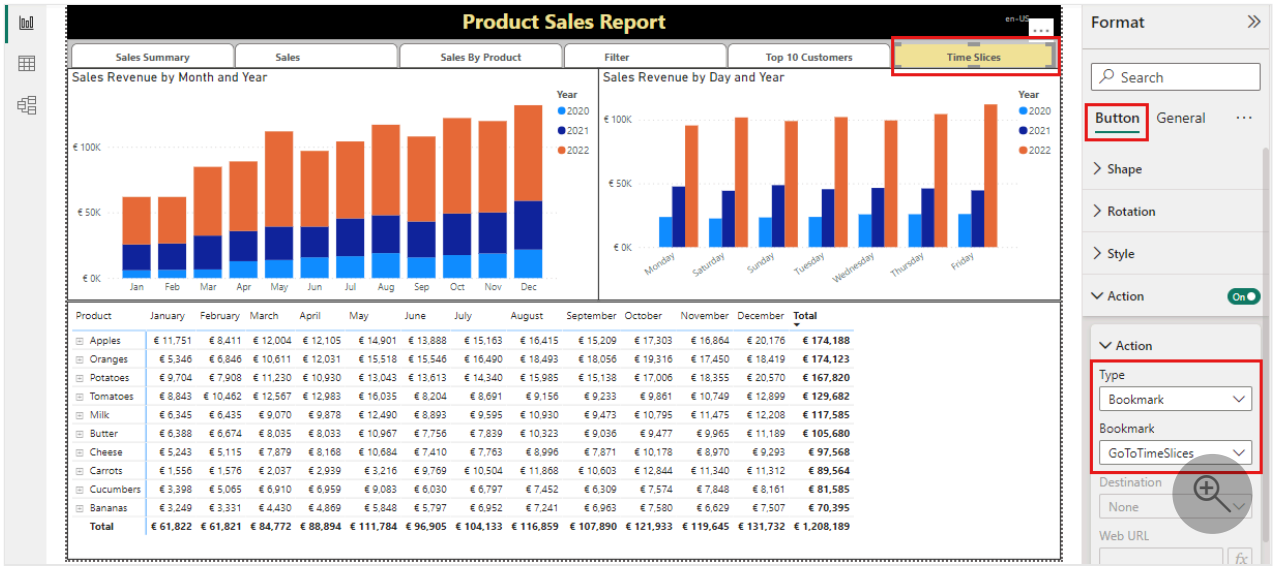
    e. Repeat these steps for each of the hidden tabs. The **Bookmark** pane has the following bookmarks:



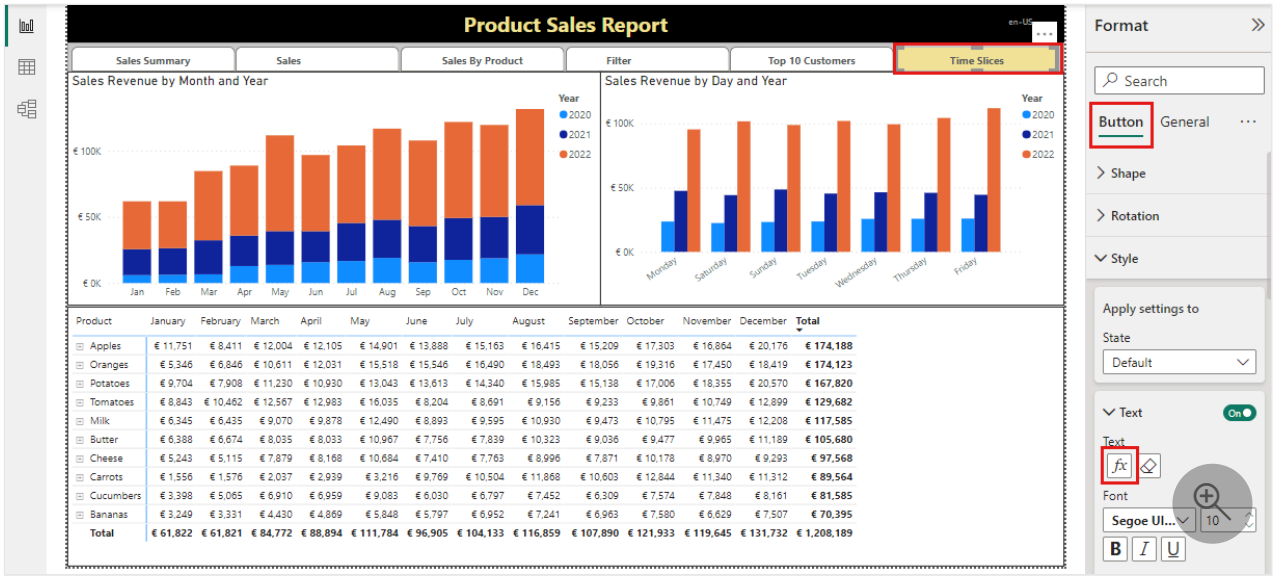
Configure buttons

The multiple-language demo project contains buttons for navigation. To learn more about adding buttons, see Create buttons in Power BI reports.

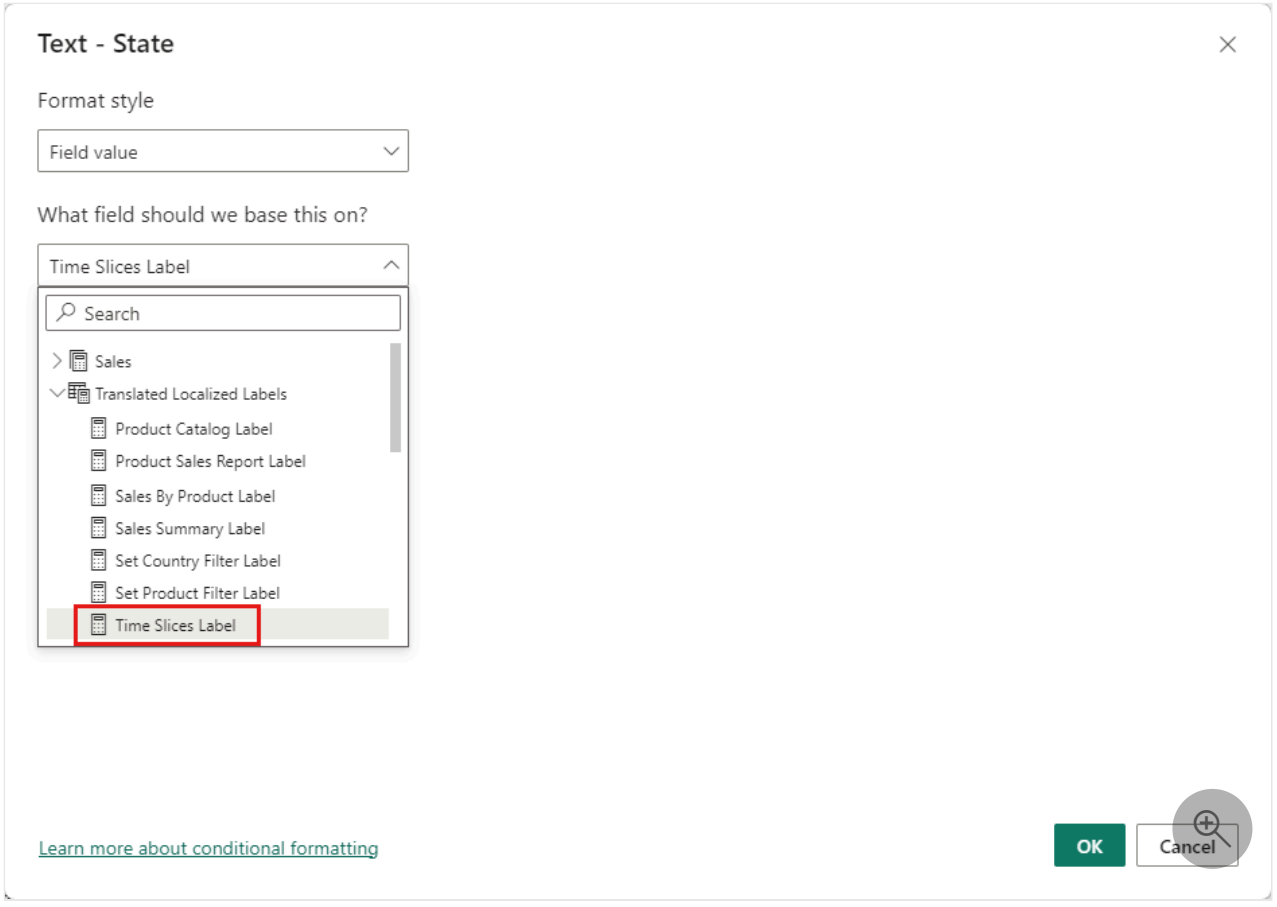
1. Select a button at the top of the report, starting with **Sales Summary**.
2. Under **Format**, select **Button** > **Action**. Set **Action** to **On**.
3. Under **Action**, set **Type** to **Bookmark** and **Bookmark** to the relevant bookmark, starting with **GoToSalesSummary**.
4. In the same way, configure each button in the navigation menu to apply a bookmark to navigate to a specific page.



5. For each button, select **Button** > **Style** > **Text** and then select the function button.



6. In the **Text - State** dialog box, from the **Translated Localized Labels** table, select the entry that corresponds to that button. For instance, **Time Slices Label** for **Time Slices**.



7. Select **OK** to save your selection.

The report now has no visible tabs when you publish it to the Power BI service. The report opens to the **Sales Summary** page. Readers can move from page to page by using the buttons, which are localized by using the **Translated Localized Labels** table.

> **Note**

With this technique, the screen reader treats each button as a separate entity and doesn't group them together as it would with tabs.

Related content

- Guidance for Power BI

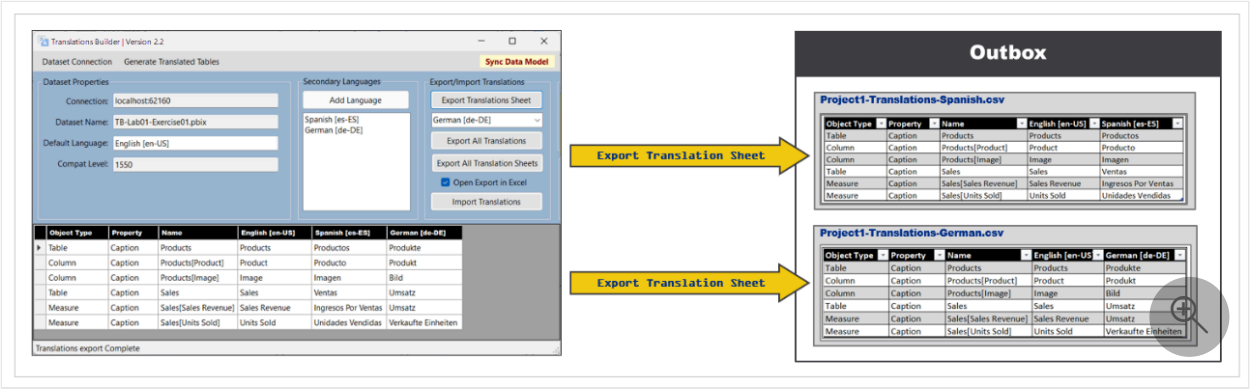
Enable workflows for human translation in Power BI reports

Article • 01/15/2024

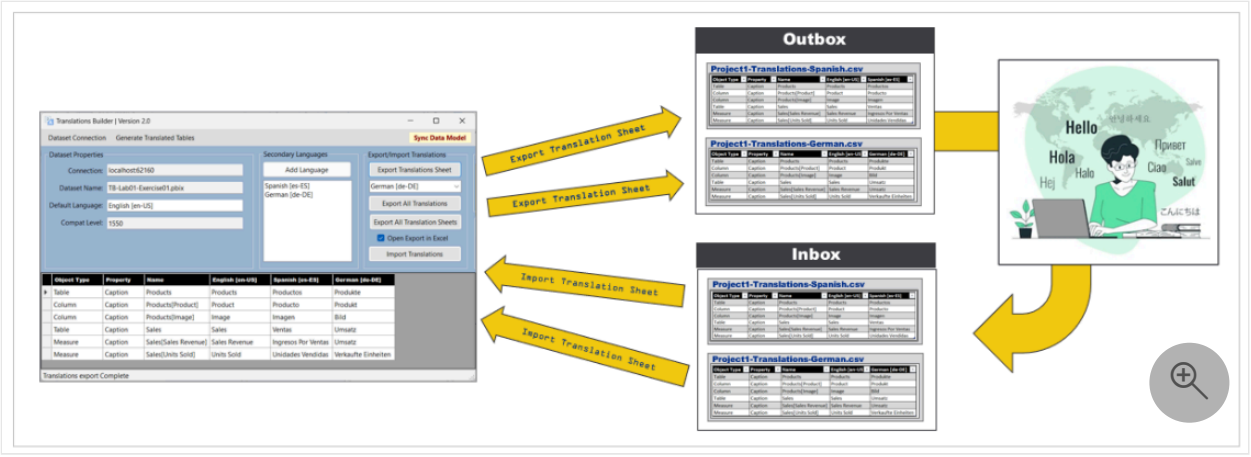
When you create multiple-language reports for Power BI, you can work quickly and efficiently by using Translations Builder and by generating machine translations. However, machine generated translations alone are inadequate for many production needs. You need to integrate other people acting as translators into a human workflow process.

Translations Builder uses a *translation sheet*, which is a .csv file that you export to send to a translator. The human acting as a translator updates the translation sheet and then returns it to you. You then import the revised sheet to integrate the changes into the current data model.

Translations Builder generates a file for a selected language using a special naming format, for example, *PbixProjectName-Translations-Spanish.csv*. The name includes the data model name and the language for translation. Translations Builder saves the generated translation sheet to a folder known as the *Outbox*.



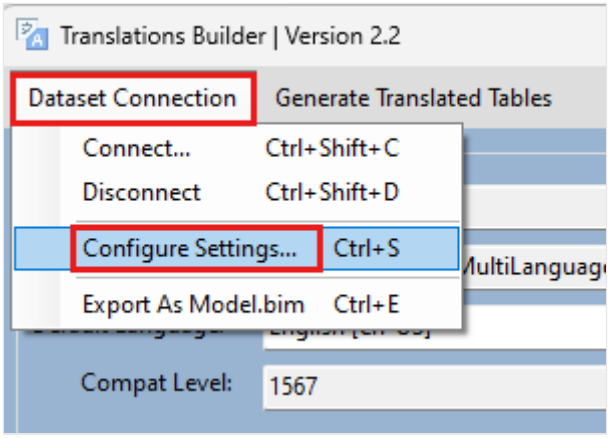
Human translators can make edits to a translation sheet using Microsoft Excel. When you receive an updated translation sheet from a translator, copy it to the *Inbox* folder. From there, you can import it to integrate those updated translations back into the data model for the current project.



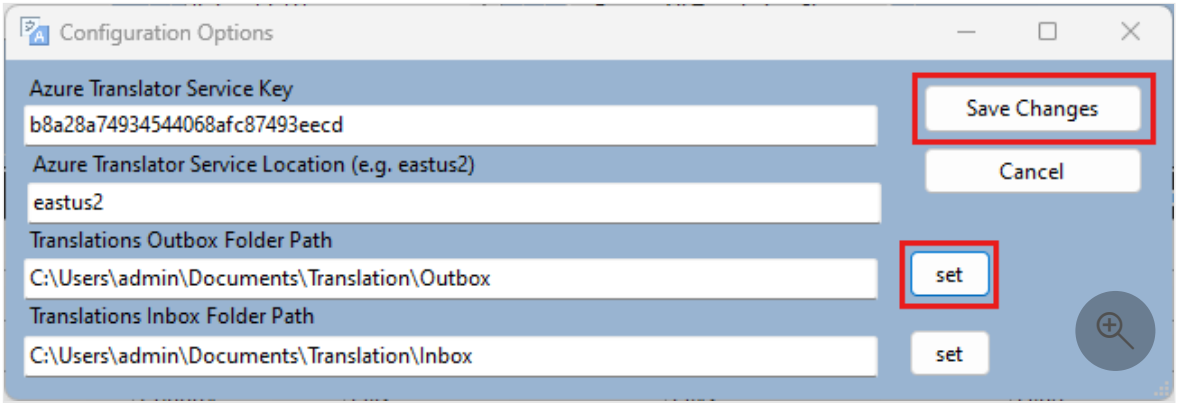
Configure import and export folders

By default, folder paths for *Outbox* and *Inbox* in Translations Builder are the *Documents* folder of the current user. To configure settings used as targets for export and import operations, follow these steps.

1. From the **Dataset Connection** menu, select **Configure Settings** to display the **Configuration Options** dialog box.

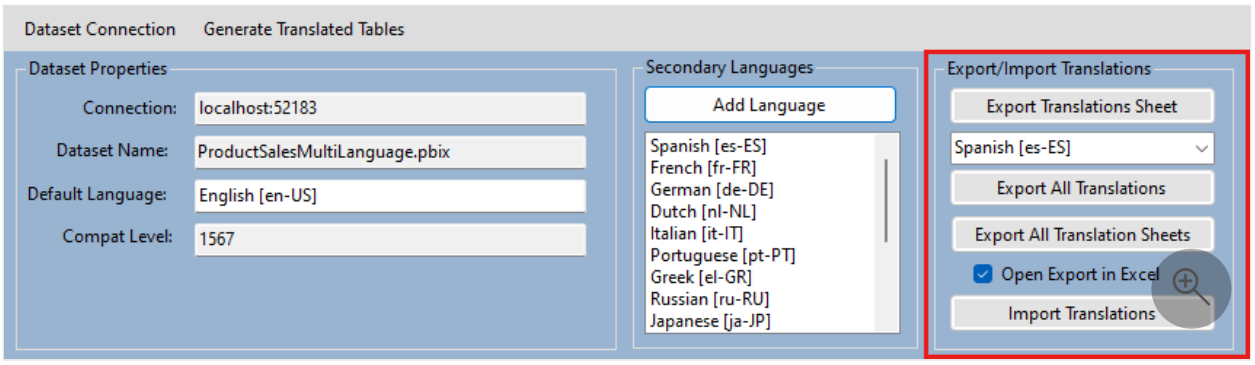


2. Select the **set** buttons to update the settings for **Translations Outbox Folder Path** and **Translations Outbox Folder Path**.



3. After you have configured paths, select **Save Changes**.

After you configure the folder paths for *Outbox* and *Inbox*, you can begin to export and import translation sheets by using the **Export/Import Translations** options.



Related content

- Export translation sheets in Translations Builder

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Export translation sheets in Translations Builder

Article • 08/04/2023

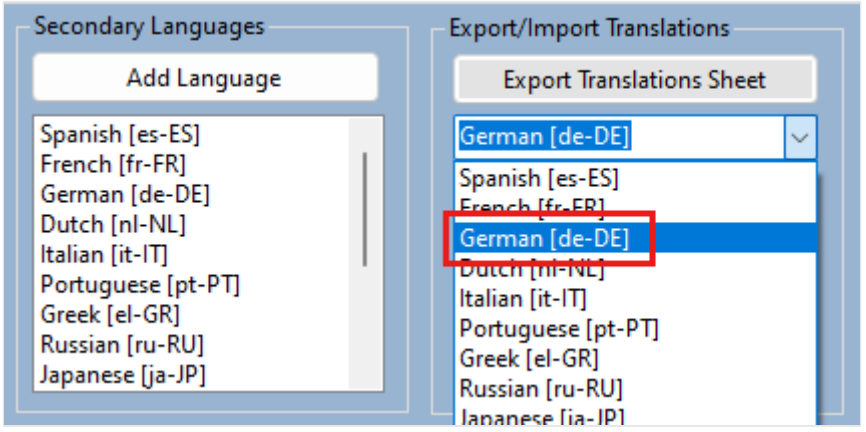
When you use Translations Builder with external translators, you need to export a *translation sheet* that contains the default language and empty cells or machine generated translations. Translators update the .csv file and return it to you.

You can export the following translations sheets:

- A translation sheet for a single language
- Translation sheets for all supported languages
- A translation sheet that contains all supported languages

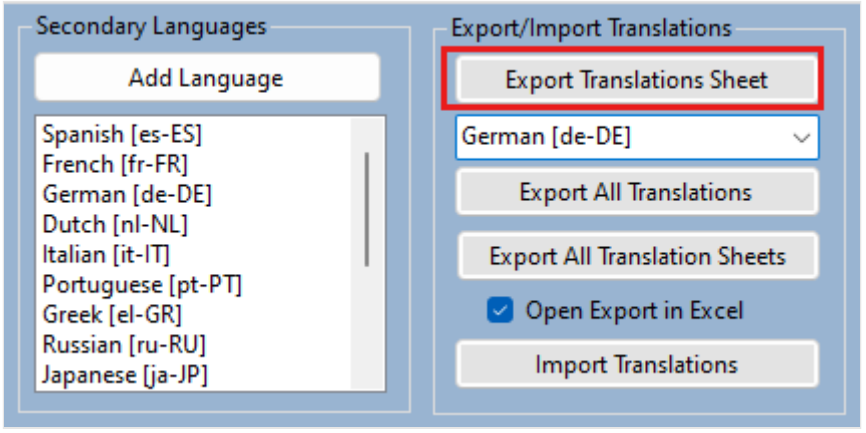
Export translation sheet for a single language

1. In Translations Builder, under **Export/Import Translations**, select a language such as **German [de-DE]**.

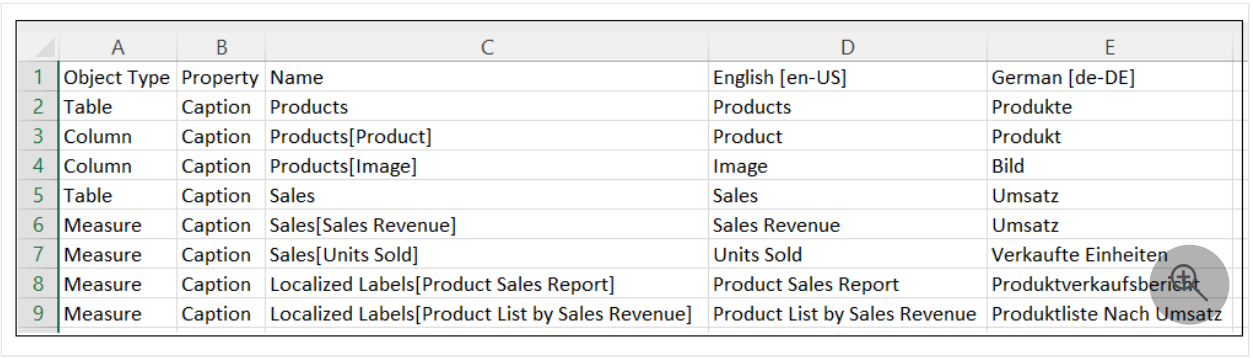


2. Select **Export Translations Sheet** to generate a translation sheet for that language.

   You can select **Open Export in Excel** to view the exported file immediately.



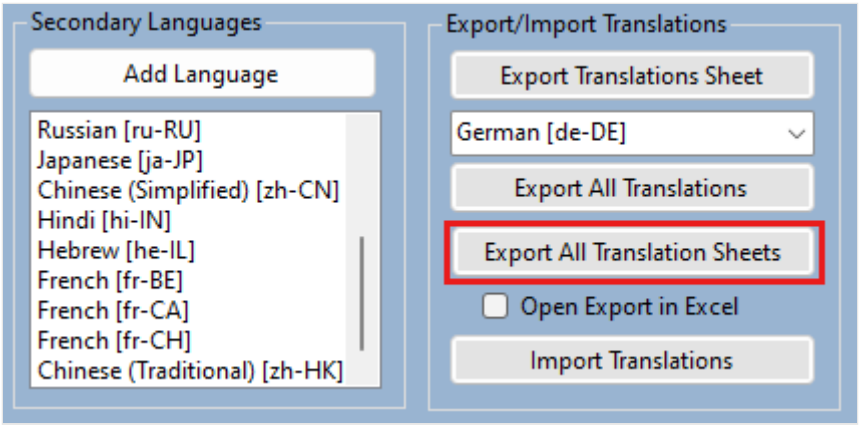
The result of the export operation is a .csv file in the *Outbox* directory. If you selected **Open Export in Excel**, you also see the result in Excel.



	A	B	C	D	E
1	Object Type	Property	Name	English [en-US]	German [de-DE]
2	Table	Caption	Products	Products	Produkte
3	Column	Caption	Products[Product]	Product	Produkt
4	Column	Caption	Products[Image]	Image	Bild
5	Table	Caption	Sales	Sales	Umsatz
6	Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Umsatz
7	Measure	Caption	Sales[Units Sold]	Units Sold	Verkaufte Einheiten
8	Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Produktverkaufsbericht
9	Measure	Caption	Localized Labels[Product List by Sales Revenue]	Product List by Sales Revenue	Produktliste Nach Umsatz

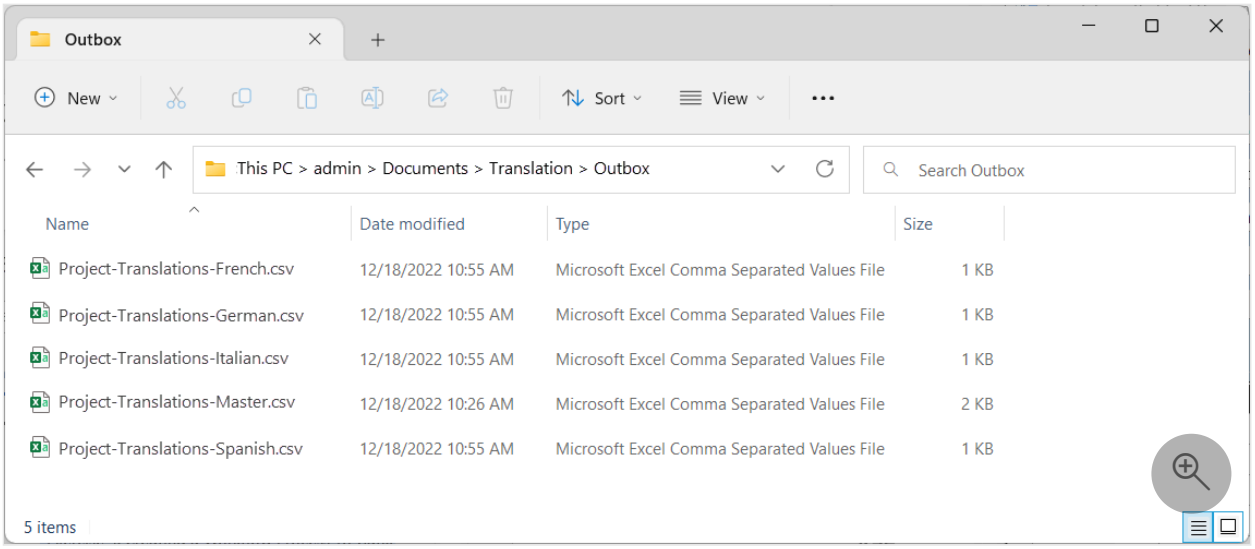
Export translation sheets for all languages

You can export translations sheets for all the languages supported for your project at once. Under **Export/Import Translations**, select **Export All Translation Sheets**.



> **Tip**
>
> Don't select **Open Export in Excel**. That option opens all of the files in Excel.

Translations Builder generates the complete set of translation sheets to be sent to translators.



Export all translations

You can export a single translation sheet that contains all the secondary languages and translations that have been added to the current project. Under **Export/Import Translations**, select **Export All Translations**.

Translations Builder generates a .csv file for the full translation sheet named *PbixProjectName-Translations-Master.csv*. When you open the translations sheet in Excel, you can see all secondary language columns and all translations. You can think of this translation sheet as a backup of all translations on a project-wide basis.

	A	B	C	D	E	F	G	H
1	Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]	Italian [it-IT]
2	Table	Caption	Products	Products	Productos	Produits	Produkte	Prodotti
3	Column	Caption	Products[Product]	Product	Producto	Produit	Produkt	Prodotto
4	Column	Caption	Products[Image]	Image	Imagen	Image	Bild	Immagine
5	Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz	Vendite
6	Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz	Ricavi Delle Vendite
7	Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkaufte Einheiten	Unità Vendute
8	Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos	Rapport Sur Les Ventes De Produits	Produktverkaufsbericht	Report Sulle Vendite Dei Prodotti
9	Measure	Caption	Localized Labels[Product List by Sales Revenue]	Product List by Sales Revenue	Lista De Productos Por Ingresos Por Ventas	Liste De Produits Par Chiffre D'Affaires	Produktliste Nach Umsatz	Elenco Dei Prodotti Per Fatturato Delle Vendite

Related content

- Import translation sheets in Translations Builder

Feedback

Was this page helpful? Yes No

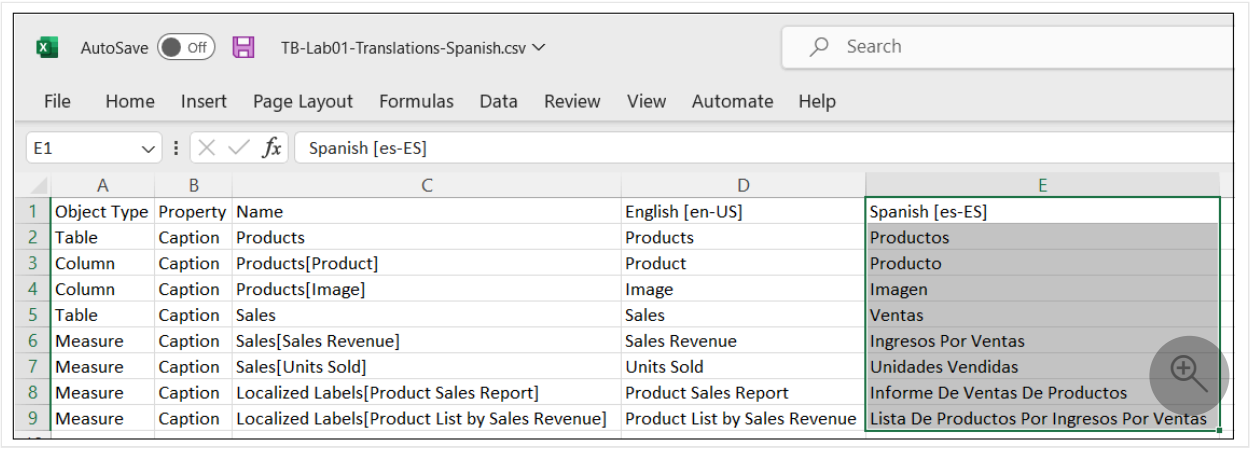
Provide product feedback | Ask the community

Import translation sheets in Translations Builder

Article • 08/04/2023

When you use Translations Builder with external translators, you need to export a *translation sheet* that contains the default language and empty cells or machine generated translations. Translators update the .csv file and return it to you.

Suppose you generate a translation sheet to send to a translator. When opened in Excel, this translation sheet looks like the following screenshot.



	A	B	C	D	E
1	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
2	Table	Caption	Products	Products	Productos
3	Column	Caption	Products[Product]	Product	Producto
4	Column	Caption	Products[Image]	Image	Imagen
5	Table	Caption	Sales	Sales	Ventas
6	Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas
7	Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas
8	Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos
9	Measure	Caption	Localized Labels[Product List by Sales Revenue]	Product List by Sales Revenue	Lista De Productos Por Ingresos Por Ventas

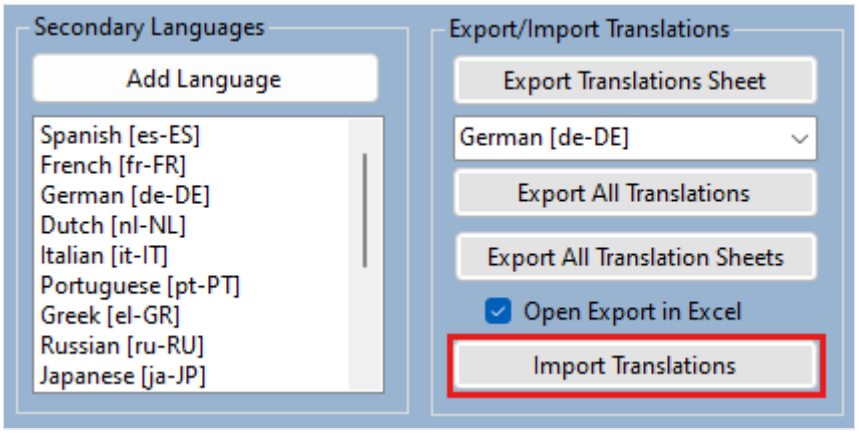
The job of the translator is to review all translations in the relevant column and to make updates where appropriate. From the perspective of the translator, the top row with column headers and the first four columns should be treated as read-only values.

Import translation sheet

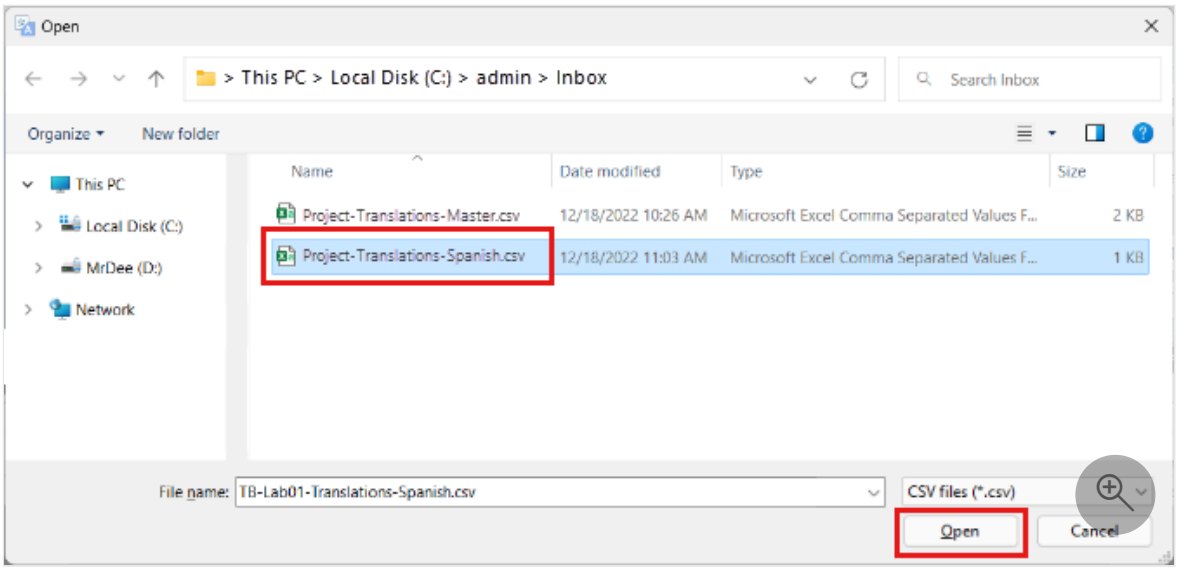
When you receive the translation sheet back from the translator with updates, follow these steps.

1. If you opened the translation sheet in Excel, close it before proceeding.

2. In Translations Builder, under **Export/Import Translations**, select **Import Translations**.



3. In the **Open** dialog box, select the translation sheet file and select **Open**.



The Spanish translation updates now appear in the translation grid.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]
Table	Caption	Products	Products	Productos	Produits
Column	Caption	Products[Product]	Product	Producto	Produit
Column	Caption	Products[Image]	Image	Imagen	Image
Table	Caption	Sales	Sales	Ventas	Ventes
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues
Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos	Rapport Sur Les Ventes De Produits
Measure	Caption	Localized Labels[Product List by Sales Revenue]	Product List by Sales Revenue	Lista De Productos Por Ingresos Por Ventas	Liste De Produits Par Chiffre D'Affaires

Import master translation sheet

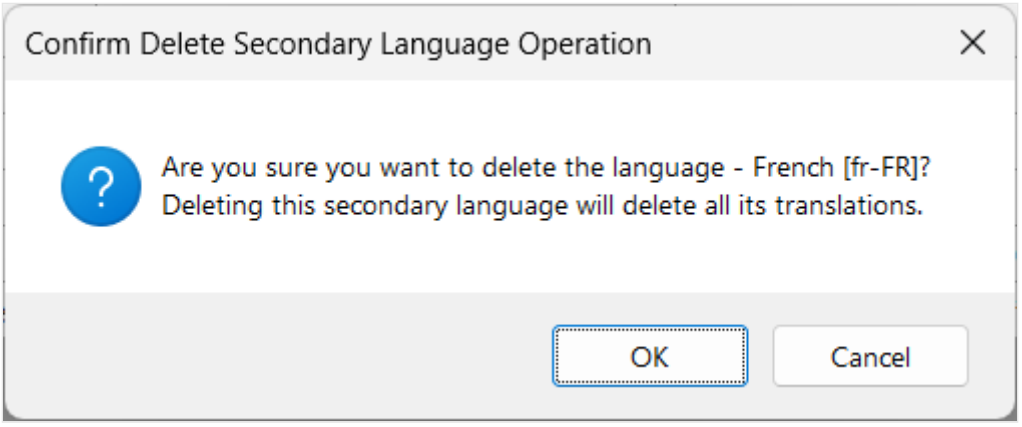
The usual workflow is to import updated translation sheets that only contain translations for a single language. However, you can also import a master translation sheet that has multiple columns for secondary languages. This approach provides a way to back up and restore the work you have done with translations on a project-wide basis. You can also use this approach to reuse translations across multiple projects.

Here's a simple example. After you generate the current master translation sheet for a project, imagine you delete French as a language from the project by right-clicking the

French [fr-FR] column header and selecting **Delete This Language From Data Model**.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]			Italian [it-IT]
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues			Unità Vendute
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires			Ricavi Delle Vendite
Measure	Caption	Sales[Ranking]	Ranking	Clasificación	Classement			Graduatoria
Measure	Caption	Sales[Rank]	Rank	Rango	Rang			Rango
Measure	Caption	Sales[Orders]	Orders	Órdenes	Ordres	Aufträge	Orders	Ordini
Measure	Caption	Sales[Last Sale]	Last Sale	Última Venta	Dernière Vente	Letzter Verkauf	Laatste Verkoop	Ultima Vendita
Measure	Caption	Sales[First Sale]	First Sale	Primera Venta	Première Vente	Erster Verkauf	Eerste Verkoop	Prima Vendita
Measure	Caption	Sales[Customer Count]	Customers	Clientela	Clientèle	Kundschaft	Klanten	Clientela
Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz	Verkoop	Vendite

When you attempt to delete the column for a language, Translations Builder prompts you to verify.



You can select **OK** to continue. After you confirm the delete operation, the column for French has been removed from the translations grid. Behind the scenes, Translations Builder has also deleted all the French translations from the project.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	German [de-DE]	Dutch [nl-NL]	Italian [it-IT]	Portuguese [pt-PT]
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Verkaufte Einheiten	Verkochte Eenheden	Unità Vendute	Unidades Vendidas
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Umsatz	Omzet	Ricavi Delle Vendite	Receitas De Vendas
Measure	Caption	Sales[Ranking]	Ranking	Clasificación	Rangordnung	Ranking	Graduatoria	Ranking
Measure	Caption	Sales[Rank]	Rank	Rango	Rang	Rang	Rango	Classificação
Measure	Caption	Sales[Orders]	Orders	Órdenes	Aufträge	Orders	Ordini	Encomendas
Measure	Caption	Sales[Last Sale]	Last Sale	Última Venta	Letzter Verkauf	Laatste Verkoop	Ultima Vendita	Última Venda
Measure	Caption	Sales[First Sale]	First Sale	Primera Venta	Erster Verkauf	Eerste Verkoop	Prima Vendita	Primeira Venda
Measure	Caption	Sales[Customer Count]	Customers	Clientela	Kundschaft	Klanten	Clientela	Clientes
Table	Caption	Sales	Sales	Ventas	Umsatz	Verkoop	Vendite	Vendas

Suppose that you deleted this language by mistake. You can use a previously generated a master translation sheet to restore the deleted language. If you import the translation sheet, the **French [fr-FR]** column reappears as the last column.

	Chinese (Traditional) [zh-HK]	French [fr-FR]
	已售單位	Unités Vendues
	銷售收入	Chiffre D'Affaires
	排名	Classement
	排	Rang
	訂單	Ordres
	最後銷售	Dernière Vente
	首次銷售	Première Vente
	客戶數量	Clientèle
	銷售	Ventes

Related content

- Manage data model translations at the enterprise level

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Manage data model translations at the enterprise level

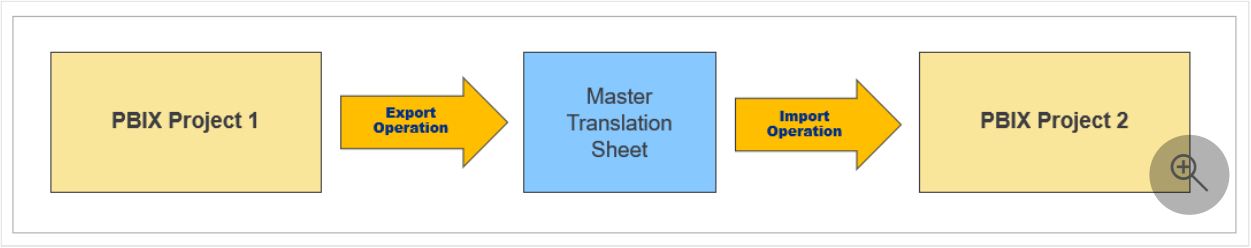
Article • 01/15/2024

You can use a master translations sheet as a project backup. Translations Builder adds a secondary language along with its translations to a .pbix project if it's found in the translation sheet but not in the target project. For more information, see Import translation sheets in Translations Builder.

You can also create an enterprise-level master translation sheet to import when you create new .pbix projects.

Import translations

Imagine you have two .pbix projects that have a similar data model in terms of the tables, columns, and measures. In the first project, you have already added metadata translations for all the unhidden data model objects. In the second project, you haven't yet started to add secondary languages or translations. You can export the master translation sheet from the first project and import it into the second project.

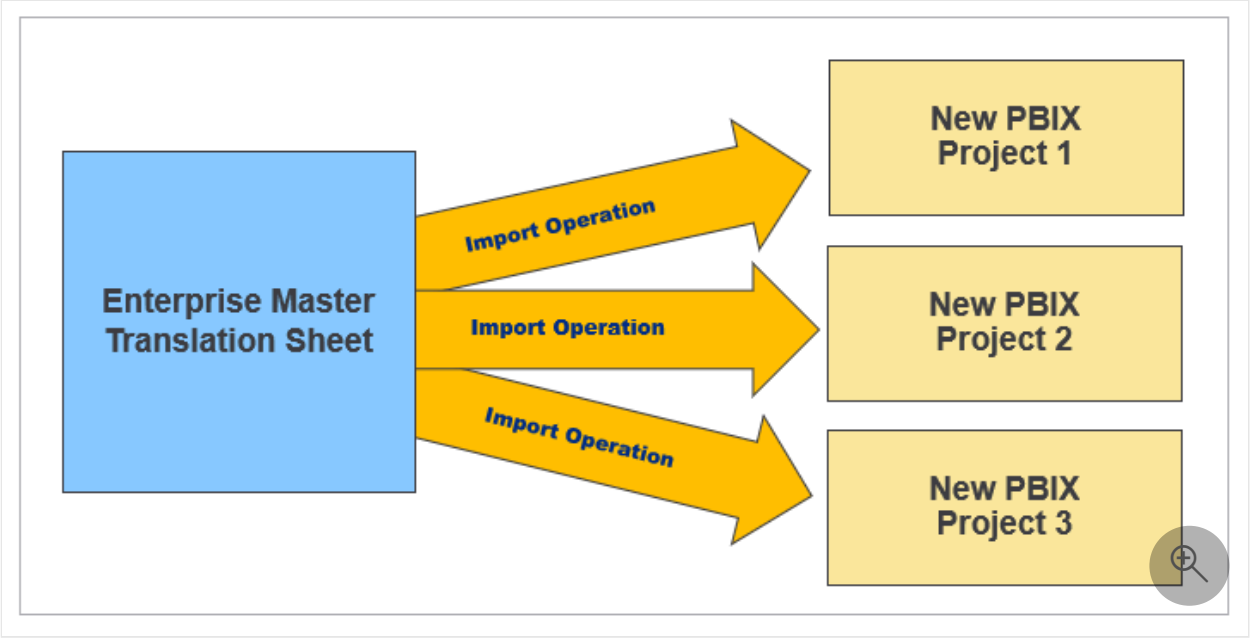


The **Import Translations** command starts by determining whether there are any secondary languages in the translation sheet that aren't in the target .pbix project. It adds any secondary languages not already present in the target project. After that, the command moves down the translation sheet row by row.

For each row, Translations Builder determines whether a data model object in the .csv file matches a data model object of the same name in the .pbix project. When it finds a match, the command copies all the translations for that data model object into the .pbix project. If it finds no match, the command ignores that row and continues to the next row.

The **Import Translations** command provides special treatment for report labels that have been added to the **Localized Labels** table. If you import a translation sheet with

one or more localized report labels into a new .pbix project, the command creates the **Localized Labels** table.



Because the **Import Translations** command creates the **Localized Labels** table and copies report labels into a target .pbix project, it can be the foundation for maintaining an enterprise-level master translation sheet. Use a set of localized report labels across multiple .pbix projects. Each time you create a new .pbix project, you can import the enterprise-level translation sheet to add the generalized set of localized report labels.

Related content

- Guidance for Power BI

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Implement a data translation strategy

Article • 08/09/2023

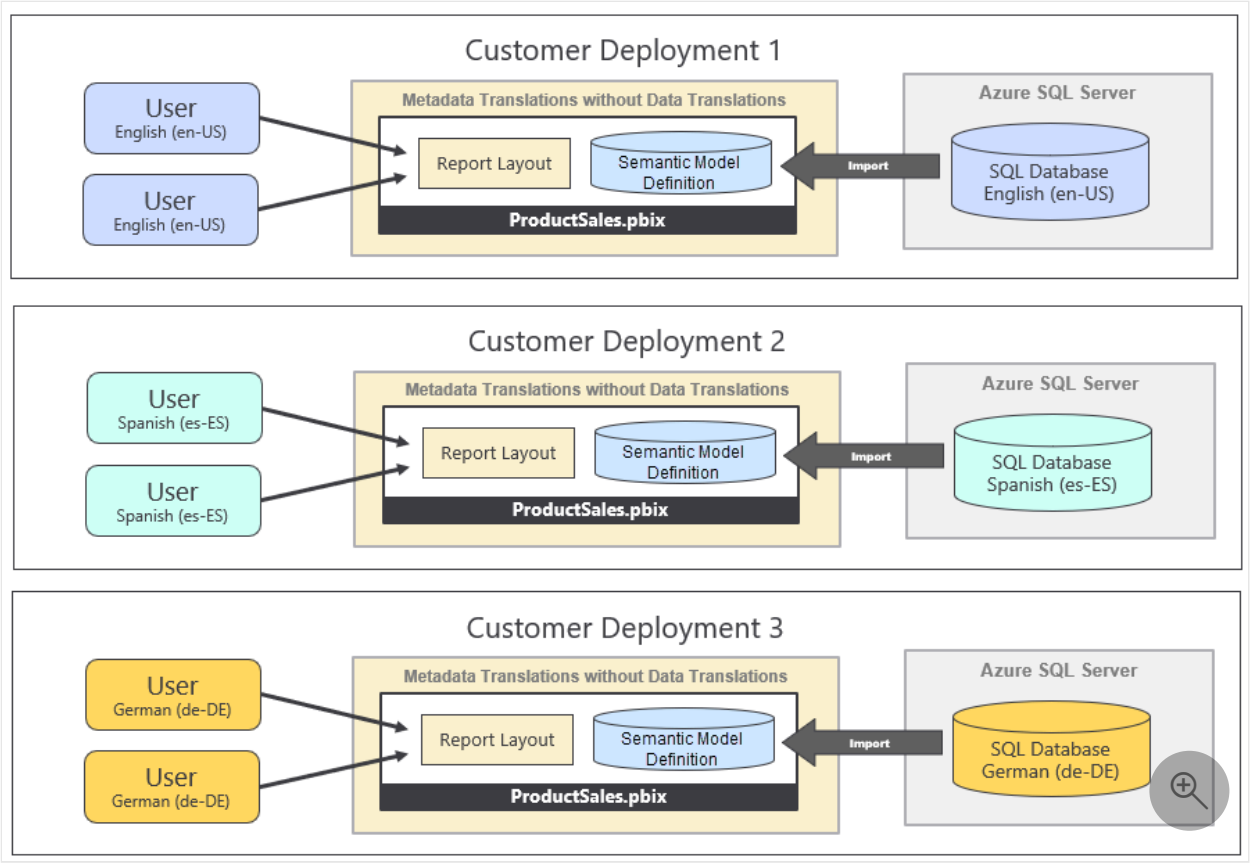
All multiple-language reports require metadata translation and report label translation, but not necessarily data translation. To determine whether your project requires data translation, think through the use cases that you plan to support. Using data translation involves planning and effort. You might decide not to support data translation unless it's a hard requirement for your project.

Implementing data translation is different from implementing metadata translation or report label translation. Power BI doesn't offer any localization features to assist you with data translation. Instead, you must implement a data translation strategy. Such a strategy involves extending the underlying data source with extra columns to track translations for text in rows of data, such as the names of products and categories.

Determine whether your solution requires data translation

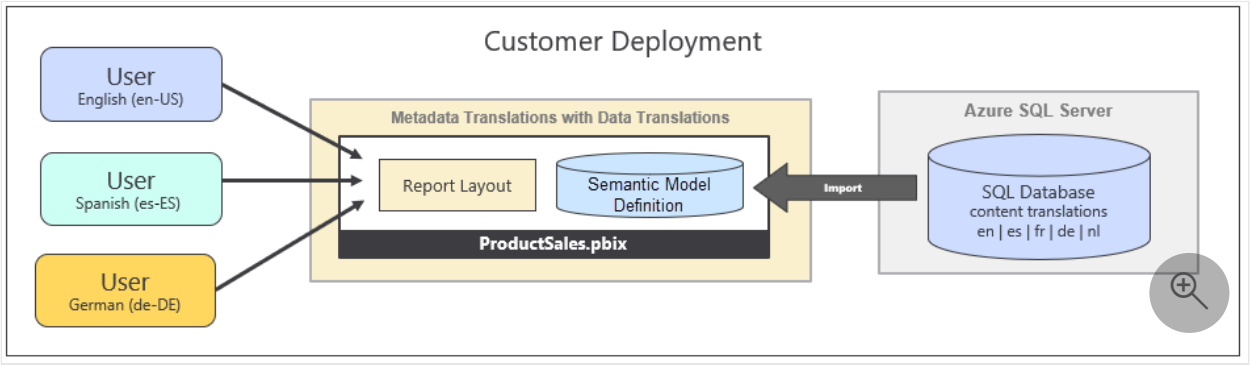
To determine whether you need to implement data translation, start by thinking about how to deploy your reporting solution. Think about the use case for its intended audience. That exercise leads to a key question: *Do you have people who use different languages looking at the same database instance?*

Suppose you're developing a report template for a software as a service (SaaS) application with a well-known database schema. Some customers maintain their database instance in English while others maintain their database instances in other languages, such as Spanish or German. There's no need to implement data translations in this use case because the data from any database instance because users view the data each in a single language.



Each customer deployment uses a single language for its database and all its users. Both metadata translations and report label translations must be implemented in this use case. You deploy a single version of the .pbix file across all customer deployments. However, there's no need to implement data translations when no database instance ever needs to be viewed in multiple languages.

A different use case introduces the requirement of data translations. The example .pbix project file uses a single database instance that contains sales performance data across several European countries/regions. This solution must display its reports in different languages with data from a single database instance.



If you have people that use different languages and locales to interact with the same database instance, you still need to address other considerations.

- Examine the text-based columns that are candidates for translation. Determine how hard translating those text values is. Columns with short text values, like

product names and product categories, are good candidates for data translations. Columns that hold longer text values, such as product descriptions, require more effort to generate high quality translations.

- Consider the number of distinct values that require translation. You can easily translate product names in a database that holds 100 products. You can probably translate product names when the number gets up to 1000. What happens if the number of translated values reaches 10,000 or 100,000? If you can't rely on machine-generate translations, your translation team might have trouble scaling up to handle that volume of human translations.

- Consider whether there's on-going maintenance. Every time someone adds a new record to the underlying database, there's the potential to introduce new text values that require translation. This consideration doesn't apply to metadata translation or report label translation. In those situations, you create a finite number of translations and then your work is done. Metadata translation and report label translation don't require on-going maintenance as long as the underlying semantic model schema and the report layout remain the same.

There are several factors that go into deciding whether to use data translation. You must decide whether it's worth the time and effort required to implement data translation properly. You might decide that implementing metadata translations and report label translations goes far enough. If your primary goal is to make your reporting solution compliant with laws or regulations, you might also find that implementing data translations isn't a requirement.

Related content

- Extend the data source schema to support data translations

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Extend the data source schema to support data translations

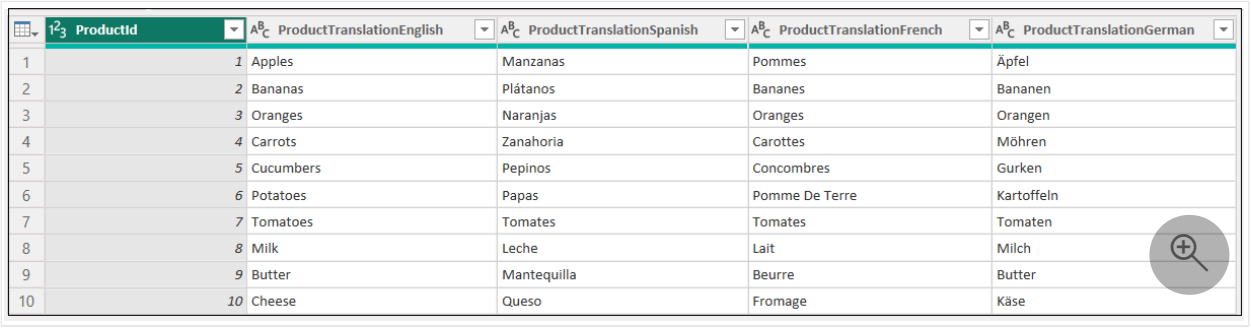
Article • 08/09/2023

There are multiple ways to implement data translations in Power BI. Some data translation strategies are better than others. Whatever approach you choose, make sure that it scales in terms of performance. You should also ensure your strategy scales in terms of the overhead required to add support for new secondary languages as part of the on-going maintenance.

The current series of articles describes a strategy for implementing data translations made possible by the Power BI Desktop feature called *field parameters*.

Modify the data source

Start by modifying the underlying data source. For example, the **Products** table can be extended with extra columns with translated product names to support data translations. In this case, the **Products** table has been extended with a separate column with product name translations in English, Spanish, French, and German.



	ProductId	ProductTranslationEnglish	ProductTranslationSpanish	ProductTranslationFrench	ProductTranslationGerman
1	1	Apples	Manzanas	Pommes	Äpfel
2	2	Bananas	Plátanos	Bananes	Bananen
3	3	Oranges	Naranjas	Oranges	Orangen
4	4	Carrots	Zanahoria	Carottes	Möhren
5	5	Cucumbers	Pepinos	Concombres	Gurken
6	6	Potatoes	Papas	Pomme De Terre	Kartoffeln
7	7	Tomatoes	Tomates	Tomates	Tomaten
8	8	Milk	Leche	Lait	Milch
9	9	Butter	Mantequilla	Beurre	Butter
10	10	Cheese	Queso	Fromage	Käse

The design approach shown here uses a three-part naming convention for table column names used to hold data translations. A name consists of the following parts:

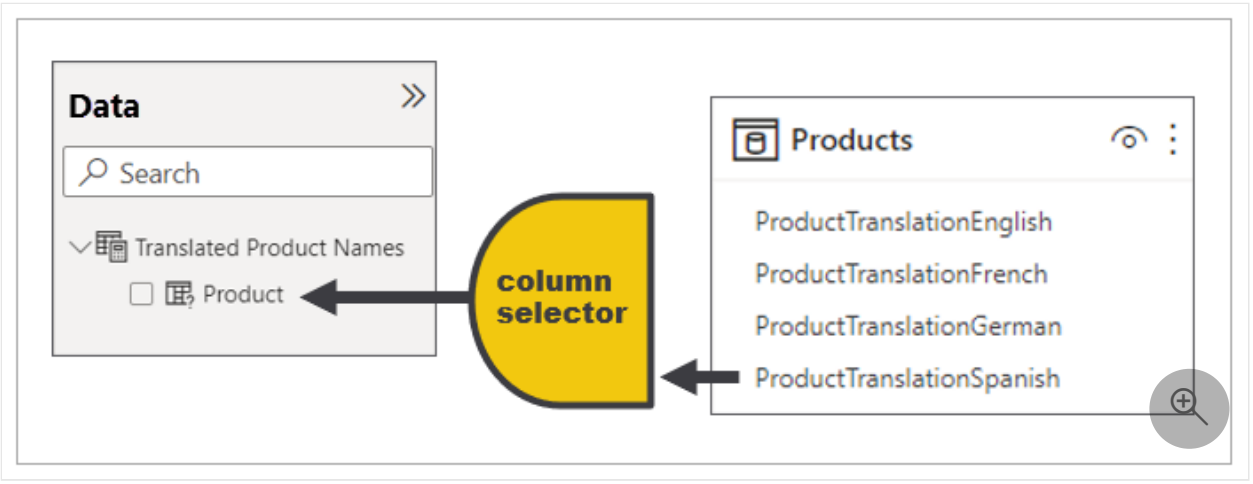
- The entity name, for instance, **Product**
- The word **Translation**
- The language name, for instance, **Spanish**

For example, the column that contains product names translated into Spanish is **ProductTranslationSpanish**. Using this three-part naming convention isn't required for implementing data translation, but Translations Builder gives these columns special treatment.

Understand field parameters

A *field parameter* is a table in which each row represents a field and where each these fields must be defined as either a column or a measure. In one sense, a field parameter is just a predefined set of fields. Given that rows in a table represent these fields, the set of fields of a field parameter supports filtering. You can think of a field parameter as a filterable set of fields.

When you create a field parameter, you can populate the fields collection using either measures or columns.



When you use field parameters to implement data translations, use columns instead of measures. The primary role that field parameters play in implementing data translations is providing a single, unified field use in report authoring that can be dynamically switched between source columns.

Related content

- Implement data translation using field parameters

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Implement data translation using field parameters

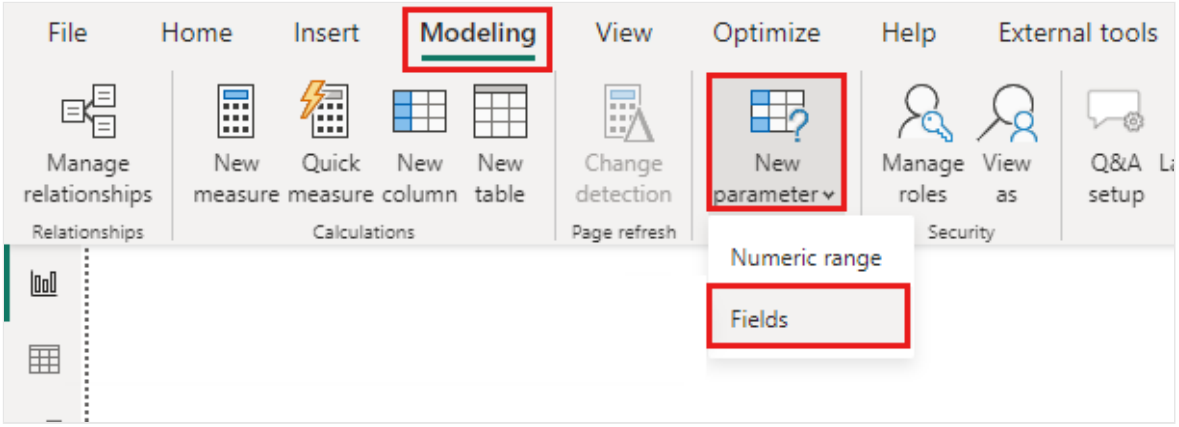
Article • 08/09/2023

This article shows you how to implement data translation by using a field parameter. The process has the following steps:

- Create a field parameter
- Use a slicer and data table
- Edit translated names
- Add a language ID column

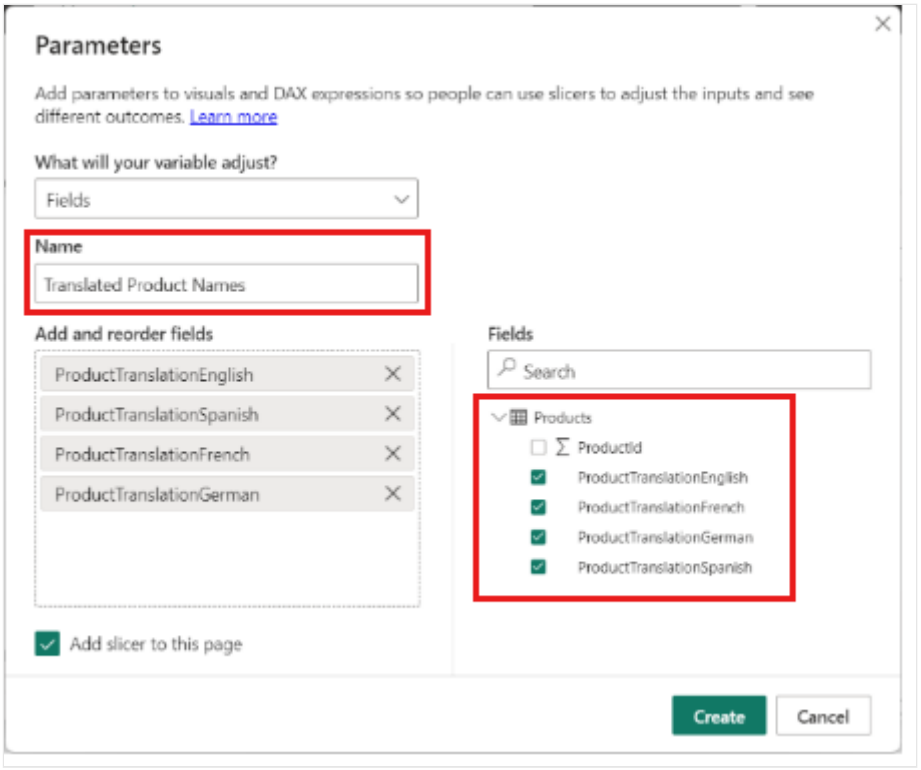
Create a field parameter

1. To create a field parameter in Power BI Desktop, in **Modeling**, select **New parameter** > **Fields**.



2. In the **Parameters** dialog box, enter a name *Translated Product Names*.

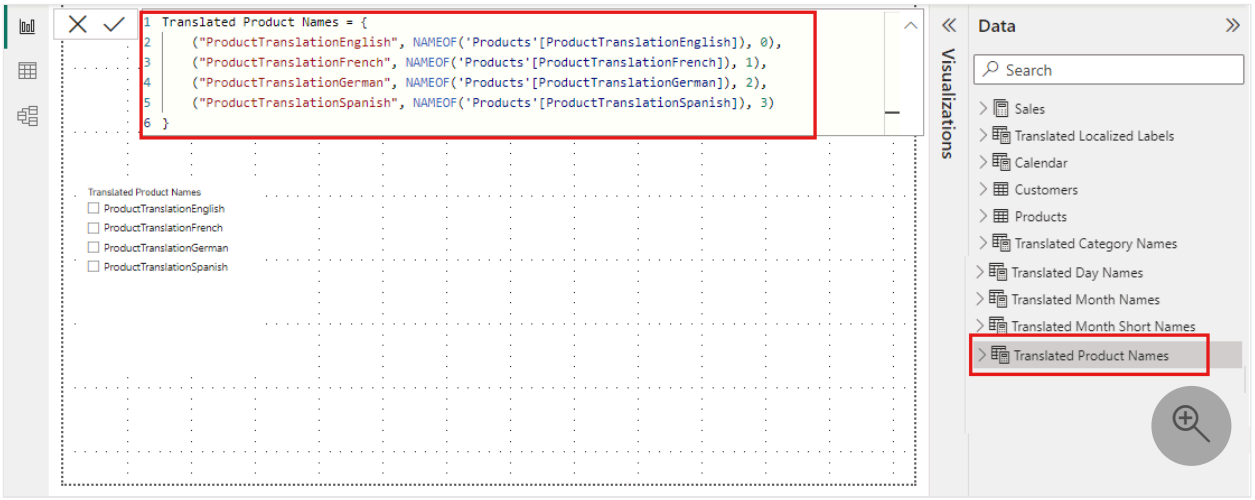
3. Populate the fields connection of this field parameter with the columns from the **Products** table with the translated product names.



4. Be sure that **Add slicer to this page** is enabled.

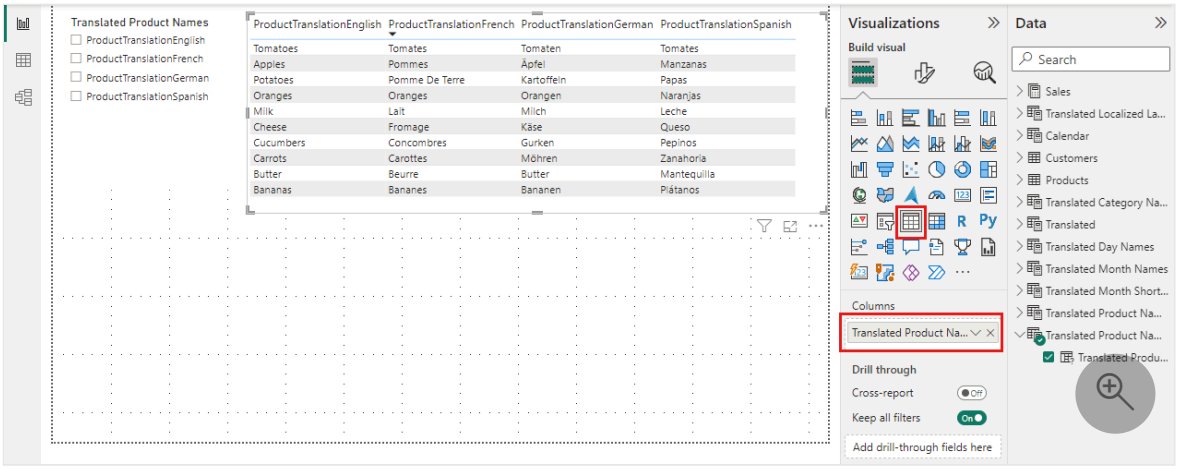
5. Select **Create**.

After you create a field parameter, it appears in the **Fields** list on the right as a new table. Under **Data**, select **Translated Product Names** to see the Data Analysis Expressions (DAX) code that defines the field parameter, as shown in the following screenshot.



Use a slicer and data table

1. Under **Data**, expand the **Translated Product Names** node. Then select the **Translated Product Names** item. A table appears in the canvas.

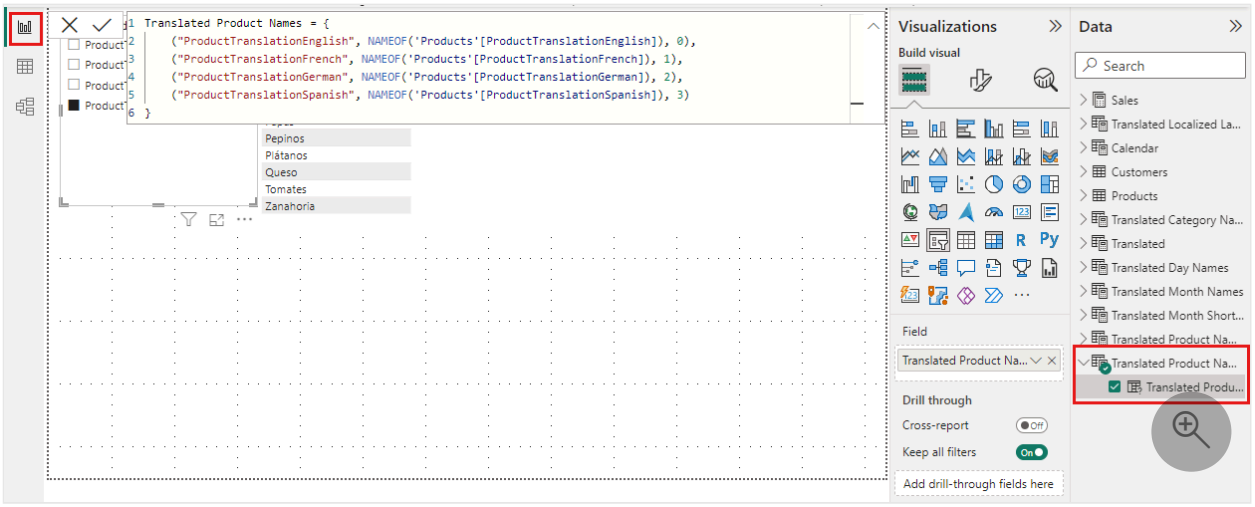


You can see the table type under **Visualizations** and **Translated Product Names** as the **Columns** value. Position both the slicer and the data table anywhere on the canvas.

2. Select one item in the slicer, such as **ProductTranslationSpanish**. The table now shows a single corresponding column.

Edit translated names

The column values for product names have been translated into Spanish. The column header still displays the column name from the underlying data source, which is **ProductTranslationSpanish**. This fact is because those column header values are hard-coded into the DAX expression when Power BI Desktop creates the field parameter.



If you examine the DAX expression, the hard-coded column names from the underlying data source appear, such as **ProductTranslationEnglish** and **ProductTranslationSpanish**.

DAX

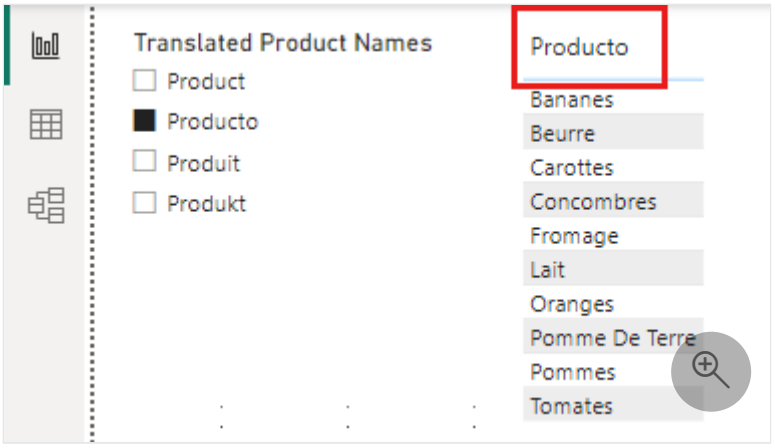
```
Translated Product Names = {
  ("ProductTranslationEnglish",
NAMEOF('Products'[ProductTranslationEnglish]), 0),
  ("ProductTranslationSpanish",
NAMEOF('Products'[ProductTranslationSpanish]), 1),
  ("ProductTranslationFrench", NAMEOF('Products'[ProductTranslationFrench]),
2),
  ("ProductTranslationGerman", NAMEOF('Products'[ProductTranslationGerman]),
3)
}
```

Update the DAX expression to replace the column names with localized translations for the word **Product** as shown in the following code.

DAX

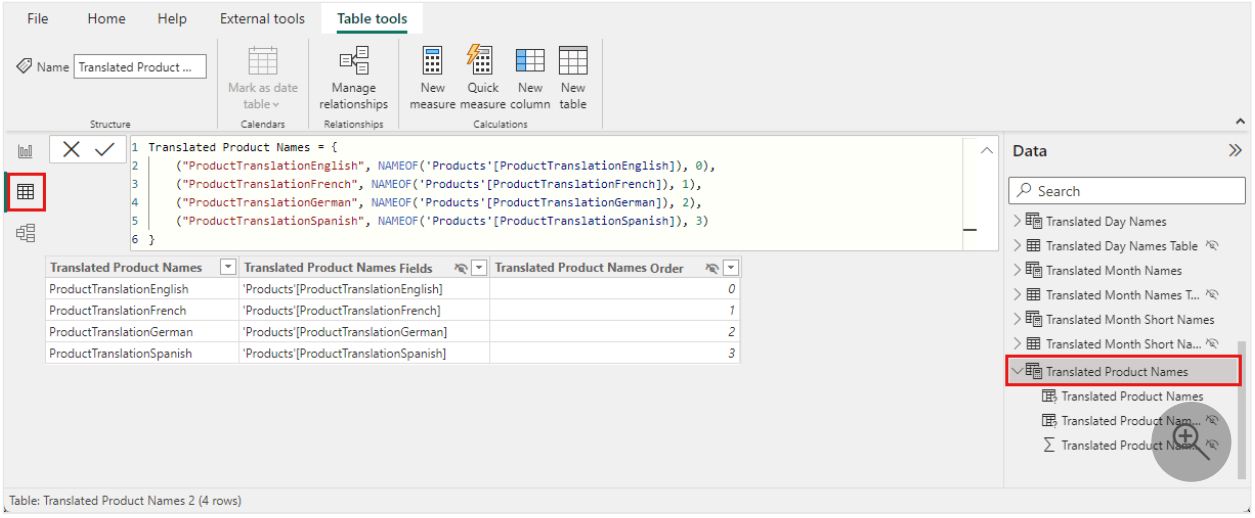
```
Translated Product Names = {
  ("Product", NAMEOF('Products'[ProductTranslationEnglish]), 0),
  ("Producto", NAMEOF('Products'[ProductTranslationSpanish]), 1),
  ("Produit", NAMEOF('Products'[ProductTranslationFrench]), 2),
  ("Produkt", NAMEOF('Products'[ProductTranslationGerman]), 3)
}
```

When you make this change, the column header is translated along with product names.



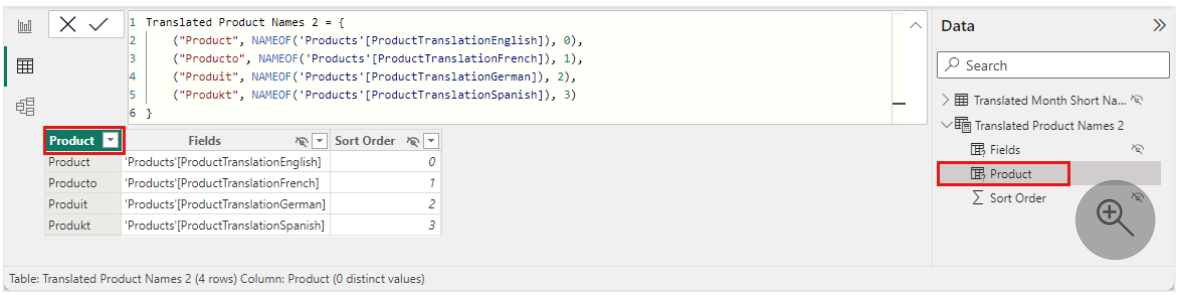
Edit column names in the Table view

Up to this point, you've looked at the field parameter in Report view. Now open the Table view. There you can see two more fields in the field parameter that are hidden in Report view.

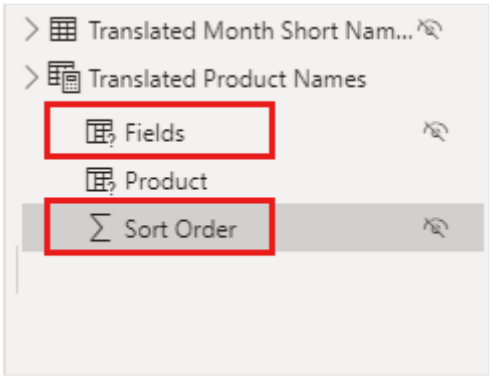


The names of the columns in a field parameter are generated based on the name you give to the top-level field parameter. You should rename the columns to simplify the data model and to improve readability.

1. To rename a column label, double-click the field. Rename **Translated Product Names** to *Product*.



2. Rename the two hidden fields with shorter names, such as *Fields* and *Sort Order*.



Add a language ID column

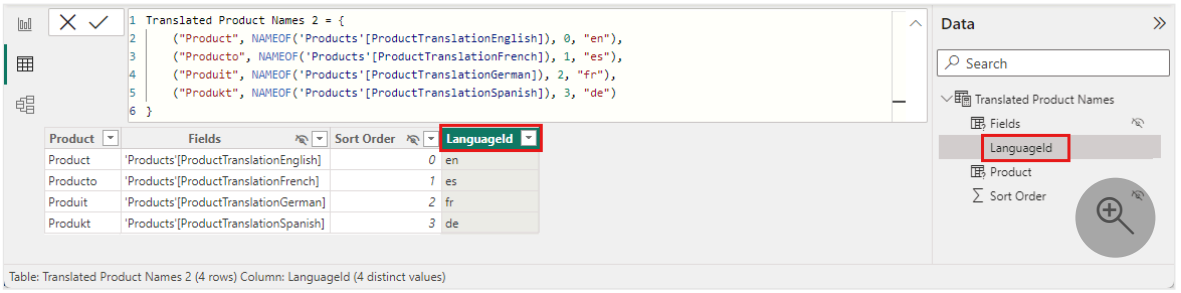
The field parameter is a table with three columns named **Product**, **Fields**, and **Sort Order**. The next step is to add a fourth column with a language identifier to enable filtering by language. You can add the column by modifying the DAX expression for the field parameter.

1. Add a fourth string parameter to the row for each language with the lower-case two character language identifier.

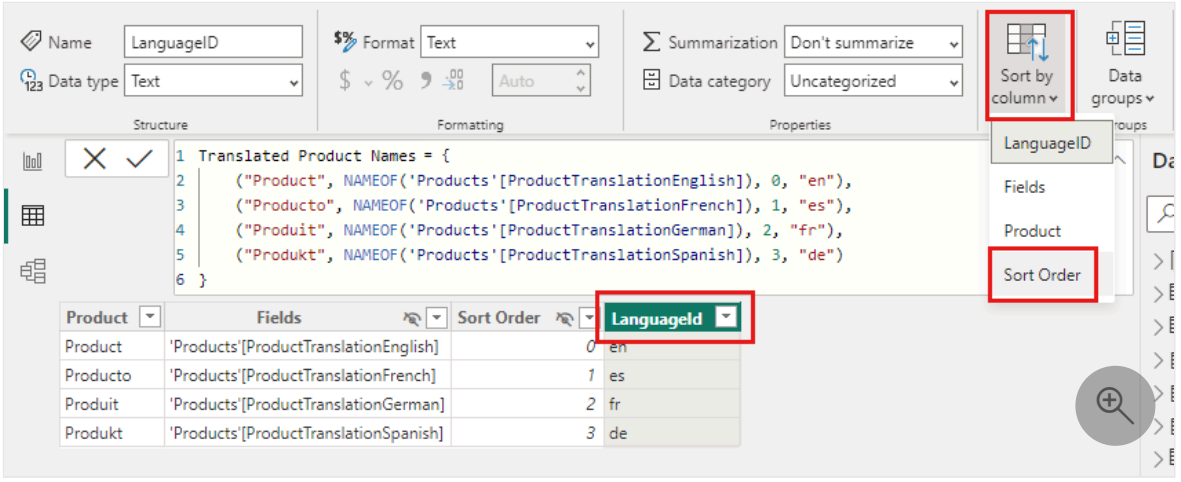
```DAX
Translated Product Names = {
  ("Product", NAMEOF('Products'[ProductTranslationEnglish]), 0, "en" ),
  ("Producto", NAMEOF('Products'[ProductTranslationSpanish]), 1, "es"
),
  ("Produit", NAMEOF('Products'[ProductTranslationFrench]), 2, "fr" ),
  ("Produkt", NAMEOF('Products'[ProductTranslationGerman]), 3, "de" )
}
```

   After you update the DAX expression with a language identifier for each language, a new column appears in the **Table** view of the **Products** table named **Value4**.

2. Double-click the name **Value4** and rename it to *LanguageId*.

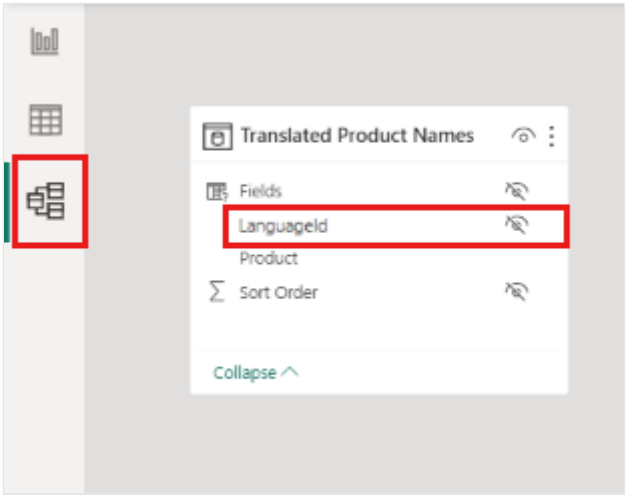


3. Select **LanguageId** to highlight it. From the control ribbon, select **Sort by column** > **Sort Order**.



You don't need to configure the sort column for the two pre-existing fields. Power BI Desktop configured them when you set up the field parameter.

4. Open the Model view and, next to **LanguageId** select **More options** (three dots). Select **Hide in report view**.



Report authors never need to see this column because it's used to select a language by filtering behind the scenes.

In this article, you created a field parameter named **Translated Product Names** and extended it with a column named **LanguageId**. The **LanguageId** column is used to filter

which source column is used. That action determines which language is displayed to report consumers.

Related content

- Add the languages table to filter field parameters

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Add the languages table to filter field parameters

Article • 08/09/2023

As a content creator working with Power BI Desktop, there are many different ways to add a new table to a data model. In this article, you use Power Query to create a table named **Languages**.

Add the table

1. In Power BI Desktop, from the **Home** ribbon, select **Transform data** > **Transform data** to open the Power Query Editor.

2. Under **Queries**, right-click and select **New Query** > **Blank Query** from the context menu.

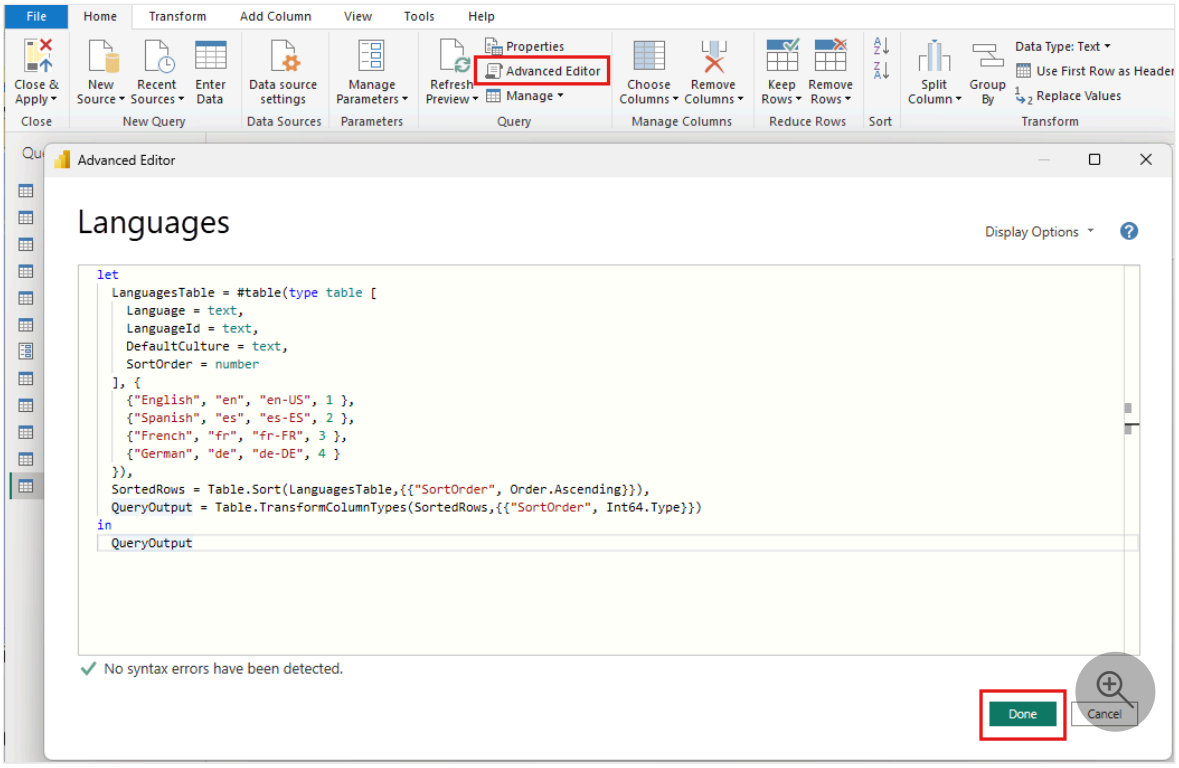
3. Select the new query. Under **Query Settings** > **Properties** > **Name**, enter *Languages* as the name of the query.

4. From the **Home** ribbon, select **Advanced Editor**.

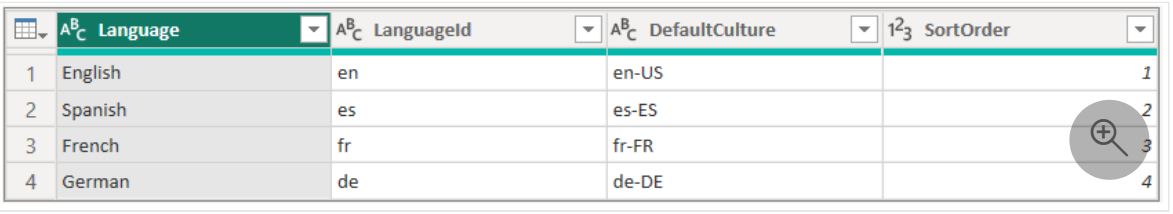
5. Copy the following M code into the editor, then select **Done**.

Power Query M

```
let
  LanguagesTable = #table(type table [
    Language = text,
    LanguageId = text,
    DefaultCulture = text,
    SortOrder = number
  ], {
    {"English", "en", "en-US", 1 },
    {"Spanish", "es", "es-ES", 2 },
    {"French", "fr", "fr-FR", 3 },
    {"German", "de", "de-DE", 4 }
  }),
  SortedRows = Table.Sort(LanguagesTable,{{"SortOrder",
Order.Ascending}}),
  QueryOutput = Table.TransformColumnTypes(SortedRows,{{"SortOrder",
Int64.Type}})
in
  QueryOutput
```



When this query runs, it generates the **Languages** table with a row for each of the four supported languages.



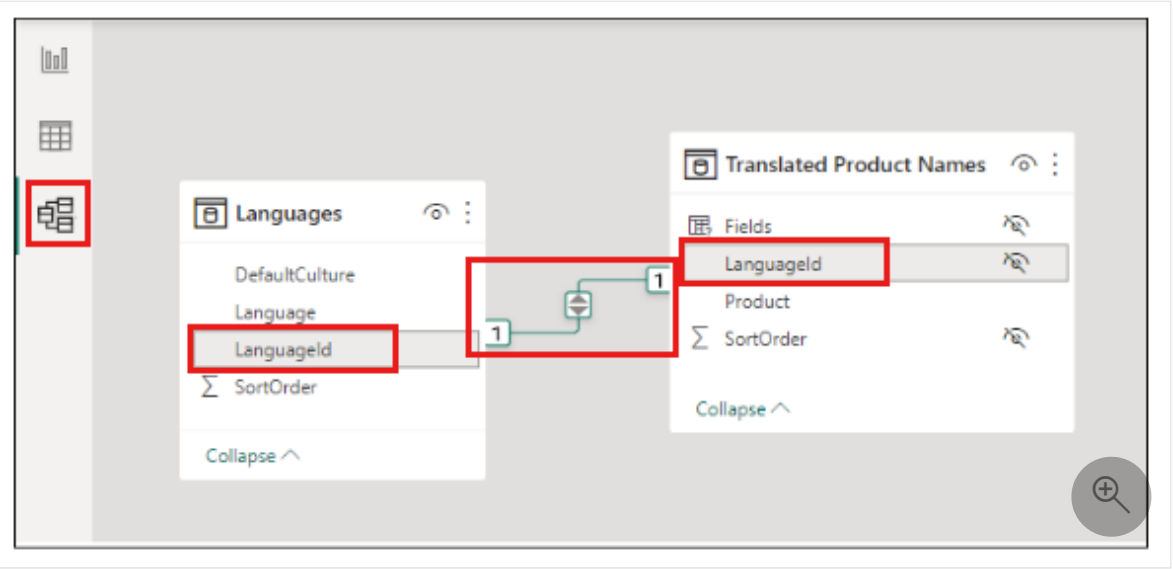
	Language	LanguageId	DefaultCulture	SortOrder
1	English	en	en-US	1
2	Spanish	es	es-ES	2
3	French	fr	fr-FR	3
4	German	de	de-DE	4

6. In the **Home** ribbon, select **Close & Apply**.

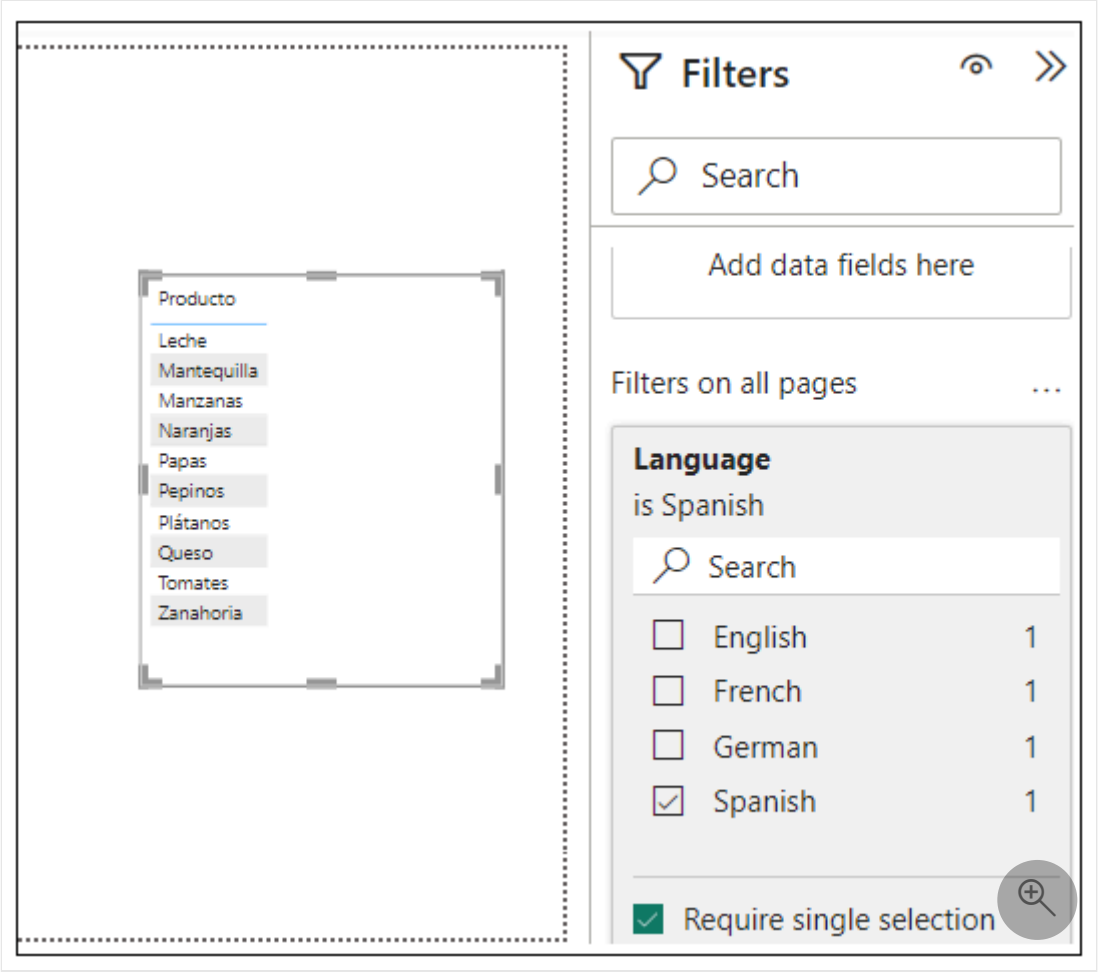
Create a relationship

Next, create a relationship between the **Languages** table and the **Translated Product Names** table created in Implement data translation using field parameters.

1. In Power BI Desktop, open the Model view.
2. Find the **Languages** table and the **Translated Product Names** table.
3. Drag the **LanguageId** column from one table to the **LanguageId** entry in the other table.



After you establish the relationship between **Languages** and **Translated Product Names**, it serves as the foundation for filtering the field parameter on a report-wide basis. For example, you can open the **Filter** pane and add the **Language** column from the **Languages** table to the **Filters on all pages** section. If you configure this filter with the **Require single selection** option, you can switch between languages using the **Filter** pane.



Related content

- Synchronize multiple field parameters

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

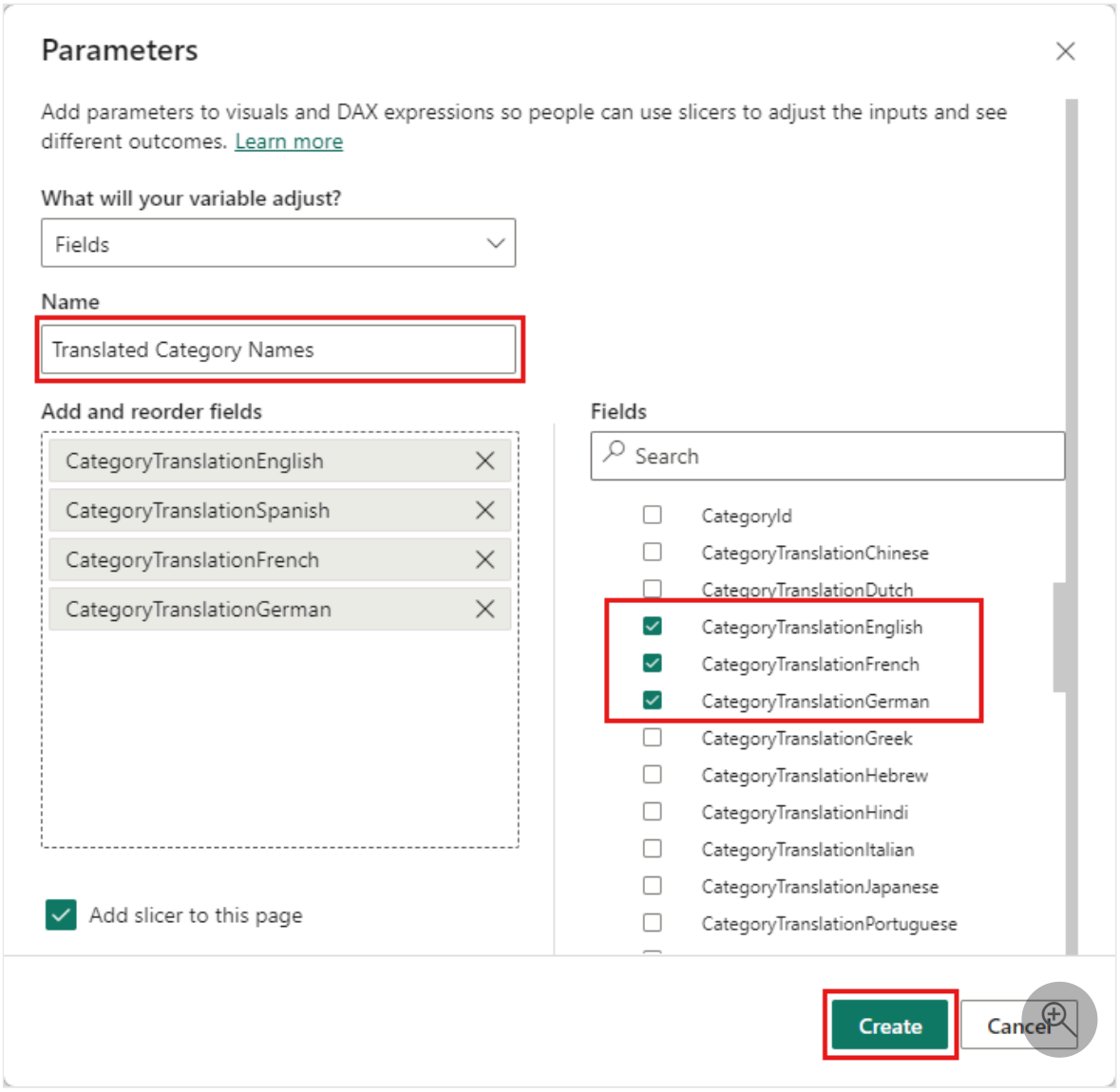
Synchronize multiple field parameters

Article • 08/09/2023

A field parameter can support translations for a column in a multiple-language report in Power BI. Most reports contain more than just one column that requires data translations. You must ensure the mechanism you use to select a language can be synchronized across multiple field parameters. To test this approach working with the project in this series of articles, create a second field parameter to translate product category names from the **Products** table.

Create a field parameter

1. In Power BI Desktop, in the **Modeling** ribbon, select **New parameter** > **Fields**.
2. In the **Parameters** dialog box, enter the name *Translated Category Names*.
3. Populate the fields with the columns from the **Products** table for the desired languages.



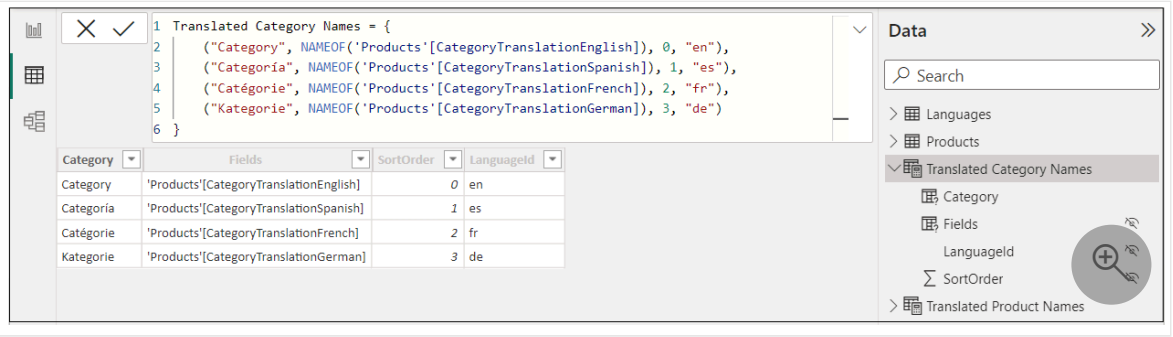
4. Select **Create**.

5. Open the Table view. Select the table to view the Data Analysis Expressions (DAX) code. Update the code to match the following code.

```DAX
Translated Category Names = {
  ("Category", NAMEOF('Products'[CategoryTranslationEnglish]), 0,
"en"),
  ("Categoría", NAMEOF('Products'[CategoryTranslationSpanish]), 1,
"es"),
  ("Catégorie", NAMEOF('Products'[CategoryTranslationFrench]), 2,
"fr"),
  ("Kategorie", NAMEOF('Products'[CategoryTranslationGerman]), 3, "de")
}
```

   After you make your changes, the **Category** value is localized and there's a new column.

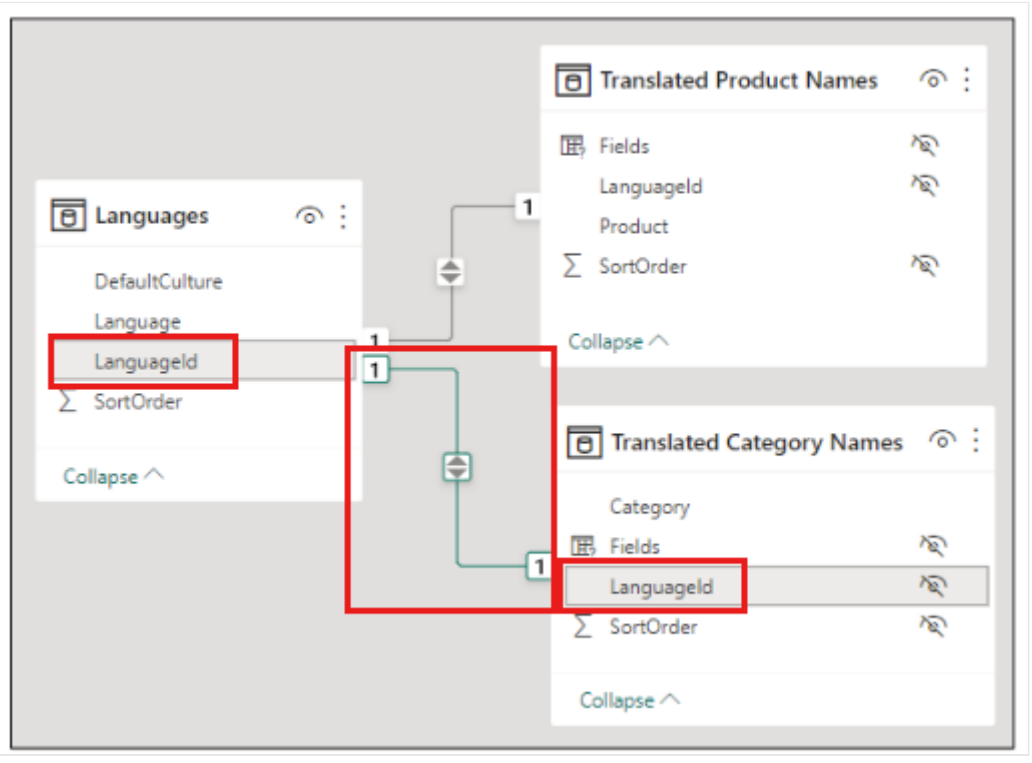
6. Double-click **Value4** and change the name to *LanguageId*.



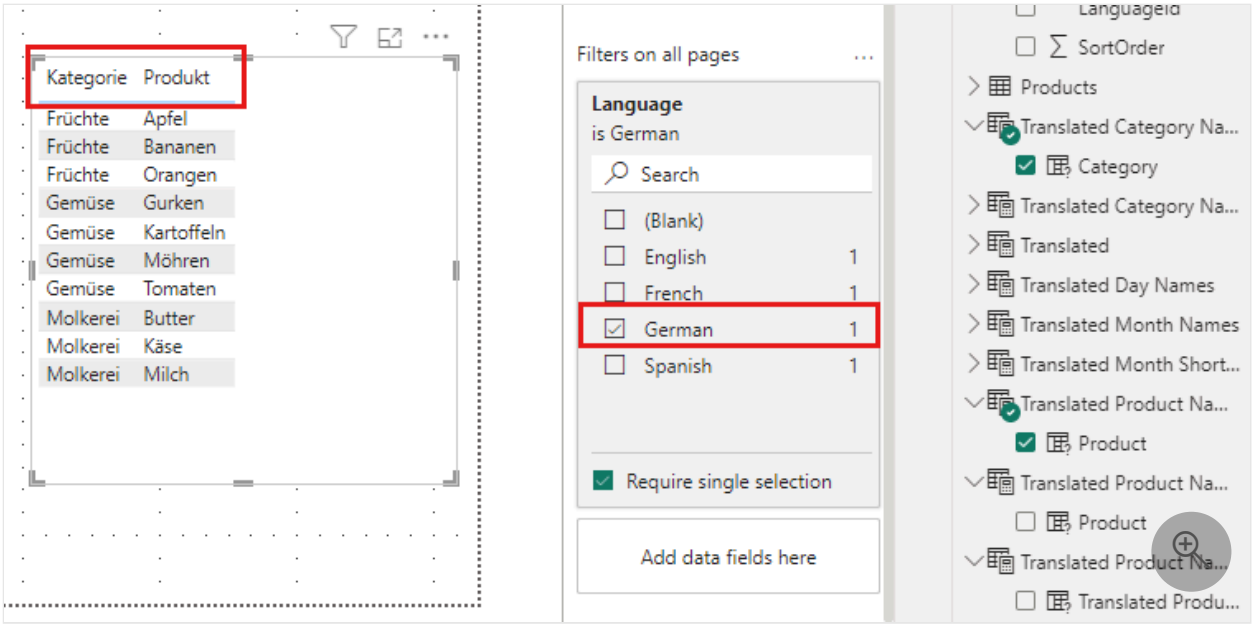
Update the model

After you create the new field parameter, you need to update the model to use it.

1. In Power BI Desktop, open the Model view.
2. Locate the **Translated Category Names** table and the **Languages** table.
3. Drag **LanguageId** from **Translated Category Names** to the **Languages** table to create a one-to-one relationship.



The language filter now affects categories.



You have now learned how to synchronize the selection of language across multiple field parameters. This example involves two field parameters. If your project involves a greater number of columns requiring data translations such as 10, 20 or even 50, you can repeat this approach and scale up as much as you need.

> **Note**
>
> You can test your implementation of data translations in Power BI Desktop by changing the filter on the **Languages** table. However, the other two types of translations don't work correctly in Power BI Desktop. You have to test metadata and report label translations in the Power BI service.

Related content

- Implement data translations for a calendar table

Feedback

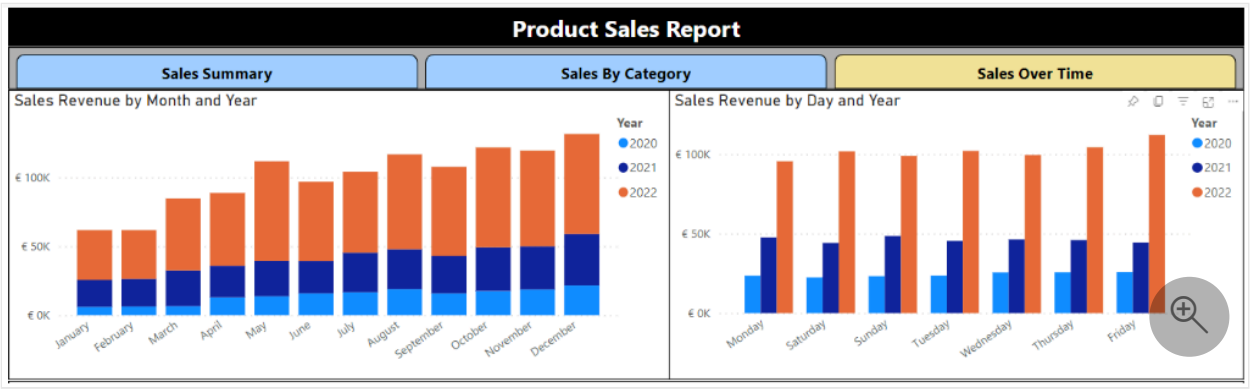
Was this page helpful? Yes No

Provide product feedback | Ask the community

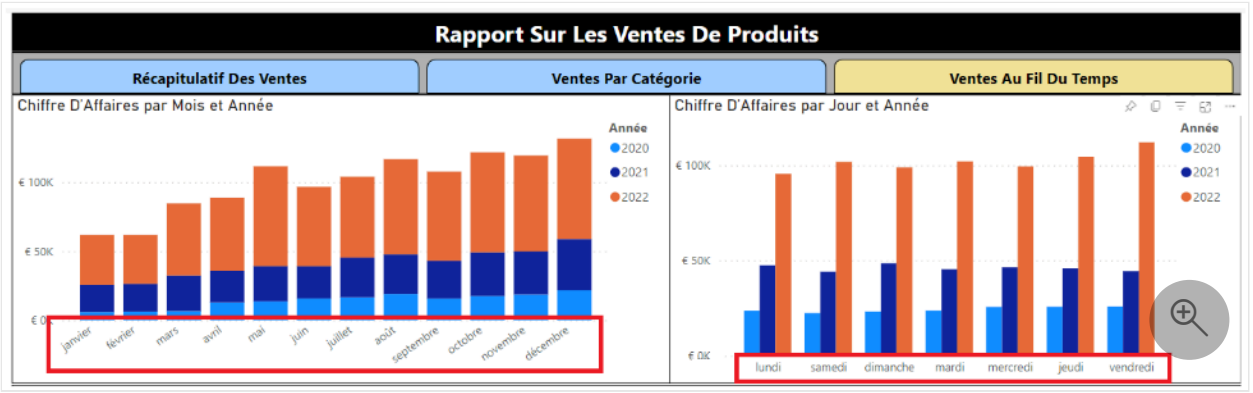
Implement data translations for a calendar table

Article • 11/10/2023

If you're implementing data translations, you can add translation support for text-based columns in calendar tables. These tables include translations for the names of months or the days of the week. This approach allows you to create visuals that mention days or months.



Translated versions make the visual easy to read in your supported languages.



The strategy in this article for calendar table column translations uses Power Query and the M query language. Power Query provides built-in functions, such as `Date.MonthName`, which accept a `Date` parameter and return a text-based calendar name. If your .pbix project has **en-US** as its default language and locale, the following Power Query function call evaluates to a text-based value of *January*.

Power Query M

```
Date.MonthName( #date(2023, 1, 1) )
```

The `Date.MonthName` function accepts an second, optional string parameter to pass a specific language and locale.

Power Query M

```
Date.MonthName( #date(2023, 1, 1), "en-US")
```

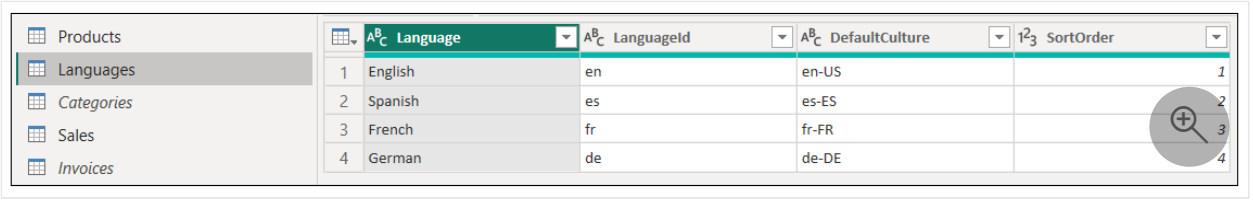
If you want to translate the month name into French, you can pass a text value of **fr-FR**.

Power Query M

```
Date.MonthName( #date(2022, 12, 1), "fr-FR")
```

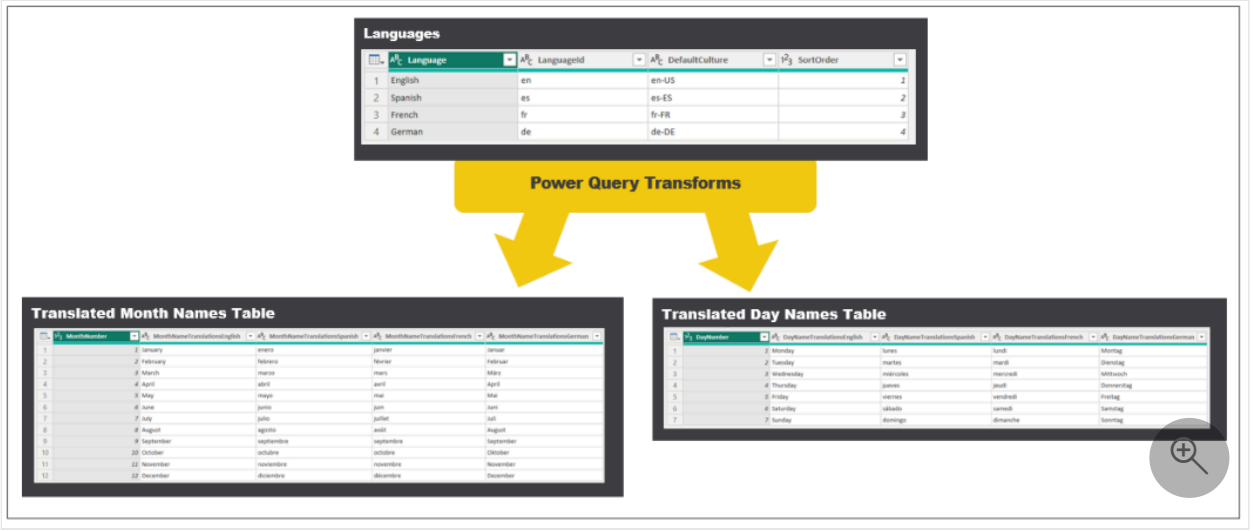
Generate calendar translation table

Look at the **Languages** table used in previous examples. It includes a **DefaultCulture** column.



	Language	LanguageId	DefaultCulture	SortOrder
1	English	en	en-US	1
2	Spanish	es	es-ES	2
3	French	fr	fr-FR	3
4	German	de	de-DE	4

Power Query is built on a functional query language named M. With that language, you can iterate through the rows of the **Languages** table to discover what languages and what default cultures the project supports. You can write a query that uses the **Languages** table as its source to generate a calendar translation table with the names of months or weekdays.



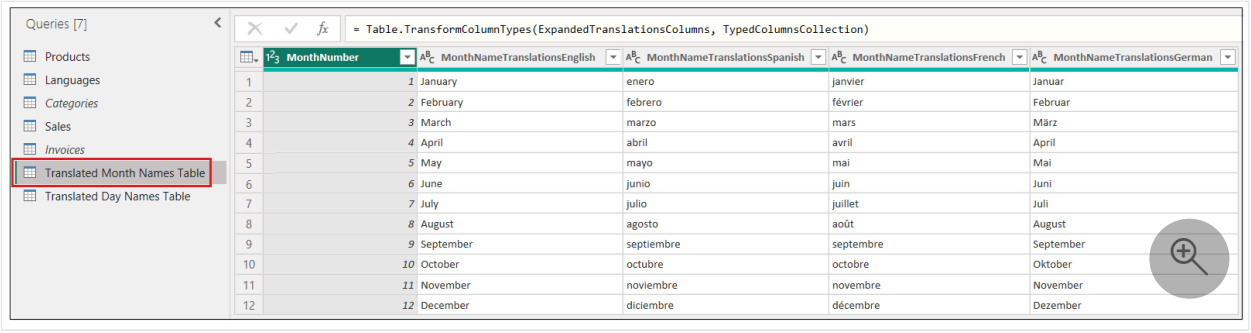
Here's the M code that generates the **Translated Month Names Table**.

Power Query M

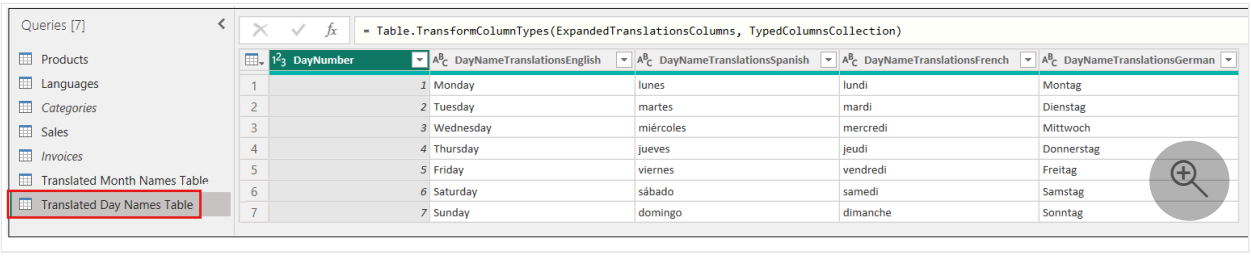
```
let
  Source = #table( type table [ MonthNumber = Int64.Type ],
List.Split({1..12},1)),
  Translations = Table.AddColumn( Source, "Translations",
    each
      [ MonthDate = #date( 2022, [ MonthNumber ], 1 ),
        Translations = List.Transform(Languages[DefaultCulture], each
Date.MonthName( MonthDate, _ ) ),
        TranslationTable = Table.FromList( Translations, null ),
        TranslationsTranspose = Table.Transpose(TranslationTable),
        TranslationsColumns = Table.RenameColumns(
          TranslationsTranspose,
          List.Zip({ Table.ColumnNames( TranslationsTranspose ),
            List.Transform(Languages[Language],
            each "MonthNameTranslations" & _ ) })
          )
      ]
  ),
  ExpandedTranslations = Table.ExpandRecordColumn(Translations,
                                                  "Translations",
                                                  { "TranslationsColumns" },
                                                  { "TranslationsColumns"
}),
  ColumnsCollection = List.Transform(Languages[Language], each
"MonthNameTranslations" & _ ),
  ExpandedTranslationsColumns =
Table.ExpandTableColumn(ExpandedTranslations,

"TranslationsColumns",
                                                        ColumnsCollection,
                                                        ColumnsCollection ),
  TypedColumnsCollection = List.Transform(ColumnsCollection, each {_, type
text}),
  QueryOutput = Table.TransformColumnTypes(ExpandedTranslationsColumns,
TypedColumnsCollection)
in
  QueryOutput
```

> **Tip**
>
> You can simply copy and paste the M code from the *ProductSalesMultiLanguage.pbix* sample whenever you need to add calendar translation tables to your project.

If the **Languages** table contains four rows for English, Spanish, French, and German, the **Translated Month Names Table** query generates a table with four translation columns as shown in the following screenshot.



Likewise, the query named **Translated Day Names Table** generates a table with weekday name translations.



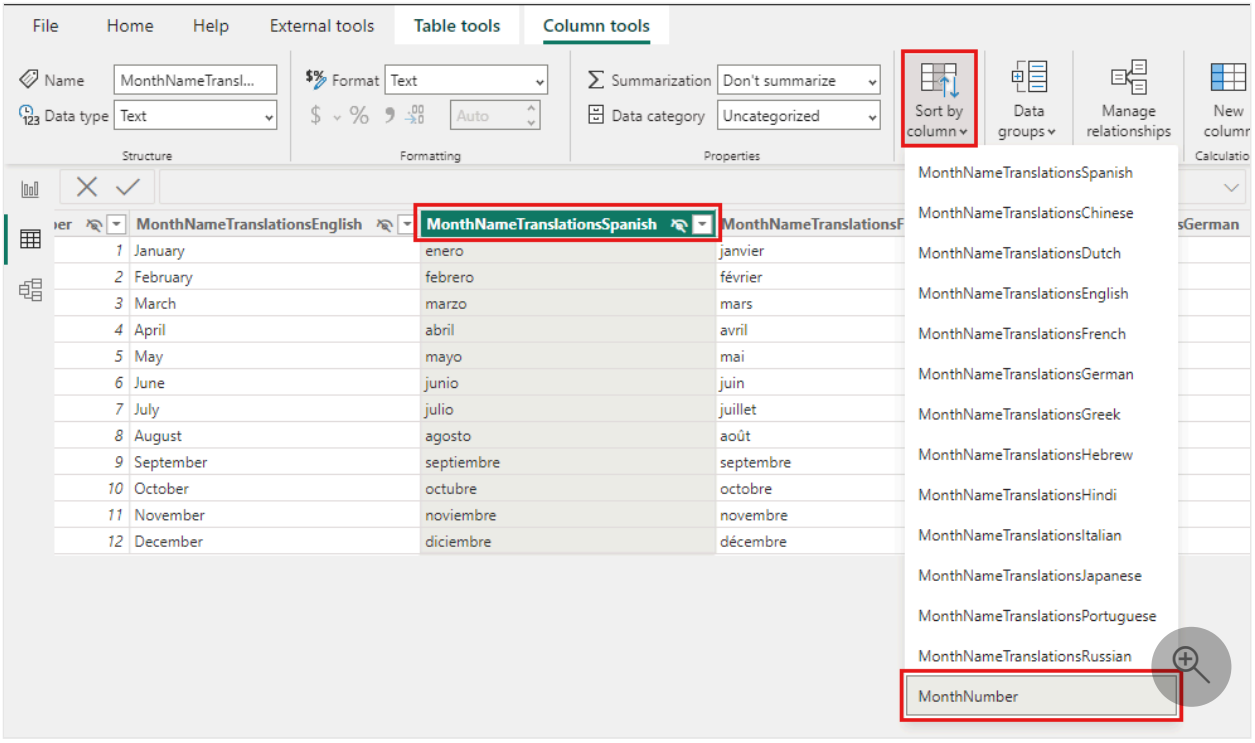
The two queries named **Translated Month Names Table** and **Translated Day Names Table** have been written to be generic. They don't contains any hard-coded column names. These queries don't require any modifications in the future when you add or remove languages from the project. All you need to do is to update the data rows in the query.

Configure sort values

When you run these two queries for the first time, they create two tables in the semantic model with the names **Translated Month Names Table** and **Translated Day Names**

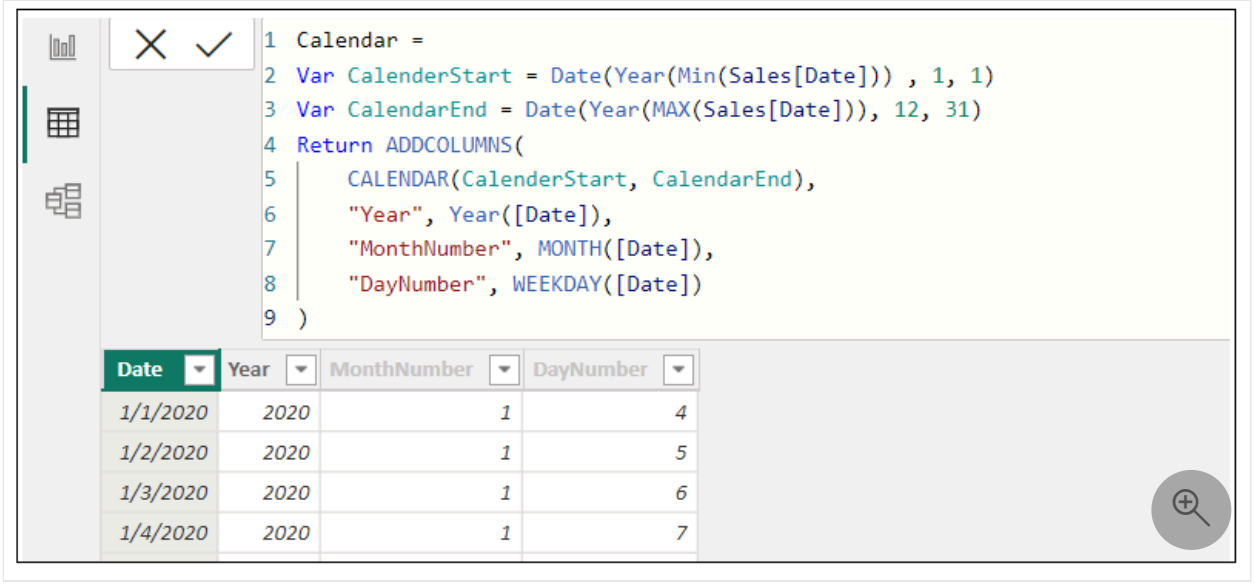
Table. There's a translation column for each language. You need to configure the sort column for each of the translation columns:

- Configure the translation columns in **Translated Month Names Table** to use the sort column **MonthNumber**
- Configure the translations columns in **Translated Day Names Table** to use the sort column **DayNumber**

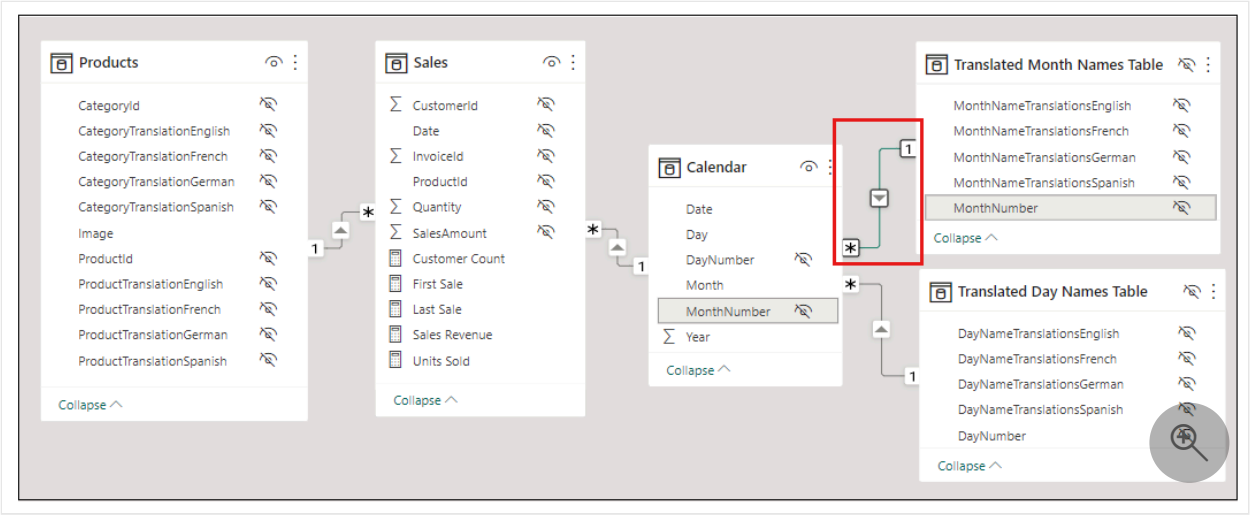


Integrate translation tables

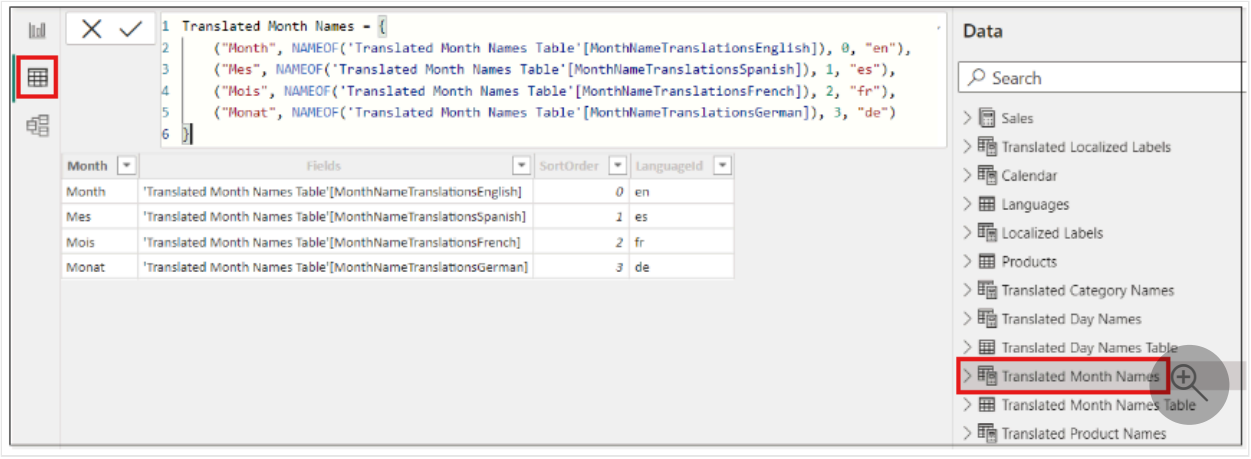
The next step is to integrate the two tables into the data model with a **Calendar** table. The **Calendar** table is a calculated table based on the following Data Analysis Expressions (DAX) expression.



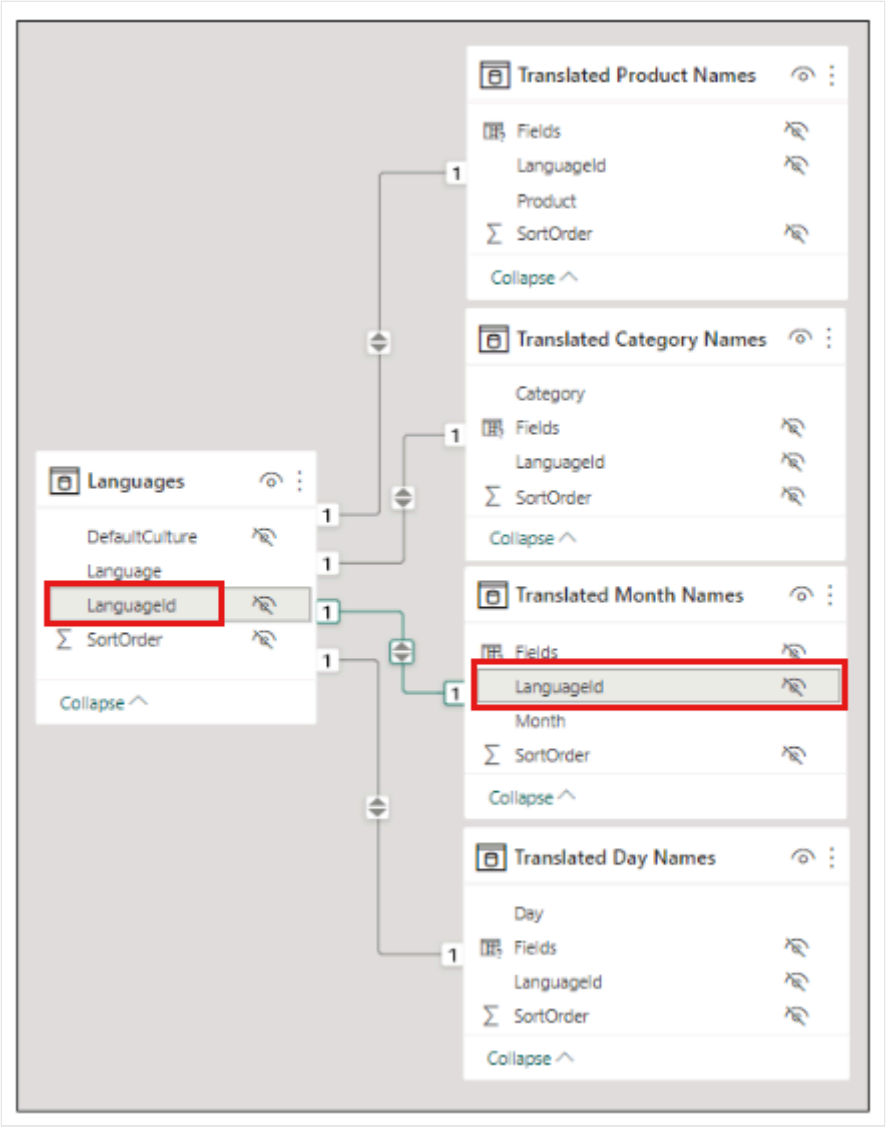
Create a relationship between the **Calendar** table and the fact tables, such as **Sales**, using the **Date** column to create a one-to-many relationship. The relationships created between the **Calendar** table and the two translations tables are based on the **MonthNumber** column and the **DayNumber** column.



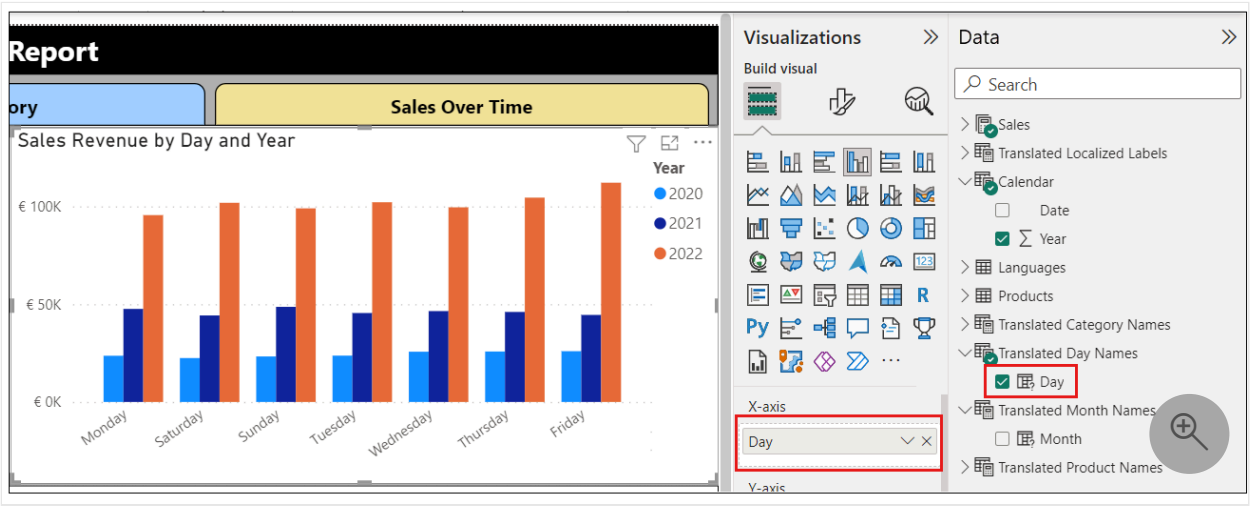
After you create the required relationships with the **Calendar** table, create a new field parameter for each of the two calendar translations tables. Creating a field parameter for a calendar translation table is just like creating the field parameters for product names and category names shown earlier.



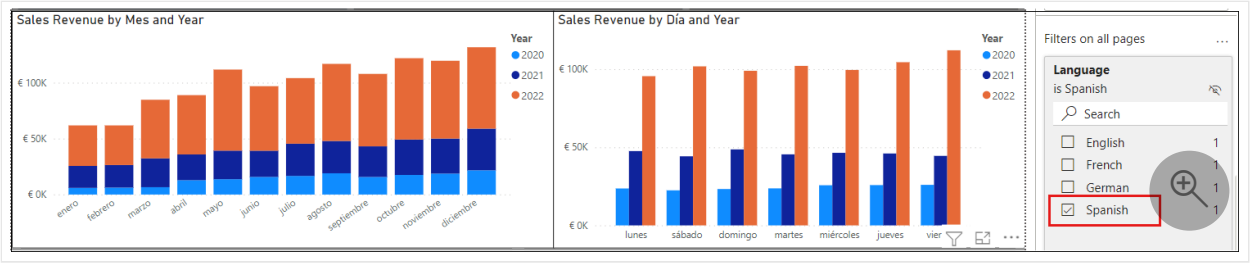
Add a relationship between these new field parameter tables and the **Languages** table to ensure the language filtering strategy works as expected.



After you create the field parameters for **Translated Month Names** and **Translated Day Names**, you can begin to surface them in a report using cartesian visuals, tables, and matrices.



After you set up everything, you can test your work using a report-level filter on the **Languages** table to switch between languages and to verify translations for names of months and days of the week work as expected.



Related content

- Load multiple-language reports

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Load multiple-language reports

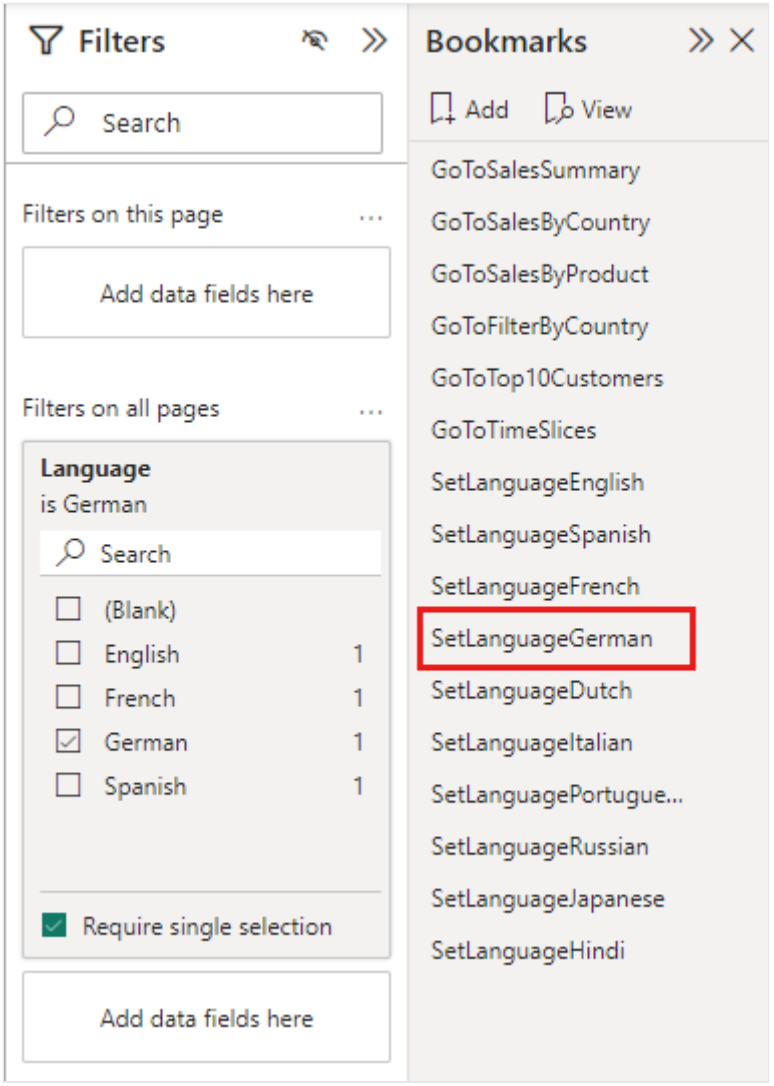
Article • 01/16/2024

To load multiple-language reports in the user's language, you can use bookmarks or embed reports.

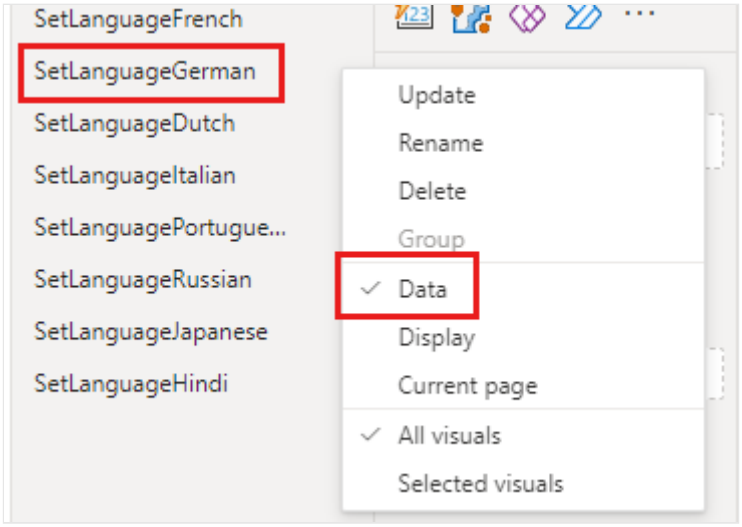
Use bookmarks to select a language

If you plan to publish a Power BI report with data translations for access by users through the Power BI service, you need to load the report with the correct language for the current user. Create a set of bookmarks that apply filters to the **Languages** table.

1. Create a separate bookmark for each language that supports data translations.



2. Disable **Display** and **Current Page** and only enable **Data** behavior.



3. To apply the bookmark for a specific language, supply a second parameter in the report URL.

```HTTP
?language=es&bookmarkGuid=Bookmark856920573e02a8ab1c2a
```

This report URL parameter is named *bookmarkGuid*. The filtering on the **Languages** table is applied before any data is displayed to the user.

Embed reports that implement data translations

Loading reports with Power BI embedding provides more flexibility than the report loading process for users accessing the report through the Power BI service. You can load a report using a specific language and locale by extending the `config` object passed to `powerbi.embed` with a `localeSettings` object containing a `language` property.

```JavaScript
let config = {
  type: "report",
  id: reportId,
  embedUrl: embedUrl,
  accessToken: embedToken,
  tokenType: models.TokenType.Embed,
  localeSettings: { language: "de-DE" }
};

// embed report using config object
let report = powerbi.embed(reportContainer, config);
```

When you embed a report with a **config** object like this which sets the **language** property of the **localeSettings** object, the metadata translations and report label translations work as expected. However, there's one more step required to filter the **Languages** table to select the appropriate language for the current user.

It's possible to apply a bookmark to an embedded a report. Instead, you can apply a filter directly on the **Languages** table as the report loads using the Power BI JavaScript API. There's no need to add bookmarks for filtering the **Languages** table if you only intend to use a report by using Power BI embedding.

To apply a filtering during the loading process of an embedded report, register an event handler for the `loaded` event. When you register an event handler for an embedded report's `loaded` event, you can provide a JavaScript event handler that runs before the rendering process begins. This approach makes the `loaded` event the ideal place to register an event handler whose purpose is to apply the correct filtering on the **Languages** table.

Here's an example of JavaScript code that registers an event handler for the `loaded` event to apply a filter to the **Languages** table for Spanish.

JavaScript

```
let report = powerbi.embed(reportContainer, config);

report.on("loaded", async (event: any) => {

  // let's filter data translations for Spanish
  let languageToLoad = "es";

  // create filter object
  const filters = [{
    $schema: "http://powerbi.comproduct/schema#basic",
    target: {      table: "Languages",
      column: "LanguageId"
    },
    operator: "In",
    values: [ languageToLoad ], // <- Filter based on Spanish
    filterType: models.FilterType.Basic,
    requireSingleSelection: true
  }];

  // call updateFilters and pass filter object to set data translations to
Spanish
  await report.updateFilters(models.FiltersOperations.Replace, filters);

});
```

> **Tip**
>
> When you set filters with the Power BI JavaScript API, you should prefer the `updateFilters` method over the `setFilters` method. The `updateFilters` method allows you to remove existing filters while `setFilters` does not.

Related content

- Guidance for Power BI

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

On-premises data gateway sizing

Article • 12/30/2024

This article targets Fabric administrators who need to install and manage the on-premises data gateway.

The gateway is required whenever Power BI must access data that isn't accessible directly over the Internet. It can be installed on a server on-premises, or VM-hosted infrastructure as a service (IaaS).

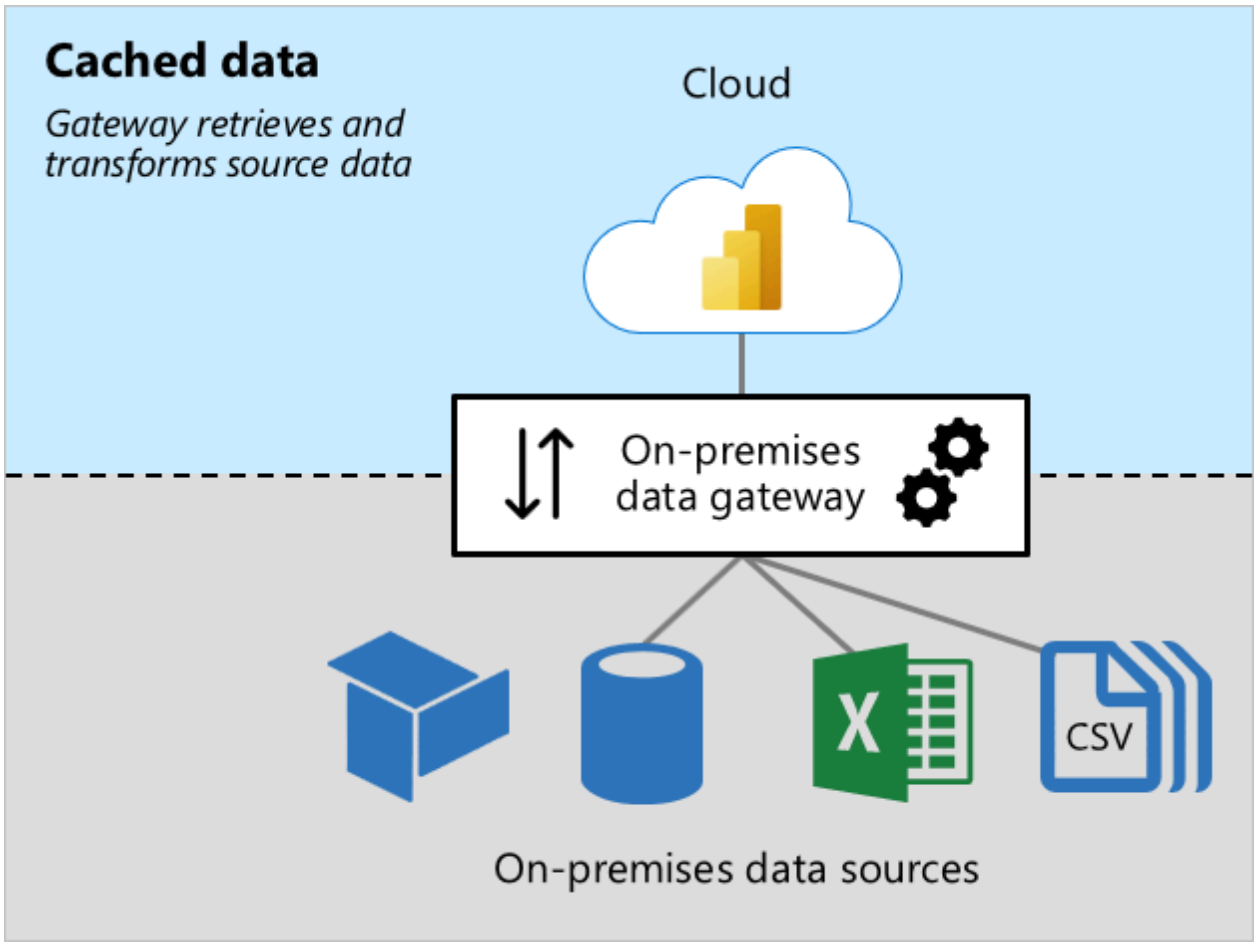
Gateway workloads

The on-premises data gateway supports two workloads. It's important you first understand these workloads before we discuss gateway sizing and recommendations.

Cached data workload

The *Cached data* workload retrieves and transforms source data for loading into Power BI semantic models. It does so in three steps:

1. **Connection**: The gateway connects to source data.
2. **Data retrieval and transformation**: Data is retrieved, and when necessary, transformed. Whenever possible, the Power Query mashup engine pushes transformation steps to the data source—it's known as *query folding*. When it's not possible, transformations must be done by the gateway. In this case, the gateway will consume more CPU and memory resources.
3. **Transfer**: Data is transferred to the Power BI service—a reliable and fast Internet connection is important, especially for large data volumes.



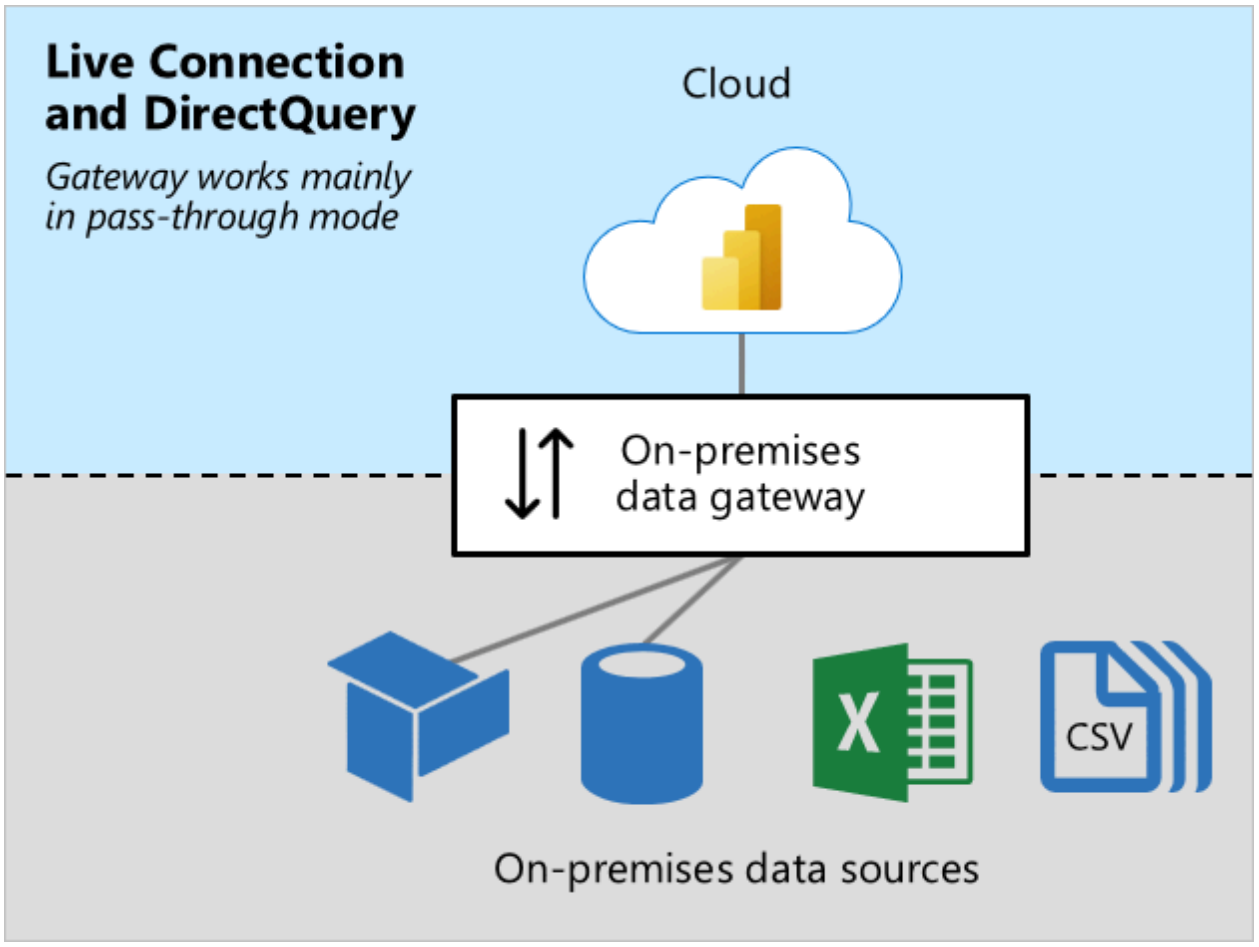
Live Connection and DirectQuery workloads

The *Live Connection and DirectQuery* workload works mostly in pass-through mode. The Power BI service sends queries, and the gateway responds with query results. Generally, query results are small in size.

- For more information about Live Connection, see Semantic models in the Power BI service (Externally-hosted models).
- For more information about DirectQuery, see Semantic model modes in the Power BI service (DirectQuery mode).

This workload requires CPU resources for routing queries and query results. Usually there's much less demand for CPU than is required by the Cache data workload—especially when it's required to transform data for caching.

Reliable, fast, and consistent connectivity is important to ensure report users have responsive experiences.



Sizing considerations

Determining the correct sizing for your gateway machine can depend on the following variables:

- **For Cache data workloads**:
  - The number of concurrent semantic model refreshes
  - The types of data sources (relational database, analytic database, data feeds, or files)
  - The volume of data to be retrieved from data sources
  - Any transformations required to be done by the Power Query mashup engine
  - The volume of data to be transferred to the Power BI service
- **For Live Connection and DirectQuery workloads**:
  - The number of concurrent report users
  - The number of visuals on report pages (each visual sends at least one query)
  - The frequency of Power BI dashboard query cache updates
  - The number of real-time reports using the Automatic page refresh feature
  - Whether semantic models enforce Row-level Security (RLS)

Generally, Live Connection and DirectQuery workloads require sufficient CPU, while Cache data workloads require more CPU and memory. Both workloads depend on good

connectivity with the Power BI service, and the data sources.

> **Note**
>
> Power BI capacities impose limits on model refresh parallelism, and Live Connection and DirectQuery throughput. There's no point sizing your gateways to deliver more than what the Power BI service supports. Limits differ by Premium SKU (and equivalently sized A SKU). For more information, see Microsoft Fabric capacity licenses and What is Power BI Premium? (Capacity nodes).

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Recommendations

Gateway sizing recommendations depend on many variables. In this section, we provide you with general recommendations that you can take into consideration.

Initial sizing

It can be difficult to accurately estimate the right size. We recommend that you start with a machine with at least 8 CPU cores, 8 GB of RAM, and multiple Gigabit network adapters. You can then measure a typical gateway workload by logging CPU and memory system counters. For more information, see Monitor and optimize on-premises data gateway performance.

Connectivity

Plan for the best possible connectivity between the Power BI service and your gateway, and your gateway and the data sources.

- Strive for reliability, fast speeds, and low, consistent latencies.
- Eliminate—or reduce—machine hops between the gateway and your data sources.

- Remove any network throttling imposed by your firewall proxy layer. For more information about Power BI endpoints, see Add Power BI URLs to your allow list.
- Set up Azure ExpressRoute to establish private, managed connections to Power BI.
- For data sources in Azure VMs, ensure the VMs are colocated with the Power BI service.
- For Live Connection workloads to SQL Server Analysis Services (SSAS) involving dynamic RLS, ensure good connectivity between the gateway machine and the on-premises Active Directory.

Clustering

For large-scale deployments, you can create a gateway with multiple cluster members. Clusters avoid single points of failure, and can load balance traffic across gateways. You can:

- Install one or more gateways in a cluster.
- Isolate workloads to standalone gateways, or clusters of gateway servers.

For more information, see Manage on-premises data gateway high-availability clusters and load balancing.

Semantic model design and settings

Semantic model design, and their settings, can impact on gateway workloads. To reduce gateway workload, you can consider the following actions.

For Import semantic models:

- Set up less frequent data refresh.
- Set up incremental refresh to minimize the amount of data to transfer.
- Whenever possible, ensure query folding takes place.
- Especially for large data volumes or a need for low-latency results, convert the design to a DirectQuery or Composite model.

For DirectQuery semantic models:

- Optimize data sources, model, and report designs—for more information, see DirectQuery model guidance in Power BI Desktop.
- Create aggregations to cache higher-level results to reduce the number of DirectQuery requests.
- Restrict Automatic page refresh intervals, in report designs and capacity settings.
- Especially when dynamic RLS is enforced, restrict dashboard cache update frequency.

- Especially for smaller data volumes or for non-volatile data, convert the design to an Import or Composite model.

For Live Connection semantic models:

- Especially when dynamic RLS is enforced, restrict dashboard cache update frequency.

Related content

For more information related to this article, check out the following resources:

- Power BI implementation planning: Data gateways
- Guidance for deploying a data gateway for Power BI
- Configure proxy settings for the on-premises data gateway
- Monitor and optimize on-premises data gateway performance
- Troubleshoot gateways - Power BI
- Troubleshoot the on-premises data gateway
- The importance of query folding
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Monitor report performance in Power BI

Article • 12/30/2024

Monitor report performance in Power BI Desktop using the Performance Analyzer. Monitoring will help you learn where the bottlenecks are, and how you can improve report performance.

Monitoring performance is relevant in the following situations:

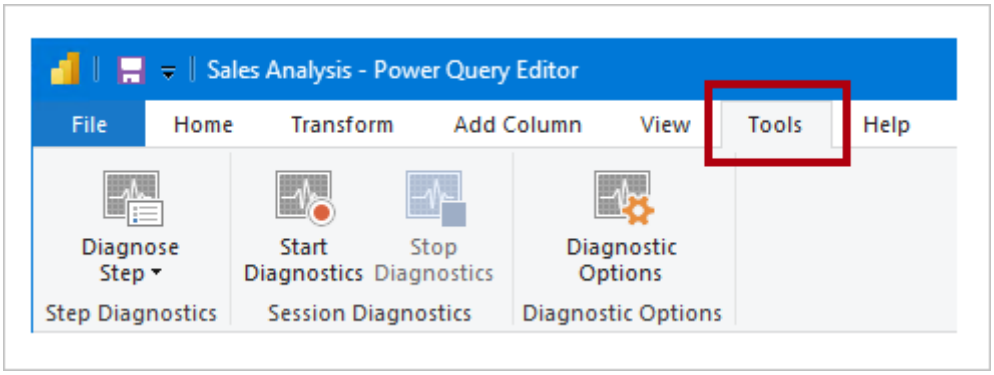
- Your Import data model refresh is slow.
- Your DirectQuery or Live Connection reports are slow.
- Your model calculations are slow.

Slow queries or report visuals should be a focal point of continued optimization.

> **Note**
>
> The Performance Analyzer cannot be used to monitor Premium Per User (PPU) activities or capacity.

Use Query Diagnostics

Use Query Diagnostics in Power BI Desktop to determine what Power Query is doing when previewing or applying queries. Further, use the *Diagnose Step* function to record detailed evaluation information for each query step. The results are made available in a Power Query, and you can apply transformations to better understand query execution.



Use Performance Analyzer

Use Performance Analyzer in Power BI Desktop to find out how each of your report elements—such as visuals and DAX formulas—are doing. It's especially useful to

determine whether it's the query or visual rendering that's contributing to performance issues.

Use SQL Server Profiler

You can also use SQL Server Profiler to identify queries that are slow.

> ⓘ **Note**
>
> SQL Server Profiler is available as part of SQL Server Management Studio.

Use SQL Server Profiler when your data source is either:

- SQL Server
- SQL Server Analysis Services
- Azure Analysis Services

> ⊗ **Caution**
>
> Power BI Desktop supports connecting to a diagnostics port. The diagnostic port allows for other tools to make connections to perform traces for diagnostic purposes. Making any changes to the Power Desktop data model is supported **only for specific operations**. Other changes to the data model with operations that aren't supported may lead to corruption and data loss.

To create a SQL Server Profiler trace, follow these instructions:

1. Open your Power BI Desktop report (so it will be easy to locate the port in the next step, close any other open reports).
2. To determine the port being used by Power BI Desktop, in PowerShell (with administrator privileges), or at the Command Prompt, enter the following command:

   ```
   netstat -b -n
   ```

   The output will be a list of applications and their open ports. Look for the port used by **msmdsrv.exe**, and record it for later use. It's your instance of Power BI Desktop.
3. To connect SQL Server Profiler to your Power BI Desktop report:
   a. Open SQL Server Profiler.
   b. In SQL Server Profiler, on the *File* menu, select *New Trace*.

c. For **Server Type**, select *Analysis Services*.
d. For **Server Name**, enter *localhost:[port recorded earlier]*.
e. Click *Run*—now the SQL Server Profiler trace is live, and is actively profiling Power BI Desktop queries.

4. As Power BI Desktop queries are executed, you'll see their respective durations and CPU times. Depending on the data source type, you may see other events indicating how the query was executed. Using this information, you can determine which queries are the bottlenecks.

A benefit of using SQL Server Profiler is that it's possible to save a SQL Server (relational) database trace. The trace can become an input to the Database Engine Tuning Advisor. This way, you can receive recommendations on how to tune your data source.

Monitor Premium metrics

Monitor performance of content deployed into your organization's Power BI Premium capacity with the help of the Fabric Capacity Metrics app.

Related content

For more information about this article, check out the following resources:

- Query Diagnostics
- Performance Analyzer
- Troubleshoot report performance in Power BI
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Troubleshoot report performance in Power BI

Article • 12/30/2024

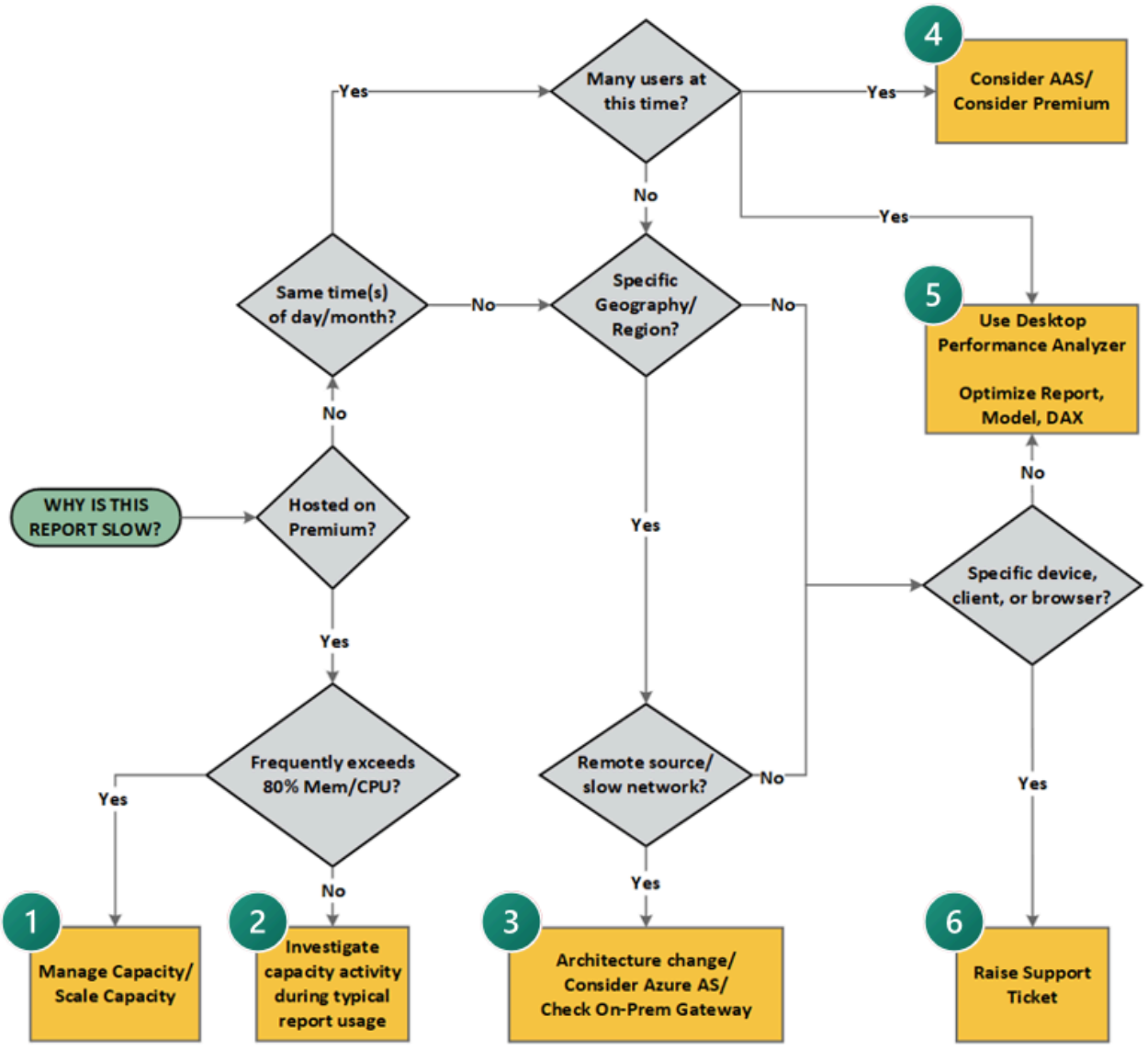
This article provides guidance that enables developers and administrators to troubleshoot slow report performance. It applies to Power BI reports, and also Power BI paginated reports.

Slow reports can be identified by report users who experience reports that are slow to load, or slow to update when interacting with slicers or other features. When reports are hosted on a Premium capacity or Fabric capacity, slow reports can also be identified by monitoring the Fabric Capacity Metrics app. This app helps you to monitor the health and capacity of your Power BI Premium subscription.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Follow flowchart steps

Use the following flowchart to help understand the cause of slow performance, and to determine what action to take.



There are six flowchart terminators, each describing action to take:

Expand table

Terminator	Action(s)
1	• Manage capacity. • Scale capacity.
2	• Investigate capacity activity during typical report usage.
3	• Architecture change. • Consider Azure Analysis Services. • Check on-premises gateway.
4	• Consider Azure Analysis Services. • Consider Power BI Premium.
5	• Use Power BI Desktop Performance Analyzer. • Optimize report, model, or DAX.

Terminator	Action(s)
6	• Raise support ticket.

Take action

The first consideration is to understand if the slow report is hosted on a Premium capacity.

Premium capacity

When the report is hosted on a Premium capacity, use the Fabric Capacity Metrics app to determine if the report-hosting capacity frequently exceeds capacity resources. When there's pressure on resources, it may be time to manage or scale the capacity (flowchart terminator 1). When there are adequate resources, investigate capacity activity during typical report usage (flowchart terminator 2).

Shared capacity

When the report is hosted on shared capacity, it's not possible to monitor capacity health. You'll need to take a different investigative approach.

First, determine if slow performance occurs at specific times of the day or month. If it does—and many users are opening the report at these times—consider two options:

- Increase query throughput by migrating the semantic model to Azure Analysis Services, or a Premium capacity (flowchart terminator 4).
- Use Power BI Desktop Performance Analyzer to find out how each of your report elements—such as visuals and DAX formulas—are doing. It's especially useful to determine whether it's the query or visual rendering that's contributing to performance issues (flowchart terminator 5).

If you determine there's no time pattern, next consider if slow performance is isolated to a specific geography or region. If it is, it's likely that the data source is remote and there's slow network communication. In this case, consider:

- Changing architecture by using Azure Analysis Services (flowchart terminator 3).
- Optimizing on-premises data gateway performance (flowchart terminator 3).

Finally, if you determine there's no time pattern *and* slow performance occurs in all regions, investigate whether slow performance occurs on specific devices, clients, or web

browsers. If it doesn't, use Power BI Desktop Performance Analyzer, as described earlier, to optimize the report or model (flowchart terminator 5).

When you determine specific devices, clients, or web browsers contribute to slow performance, we recommend creating a support ticket through the Power BI support page (flowchart terminator 6).

Related content

For more information about this article, check out the following resources:

- Power BI guidance
- Fabric adoption roadmap
- Monitoring report performance
- Performance Analyzer
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Best practices for lifecycle management in Fabric

Article • 04/15/2025

This article provides guidance for data & analytics creators who are managing their content throughout its lifecycle in Microsoft Fabric. The article focuses on the use of Git integration for source control and deployment pipelines as a release tool. For a general guidance on Enterprise content publishing, Enterprise content publishing.

The article is divided into four sections:

- **Content preparation** - Prepare your content for lifecycle management.
- **Development** - Learn about the best ways of creating content in the deployment pipelines development stage.
- **Test** - Understand how to use a deployment pipelines test stage to test your environment.
- **Production** - Utilize a deployment pipelines production stage to make your content available for consumption.

Best practices for content preparation

To best prepare your content for ongoing management throughout its lifecycle, review the information in this section before you:

- Release content to production.
- Start using a deployment pipeline for a specific workspace.

Separate development between teams

Different teams in the org usually have different expertise, ownership, and methods of work, even when working on the same project. It's important to set boundaries while giving each team their independence to work as they like. Consider having separate workspaces for different teams. With separate workspaces, each team can have different permissions, work with different source control repos, and ship content to production in a different cadence. Most items can connect and use data across workspaces, so it doesn't block collaboration on the same data and project.

Plan your permission model

Both Git integration and deployment pipelines require different permissions than just the workspace permissions. Read about the permission requirements for Git integration and deployment pipelines.

To implement a secure and easy workflow, plan who gets access to each part of the environments being used, both the Git repository and the dev/test/prod stages in a pipeline. Some of the considerations to take into account are:

- Who should have access to the source code in the Git repository?
- Which operations should users with pipeline access be able to perform in each stage?
- Who's reviewing content in the test stage?
- Should the test stage reviewers have access to the pipeline?
- Who should oversee deployment to the production stage?
- Which workspace are you assigning to a pipeline, or connecting to git?
- Which branch are you connecting the workspace to? What's the policy defined for that branch?
- Is the workspace shared by multiple team members? Should they make changes directly in the workspace, or only through Pull requests?
- Which stage are you assigning your workspace to?
- Do you need to make changes to the permissions of the workspace you're assigning?

Connect different stages to different databases

A production database should always be stable and available. It's best not to overload it with queries generated by BI creators for their development or test semantic models. Build separate databases for development and testing in order to protect production data and not overload the development database with the entire volume of production data.

Use parameters for configurations that will change between stages

Whenever possible, add parameters to any definition that might change between dev/test/prod stages. Using parameters helps you change the definitions easily when you move your changes to production.
Parameters have different uses, such as defining connections to data sources, or to internal

items in Fabric. They can also be used to make changes to queries, filters, and the text displayed to users.
In deployment pipelines, you can configure parameter rules to set different values for each deployment stage.

Best practices for deployment pipelines development stage

This section provides guidance for working with the deployment pipelines and using it for your development stage.

Back up your work into a Git repository

With Git integration, any developer can back up their work by committing it into Git. To back up your work properly in Fabric, here are some basic rules:

- Make sure you have an isolated environment to work in, so others don't override your work before it gets committed. This means working in a Desktop tool (such as VS Code, Power BI Desktop or others), or in a separate workspace that other users can't access.
- Commit to a branch that you created and no other developer is using. If you're using a workspace as an authoring environment, read about working with branches.
- Commit together changes that must be deployed together. This advice applies for a single item, or multiple items that are related to the same change. Committing all related changes together can help you later when deploying to other stages, creating pull requests, or reverting changes back.
- Big commits might hit a max commit size limit. Be mindful of the number of items you commit together, or the general size of an item. For example, reports can grow large when adding large images. It's bad practice to store large-size items in source control systems, even if it works. Consider ways to reduce the size of your items if they have lots of static resources, like images.

Rolling back changes

After you back up your work, there might be cases where you want to revert to a previous version and restore it in the workspace. There are a few ways to revert to a previous version:

- **Undo button**: The *Undo* operation is an easy and fast way to revert the immediate changes you made, as long as they aren't committed yet. You can also undo each item

separately. Read more about the undo operation.

- **Reverting to older commits**: There's no direct way to go back to a previous commit in the UI. The best option is to promote an older commit to be the HEAD using git revert or git reset. Doing this shows that there's an update in the source control pane, and you can update the workspace with that new commit.

Since data isn't stored in Git, keep in mind that reverting a data item to an older version might break the existing data and could possibly require you to drop the data or the operation might fail. Check this in advance before reverting changes back.

Working with a 'private' workspace

When you want to work in isolation, use a separate workspace as an isolated environment. Read more about isolating your work environment in working with branches. For an optimal workflow for you and the team, consider the following points:

- **Setting up the workspace**: Before you start, make sure you can create a new workspace (if you don't already have one), that you can assign it to a Fabric capacity, and that you have access to data to work in your workspace.

- **Creating a new branch**: Create a new branch from the *main* branch, so you have the most up-to-date version of your content. Also make sure you connect to the correct folder in the branch, so you can pull the right content into the workspace.

- **Small, frequent changes**: It's a Git best practice to make small incremental changes that are easy to merge and less likely to get into conflicts. If that's not possible, make sure to update your branch from main so you can resolve conflicts on your own first.

- **Configuration changes**: If necessary, change the configurations in your workspace to help you work more productively. Some changes can include connection between items, or to different data sources or changes to parameters on a given item. Just remember that anything you commit becomes part of the commit and can accidentally be merged into the main branch.

Use Client tools to edit your work

For items and tools that support it, it might be easier to work with client tools for authoring, such as Power BI Desktop for semantic models and reports, VS Code for Notebooks etc. These tools can be your local development environment. After you complete your work, push the changes into the remote repo, and sync the workspace to upload the changes. Just make sure you're working with the supported structure of the item you're authoring. If you're not

sure, first clone a repo with content already synced to a workspace, then start authoring from there, where the structure is already in place.

Managing workspaces and branches

Since a workspace can only be connected to a single branch at a time, it's recommended to treat this as a 1:1 mapping. However, to reduce the amount of workspace it entails, consider these options:

- If a developer set up a private workspace with all required configurations, they can continue to use that workspace for any future branch they create. When a sprint is over, your changes are merged and you're starting a fresh new task, just switch the connection to a new branch on the same workspace. You can also do this if you suddenly need to fix a bug in the middle of a sprint. Think of it as a working directory on the web.

- Developers using a client tool (such as VS Code, Power BI Desktop, or others), don't necessarily need a workspace. They can create branches and commit changes to that branch locally, push those to the remote repo and create a pull request to the main branch, all without a workspace. A workspace is needed only as a testing environment to check that everything works in a real-life scenario. It's up to you to decide when that should happen.

Duplicate an item in a Git repository

To duplicate an item in a Git repository:

1. Copy the entire item directory.
2. Change the logicalId to something unique for that connected workspace.
3. Change the display name to differentiate it from the original item and to avoid duplicate display name error.
4. If necessary, update the logicalId and/or display names in any dependencies.

Best practices for deployment pipelines test stage

This section provides guidance for working with a deployment pipelines test stage.

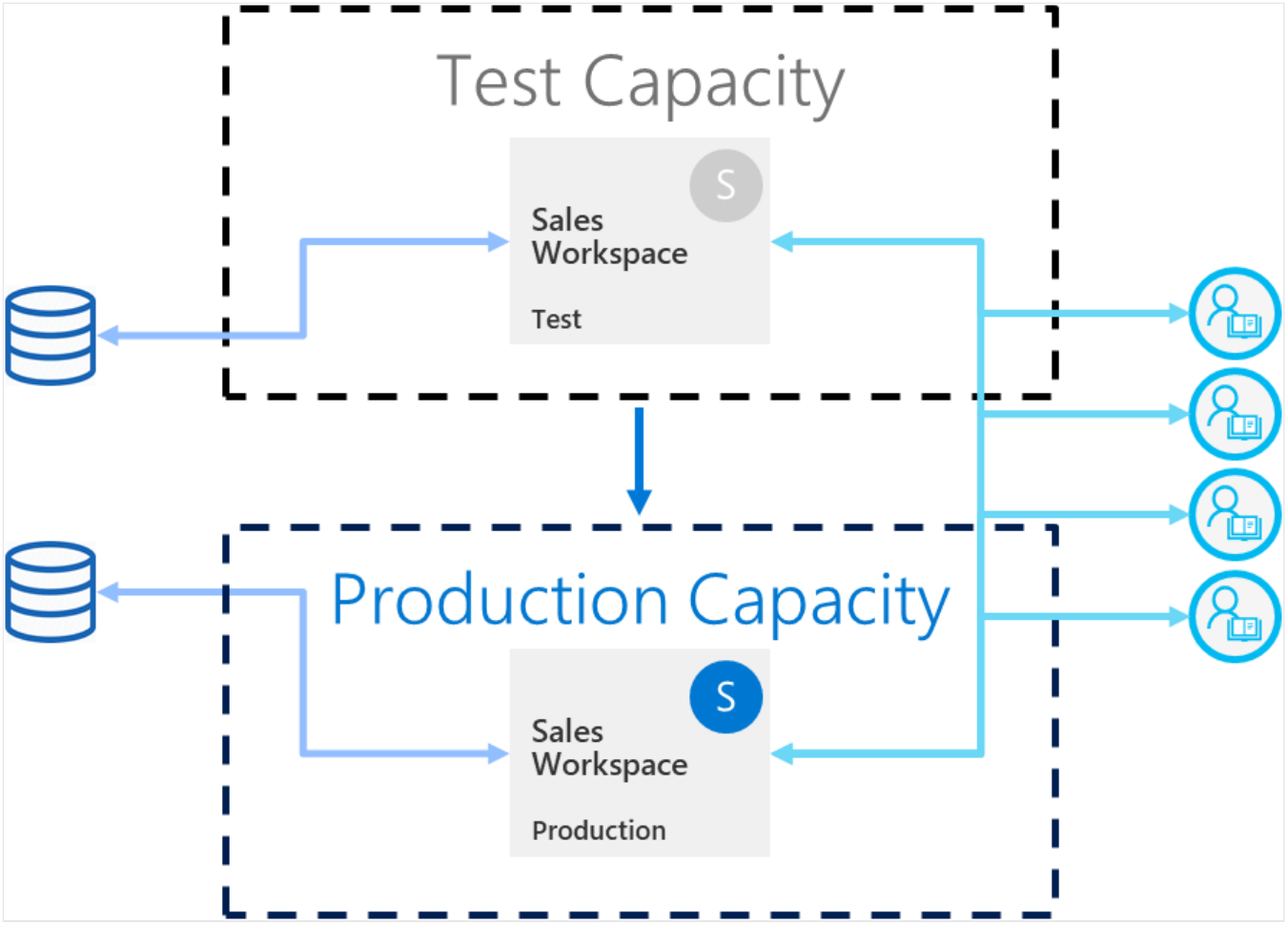
Simulate your production environment

It's important to see how your proposed change will impact the production stage. A deployment pipelines test stage allows you to simulate a real production environment for testing purposes. Alternatively, you can simulate this by connecting Git to another workspace.

Make sure that these three factors are addressed in your test environment:

- Data volume
- Usage volume
- A similar capacity as in production

When testing, you can use the same capacity as the production stage. However, using the same capacity can make production unstable during load testing. To avoid unstable production, test using a different capacity similar in resources to the production capacity. To avoid extra costs, use a capacity where you can pay only for the testing time.



Use deployment rules with a real-life data source

If you're using the test stage to simulate real life data usage, it's best to separate the development and test data sources. The development database should be relatively small, and the test database should be as similar as possible to the production database. Use data source rules to switch data sources in the test stage or parameterize the connection if not working through deployment pipelines.

Check related items

Changes you make can also affect the dependent items. During testing, verify that your changes don't affect or break the performance of existing items, which can be dependent on the updated ones.

You can easily find the related items by using impact analysis.

Updating data items

Data items are items that store data. The item's definition in Git defines how the data is stored. When updating an item in the workspace, we're importing its definition into the workspace and applying it on the existing data. The operation of updating data items is the same for Git and deployment pipelines.

As different items have different capabilities when it comes to retaining data when changes to the definition are applied, be mindful when applying the changes. Some practices that can help you apply the changes in the safest way:

- Know in advance what the changes are and what their impact might be on the existing data. Use commit messages to describe the changes made.
- To see how that item handles the change with test data, upload the changes first to a dev or test environment.
- If everything goes well, it's recommended to also check it on a staging environment, with real-life data (or as close to it as possible), to minimize the unexpected behaviors in production.
- Consider the best timing when updating the Prod environment to minimize the damage that any errors might cause to your business users who consume the data.
- After deployment, post-deployment tests in Prod to verify that everything is working as expected.
- Some changes will always be considered *breaking changes*. Hopefully, the preceding steps will help you track them before production. Build a plan for how to apply the changes in Prod and recover the data to get back to normal state and minimize downtime for business users.

Test your app

If you're distributing content to your customers through an app, review the app's new version *before* it's in production. Since each deployment pipeline stage has its own workspace, you can

easily publish and update apps for development and test stages. Publishing and updating apps allows you to test the app from an end user's point of view.

> **Important**
>
> The deployment process doesn't include updating the app content or settings. To apply changes to content or settings, manually update the app in the required pipeline stage.

Best practices for deployment pipelines productions stage

This section provides guidance to the deployment pipelines production stage.

Manage who can deploy to production

Because deploying to production should be handled carefully, it's good practice to let only specific people manage this sensitive operation. However, you probably want all BI creators for a specific workspace to have access to the pipeline. Use production workspace permissions to manage access permissions. Other users can have a production workspace *viewer* role to see content in the workspace but not make changes from Git or deployment pipelines.

In addition, limit access to the repo or pipeline by only enabling permissions to users that are part of the content creation process.

Set rules to ensure production stage availability

Deployment rules are a powerful way to ensure the data in production is always connected and available to users. With deployment rules applied, deployments can run while you have the assurance that customers can see the relevant information without disturbance.

Make sure that you set production deployment rules for data sources and parameters defined in the semantic model.

Update the production app

Deployment in a pipeline through the UI updates the workspace content. To update the associated app, use the deployment pipelines API. It's not possible to update the app through the UI. If you use an app for content distribution, don’t forget to update the app after deploying to production so that end users are immediately able to use the latest version.

Deploying into production using Git branches

As the repo serves as the 'single-source-of-truth', some teams might want to deploy updates into different stages directly from Git. This is possible with Git integration, with a few considerations:

- We recommend using release branches. You need to continuously change the connection of workspace to the new release branches before every deployment.
- If your build or release pipeline requires you to change the source code, or run scripts in a build environment before deployment to the workspace, then connecting the workspace to Git won't help you.
- After deploying to each stage, make sure to change all the configuration specific to that stage.

Quick fixes to content

Sometimes there are issues in production that require a quick fix. Deploying a fix without testing it first is bad practice. Therefore, always implement the fix in the development stage and push it to the rest of the deployment pipeline stages. Deploying to the development stage allows you to check that the fix works before deploying it to production. Deploying across the pipeline takes only a few minutes.

If you're using deployment from Git, we recommend following the practices described in Adopt a Git branching strategy.

Related content

- End to end lifecycle management tutorial
- Get started with Git integration
- Get started with deployment pipelines

Access the Power BI activity log

Article • 12/30/2024

This article targets Fabric administrators who need to access and analyze data sourced from the Power BI activity log. It focuses on the programmatic retrieval of Power BI activities by using the Get-PowerBIActivityEvent cmdlet from the Power BI Management module. Up to four weeks of history is available. This cmdlet uses the Get Activity Events Power BI REST API operation, which is an admin API. PowerShell cmdlets add a layer of abstraction on top of the underlying APIs. Therefore, the PowerShell cmdlet simplifies access to the Power BI activity log.

There are other manual and programmatic ways to retrieve Power BI activities. For more information, see the Access user activity data.

Analyzing the Power BI activity log is crucial for governance, compliance, and to track adoption efforts. For more information about the Power BI activity log, see Track user activities in Power BI.

> **Tip**
>
> We recommend that you fully review the **Tenant-level auditing** article. This article covers planning, key decisions, prerequisites, and key solution development activities to consider when building an end-to-end auditing solution.

Examples available

The goal of this article is to provide you with examples to help get you started. The examples include scripts that retrieve data from the activity log by using the Power BI Management PowerShell module.

> **⚠ Warning**
>
> The scripts aren't production-ready because they're intended only for educational purposes. You can, however, adapt the scripts for production purposes by adding logic for logging, error handling, alerting, and refactoring for code reuse and modularization.

Because they're intended for learning, the examples are simplistic, yet they're real-world. We recommend that you review all examples to understand how they apply slightly

different techniques. Once you identify the type of activity data that you need, you can mix and match the techniques to produce a script that best suits your requirements.

This article includes the following examples.

Expand table

Example name	Type of activity data
Authenticate with the Power BI service	N/A
View all activities for a user for one day	All
View an activity for N days	Share report (link or direct access)
View three activities for N days	Create app, update app, and install app
View all activities for a workspace for one day	All
Export all activities for the previous N days	All

For simplicity, most of the examples output their result to the screen. For instance, in Visual Studio Code, the data is output to the terminal panel, which holds a buffer set of data in memory.

Most of the examples retrieve raw JSON data. Working with the raw JSON data has many advantages.

- All of the information that's available for each activity event is returned. That's helpful for you to learn what data is available. Keep in mind that the contents of an API response differs depending on the actual activity event. For example, the data available for a *CreateApp* event is different to the *ViewReport* event.
- Because data that's available in the activity log changes as Power BI evolves over time, you can expect the API responses to change too. That way, new data that's introduced won't be missed. Your process is also more resilient to change and less likely to fail.
- The details of an API response can differ for the Power BI commercial cloud and the national/regional clouds.
- If you have different team members (such as data engineers) who get involved with this process, simplifying the initial process to extract the data makes it easier for multiple teams to work together.

Tip

We recommend that you keep your scripts that extract data as simple as possible. Therefore, avoid parsing, filtering, or formatting the activity log data as it's extracted. This approach uses an *ELT* methodology, which has separate steps to Extract, Load, and Transform data. This article only focuses on the first step, which is concerned with extracting the data.

Requirements

To use the example scripts, you must meet the following requirements.

- **PowerShell client tool**: Use your preferred tool for running PowerShell commands. All examples were tested by using the PowerShell extension for Visual Studio Code with PowerShell 7. For information about client tools and PowerShell versions, see Tenant-level auditing.
- **Power BI Management module**: Install all Power BI PowerShell modules. If you previously installed them, we recommend that you update the modules to ensure that you're using the latest published version.
- **Fabric administrator role**: The example scripts are designed to use an interactive authentication flow. Therefore, the user running the PowerShell example scripts must sign in to use the Power BI REST APIs. To retrieve activity log data, the authenticating user must belong to the Fabric administrator role (because retrieving activity events is done with an admin API). Service principal authentication is out of scope for these learning examples.

The remainder of this article includes sample scripts that show you different ways to retrieve activity log data.

Example 1: Authenticate with the Power BI service

All Power BI REST API operations require you to sign in. *Authentication* (who is making the request) and *authorization* (what the user has permission to do) are managed by the Microsoft identity platform. The following example uses the Connect-PowerBIServiceAccount cmdlet from the Power BI Management module. This cmdlet supports a simple method to sign in.

Sample request 1

The first script redirects you to a browser to complete the sign in process. User accounts that have multi-factor authentication (MFA) enabled are able to use this interactive authentication flow to sign in.

PowerShell

```
Connect-PowerBIServiceAccount
```

Important

Users without Fabric administrator privileges can't run any of the sample scripts that follow in this article. Fabric administrators have permission to manage the Power BI service and to retrieve tenant-wide metadata (such as activity log data). Although using service principal authentication is out of scope for these examples, we strongly recommend that you set up a service principal for production-ready, unattended scripts that will run on a schedule.

Be sure to sign in before running any of the following scripts.

Example 2: View all activities for a user for one day

Sometimes you need to check all the activities that a specific user performed on a specific day.

Tip

When extracting data from the activity log by using the PowerShell cmdlet, each request can extract data for one day (a maximum of 24 hours). Therefore, the goal of this example is to start simply by checking one user for one day. There are other examples later in this article that show you how to use a loop to export data for multiple days.

Sample request 2

This script declares two PowerShell variables to make it easier to reuse the script:

- `$UserEmailAddr`: The email address for the user you're interested in.

- `$ActivityDate`: The date you're interested in. The format is YYYY-MM-DD (ISO 8601 format). You can't request a date earlier than 28 days before the current date.

PowerShell

```
#Input values before running the script:
$UserEmailAddr = 'jordan@contoso.com'
$ActivityDate = '2023-03-15'
#---------------------------------------------------------------------
#View activity events:
Get-PowerBIActivityEvent `
    -StartDateTime ($ActivityDate + 'T00:00:00.000') `
    -EndDateTime ($ActivityDate + 'T23:59:59.999') `
    -User $UserEmailAddr
```

> **Note**
>
> You might notice a backtick (`) character at the end of some of the lines in the PowerShell scripts. In PowerShell, one way you can use the backtick character is as a line continuation character. We've used it to improve the readability of the scripts in this article.

> **Tip**
>
> In the script, each of the **PowerShell variables** correlate to a required or optional **parameter** value in the **Get-PowerBIActivityEvent** cmdlet. For example, the value you assign to the `$UserEmailAddr` variable is passed to the `-User` parameter. Declaring PowerShell variables in this way is a lightweight approach to avoid hard-coding values that could change in your script. That's a good habit to adopt, and it will be useful as your scripts become more complex. **PowerShell parameters** are more robust than variables, but they're out of scope for this article.

Sample response 2

Here's a sample JSON response. It includes two activities that the user performed:

- The first activity shows that a user viewed a report.
- The second activity shows that an administrator exported data from the Power BI activity log.

JSON

```
[
  {
    "Id": "10af656b-b5a2-444c-bf67-509699896daf",
    "RecordType": 20,
    "CreationTime": "2023-03-15T15:18:30Z",
    "Operation": "ViewReport",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 0,
    "UserKey": "100FFF92C7717B",
    "Workload": "PowerBI",
    "UserId": "jordan@contoso.com",
    "ClientIP": "192.168.1.1",
    "Activity": "ViewReport",
    "ItemName": "Gross Margin Analysis",
    "WorkSpaceName": "Sales Analytics",
    "DatasetName": "Sales Data",
    "ReportName": "Gross Margin Analysis",
    "WorkspaceId": "e380d1d0-1fa6-460b-9a90-1a5c6b02414c",
    "ObjectId": "Gross Margin Analysis",
    "DatasetId": "cfafbeb1-8037-4d0c-896e-a46fb27ff229",
    "ReportId": "94e57e92-Cee2-486d-8cc8-218c97200579",
    "ArtifactId": "94e57e92-Cee2-486d-8cc8-218c97200579",
    "ArtifactName": "Gross Margin Analysis",
    "IsSuccess": true,
    "ReportType": "PowerBIReport",
    "RequestId": "53451b83-932b-f0b0-5328-197133f46fa4",
    "ActivityId": "beb41a5d-45d4-99ee-0e1c-b99c451e9953",
    "DistributionMethod": "Workspace",
    "ConsumptionMethod": "Power BI Web",
    "SensitivityLabelId": "e3dd4e72-5a5d-4a95-b8b0-a0b52b827793",
    "ArtifactKind": "Report"
  },
  {
    "Id": "5c913f29-502b-4a1a-a089-232edaf176f7",
    "RecordType": 20,
    "CreationTime": "2023-03-15T17:22:00Z",
    "Operation": "ExportActivityEvents",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 2,
    "UserKey": "100FFF92C7717B",
    "Workload": "PowerBI",
    "UserId": "jordan@contoso.com",
    "ClientIP": "192.168.1.1",
    "UserAgent": "MicrosoftPowerBIMgmt/1.2.1111.0",
    "Activity": "ExportActivityEvents",
    "IsSuccess": true,
    "RequestId": "2af6a22d-6f24-4dc4-a26a-5c234ab3afad",
    "ActivityId": "00000000-0000-0000-0000-000000000000",
    "ExportEventStartDateTimeParameter": "2023-03-17T00:00:00Z",
    "ExportEventEndDateTimeParameter": "2023-03-17T23:59:59.999Z"
  }
]
```

> **Tip**
>
> Extracting the Power BI activity log data is also a logged operation, as shown in the previous response. When you analyze user activities, you might want to omit administrator activities—or analyze them separately.

Example 3: View an activity for N days

Sometimes you might want to investigate one specific type of activity for a series of days. This example shows how to retrieve per-item report sharing activities. It uses a loop to retrieve activities from the previous seven days.

Sample request 3

The script declares two variables:

- `$ActivityType`: The operation name for the activity that you're investigating.
- `$NbrOfDaysToCheck`: How many days you're interested in checking. It performs a loop working backward from the current day. The maximum value allowed is 28 days (because the earliest date that you can retrieve is 28 days before the current day).

PowerShell

```
#Input values before running the script:
$ActivityType = 'ShareReport'
$NbrOfDaysToCheck = 7
#-----------------------------------------------------------------------

#Use today to start counting back the number of days to check:
$DayUTC = (([datetime]::Today.ToUniversalTime()).Date)

#Iteratively loop through each of the last N days to view events:
For($LoopNbr=0; $LoopNbr -le $NbrOfDaysToCheck; $LoopNbr++)
{
    $PeriodStart=$DayUTC.AddDays(-$LoopNbr)
    $ActivityDate=$PeriodStart.ToString("yyyy-MM-dd")
    Write-Verbose "Checking $ActivityDate" -Verbose

    #Check activity events once per loop (once per day):
    Get-PowerBIActivityEvent `
        -StartDateTime ($ActivityDate + 'T00:00:00.000') `
        -EndDateTime ($ActivityDate + 'T23:59:59.999') `
        -ActivityType $ActivityType
}
```

> **Tip**
>
> You can use this looping technique to check any of the **operations** recorded in the activity log.

Sample response 3

Here's a sample JSON response. It includes two activities that the user performed:

- The first activity shows that a sharing link for a user was created. Note that the *SharingAction* value differs depending on whether the user created a link, edited a link, or deleted a link. For brevity, only one type of sharing link activity is shown in the response.
- The second activity shows that direct access sharing for a group was created. Note that the *SharingInformation* value differs depending on the action taken. For brevity, only one type of direct access sharing activity is shown in the response.

JSON


```
[
  {
    "Id": "be7506e1-2bde-4a4a-a210-bc9b156142c0",
    "RecordType": 20,
    "CreationTime": "2023-03-15T19:52:42Z",
    "Operation": "ShareReport",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 0,
    "UserKey": "900GGG12D2242A",
    "Workload": "PowerBI",
    "UserId": "morgan@contoso.com",
    "ClientIP": "192.168.1.1",
    "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/110.0",
    "Activity": "ShareReport",
    "ItemName": "Call Center Stats",
    "WorkSpaceName": "Sales Analytics",
    "SharingInformation": [
      {
        "RecipientEmail": "ellis@contoso.com",
        "RecipientName": "Turner",
        "ObjectId": "fc9bbc6c-e39b-44cb-9c8a-d37d5665ec57",
        "ResharePermission": "ReadReshare",
        "UserPrincipalName": "ellis@contoso.com"
      }
    ],
    "WorkspaceId": "e380d1d0-1fa6-460b-9a90-1a5c6b02414c",
    "ObjectId": "Call Center Stats",
    "Datasets": [
```

```
      {
        "DatasetId": "fgagrwa3-9044-3e1e-228f-k24bf72gg995",
        "DatasetName": "Call Center Data"
      }
    ],
    "ArtifactId": "81g22w11-vyy3-281h-1mn3-822a99921541",
    "ArtifactName": "Call Center Stats",
    "IsSuccess": true,
    "RequestId": "7d55cdd3-ca3d-a911-5e2e-465ac84f7aa7",
    "ActivityId": "4b8b53f1-b1f1-4e08-acdf-65f7d3c1f240",
    "SharingAction": "CreateShareLink",
    "ShareLinkId": "J_5UZg-36m",
    "ArtifactKind": "Report",
    "SharingScope": "Specific People"
  },
  {
    "Id": "b4d567ac-7ec7-40e4-a048-25c98d9bc304",
    "RecordType": 20,
    "CreationTime": "2023-03-15T11:57:26Z",
    "Operation": "ShareReport",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 0,
    "UserKey": "900GGG12D2242A",
    "Workload": "PowerBI",
    "UserId": "morgan@contoso.com",
    "ClientIP": "69.132.26.0",
    "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0",
    "Activity": "ShareReport",
    "ItemName": "Gross Margin Analysis",
    "WorkSpaceName": "Sales Analytics",
    "SharingInformation": [
      {
        "RecipientName": "SalesAndMarketingGroup-NorthAmerica",
        "ObjectId": "ba21f28b-6226-4296-d341-f059257a06a7",
        "ResharePermission": "Read"
      }
    ],
    "CapacityId": "1DB44EEW-6505-4A45-B215-101HBDAE6A3F",
    "CapacityName": "Shared On Premium - Reserved",
    "WorkspaceId": "e380d1d0-1fa6-460b-9a90-1a5c6b02414c",
    "ObjectId": "Gross Margin Analysis",
    "Datasets": [
      {
        "DatasetId": "cfafbeb1-8037-4d0c-896e-a46fb27ff229",
        "DatasetName": "Sales Data"
      }
    ],
    "ArtifactId": "94e57e92-Cee2-486d-8cc8-218c97200579",
    "ArtifactName": "Gross Margin Analysis",
    "IsSuccess": true,
    "RequestId": "82219e60-6af0-0fa9-8599-c77ed44fff9c",
    "ActivityId": "1d21535a-257e-47b2-b9b2-4f875b19855e",
    "SensitivityLabelId": "16c065f5-ba91-425e-8693-261e40ccdbef",
    "SharingAction": "Direct",
```

```
    "ArtifactKind": "Report",
    "SharingScope": "Specific People"
  }
]
```

> ⓘ **Note**
>
> This JSON response shows that the data structure is different based on the type of event. Even the same type of event can have different characteristics that produce a slightly different output. As recommended earlier in this article, you should get accustomed to retrieving the raw data.

Example 4: View three activities for N days

Sometimes you might want to investigate several related activities. This example shows how to retrieve three specific activities for the previous seven days. It focuses on activities related to Power BI apps including creating an app, updating an app, and installing an app.

Sample request 4

The script declares the following variables:

- `$NbrOfDaysToCheck`: How many days you're interested in checking. It performs a loop that works backward from the current day. The maximum value allowed is 28 days (because the earliest date that you can retrieve is 28 days before the current day).
- `$Activity1`: The operation name for the first activity that you're investigating. In this example, it's searching for Power BI app creation activities.
- `$Activity2`: The second operation name. In this example, it's searching for Power BI app update activities.
- `$Activity3`: The third operation name. In this example, it's searching for Power BI app installation activities.

You can only retrieve activity events for one activity at a time. So, the script searches for each operation separately. It combines the search results into a variable named `$FullResults`, which it then outputs to the screen.

> ⊗ **Caution**

Running many loops many times greatly increases the likelihood of API *throttling*. Throttling can happen when you exceed the number of requests you're allowed to make in a given time period. The **Get Activity Events** operation is limited to 200 requests per hour. When you design your scripts, take care not to retrieve the original data more times than you need. Generally, it's a better practice to extract all of the raw data once per day and then query, transform, filter, or format that data separately.

The script shows results for the current day.

> **Note**
>
> To retrieve results for the previous day only—avoiding partial day results—see the Export all activities for previous N days example.)

PowerShell

```
#Input values before running the script:
$NbrOfDaysToCheck = 7
$Activity1 = 'CreateApp'
$Activity2 = 'UpdateApp'
$Activity3 = 'InstallApp'
#-----------------------------------------------------------------------
#Initialize array which will contain the full resultset:
$FullResults = @()

#Use today to start counting back the number of days to check:
$DayUTC = (([datetime]::Today.ToUniversalTime()).Date)

#Iteratively loop through each day (<Initilize> ; <Condition> ; <Repeat>)
#Append each type of activity to an array:
For($LoopNbr=0; $LoopNbr -le $NbrOfDaysToCheck; $LoopNbr++)
{
    $PeriodStart=$DayUTC.AddDays(-$LoopNbr)
    $ActivityDate=$PeriodStart.ToString("yyyy-MM-dd")
    Write-Verbose "Checking $ActivityDate" -Verbose

    #Get activity 1 and append its results into the full resultset:
    $Activity1Results = @()
    $Activity1Results += Get-PowerBIActivityEvent `
        -StartDateTime ($ActivityDate+'T00:00:00.000') `
        -EndDateTime ($ActivityDate+'T23:59:59.999') `
        -ActivityType $Activity1 | ConvertFrom-Json
    If ($null -ne $Activity1Results) {$FullResults += $Activity1Results}

    #Get activity 2 and append its results into the full resultset:
    $Activity2Results = @()
    $Activity2Results += Get-PowerBIActivityEvent `
```

```
        -StartDateTime ($ActivityDate+'T00:00:00.000') `
        -EndDateTime ($ActivityDate+'T23:59:59.999') `
        -ActivityType $Activity2 |
    ConvertFrom-Json
    If ($null -ne $Activity2Results) {$FullResults += $Activity2Results}

    #Get activity 3 and append its results into the full resultset:
    $Activity3Results = @()
    $Activity3Results += Get-PowerBIActivityEvent `
        -StartDateTime ($ActivityDate+'T00:00:00.000') `
        -EndDateTime ($ActivityDate+'T23:59:59.999') `
        -ActivityType $Activity3 |
    ConvertFrom-Json
    If ($null -ne $Activity3Results) {$FullResults += $Activity3Results}

}
#Convert all of the results back to a well-formed JSON object:
$FullResults = $FullResults | ConvertTo-Json

#Display results on the screen:
$FullResults
```

Sample response 4

Here's a sample JSON response. It includes three activities that the user performed:

- The first activity shows a Power BI app was created.
- The second activity shows that a Power BI app was updated.
- The third activity shows that a Power BI app was installed by a user.

> ⚠ **Warning**
>
> The response only includes the user permissions that were modified. For example, it's possible that three audiences could've been created in a *CreateApp* event. In the *UpdateApp* event, if only one audience changed, then only one audience would appear in the *OrgAppPermission* data. For that reason, relying on the *UpdateApp* event for tracking all app permissions is incomplete because the activity log only shows what's changed.
>
> For a snapshot of all Power BI app permissions, use the Get App Users as Admin API operation instead.

JSON


```
[
  {
    "Id": "65a26480-981a-4905-b3aa-cbb3df11c7c2",
```

```
    "RecordType": 20,
    "CreationTime": "2023-03-15T18:42:13Z",
    "Operation": "CreateApp",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 0,
    "UserKey": "100FFF92C7717B",
    "Workload": "PowerBI",
    "UserId": "jordan@contoso.com",
    "ClientIP": "192.168.1.1",
    "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0",
    "Activity": "CreateApp",
    "ItemName": "Sales Reconciliations App",
    "WorkSpaceName": "Sales Reconciliations",
    "OrgAppPermission": {
      "recipients": "Sales Reconciliations App(Entire Organization)",
      "permissions": "Sales Reconciliations App(Read,CopyOnWrite)"
    },
    "WorkspaceId": "9325a31d-067e-4748-a592-626d832c8001",
    "ObjectId": "Sales Reconciliations App",
    "IsSuccess": true,
    "RequestId": "ab97a4f1-9f5e-4a6f-5d50-92c837635814",
    "ActivityId": "9bb54a9d-b688-4028-958e-4d7d21ca903a",
    "AppId": "42d60f97-0f69-470c-815f-60198956a7e2"
  },
  {
    "Id": "a1dc6d26-b006-4727-bac6-69c765b7978f",
    "RecordType": 20,
    "CreationTime": "2023-03-16T18:39:58Z",
    "Operation": "UpdateApp",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 0,
    "UserKey": "100GGG12F9921B",
    "Workload": "PowerBI",
    "UserId": "morgan@contoso.com",
    "ClientIP": "192.168.1.1",
    "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0",
    "Activity": "UpdateApp",
    "ItemName": "Sales Analytics",
    "WorkSpaceName": "Sales Analytics",
    "OrgAppPermission": {
      "recipients": "Sales Reps Audience(SalesAndMarketingGroup-
NorthAmerica,SalesAndMarketingGroup-Europe)",
      "permissions": "Sales Reps Audience(Read,CopyOnWrite)"
    },
    "WorkspaceId": "c7bffcd8-8156-466a-a88f-0785de2c8b13",
    "ObjectId": "Sales Analytics",
    "IsSuccess": true,
    "RequestId": "e886d122-2c09-4189-e12a-ef998268b864",
    "ActivityId": "9bb54a9d-b688-4028-958e-4d7d21ca903a",
    "AppId": "c03530c0-db34-4b66-97c7-34dd2bd590af"
  },
  {
    "Id": "aa002302-313d-4786-900e-e68a6064df1a",
```

```
    "RecordType": 20,
    "CreationTime": "2023-03-17T18:35:22Z",
    "Operation": "InstallApp",
    "OrganizationId": "927c6607-8060-4f4a-a5f8-34964ac78d70",
    "UserType": 0,
    "UserKey": "100HHH12F4412A",
    "Workload": "PowerBI",
    "UserId": "ellis@contoso.com",
    "ClientIP": "192.168.1.1",
    "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0",
    "Activity": "InstallApp",
    "ItemName": "Sales Reconciliations App",
    "ObjectId": "Sales Reconciliations App",
    "IsSuccess": true,
    "RequestId": "7b3cc968-883f-7e13-081d-88b13f6cfbd8",
    "ActivityId": "9bb54a9d-b688-4028-958e-4d7d21ca903a"
  }
]
```


Example 5: View all activities for a workspace for one day

Sometimes you might want to investigate activities related to a specific workspace. This example retrieves all activities for all users for one day. It then filters the results so that you can focus on analyzing activities from one workspace.

Sample request 5

The script declares two variables:

- `$ActivityDate`: The date you're interested in. The format is YYYY-MM-DD. You can't request a date earlier than 28 days before the current date.
- `$WorkspaceName`: The name of the workspace you're interested in.

The script stores the results in the `$Results` variable. It then converts the JSON data to an object so the results can be parsed. It then filters the results to retrieve five specific columns. The *CreationTime* data is renamed as *ActivityDateTime*. The results are filtered by the workspace name, then output to the screen.

There isn't a parameter for the Get-PowerBIActivityEvent cmdlet that allows you to specify a workspace when checking the activity log (earlier examples in this article used PowerShell parameters to set a specific user, date, or activity name). In this example, the script retrieves all of the data and then parses the JSON response to filter the results for a specific workspace.

⊗ **Caution**

If you're in a large organization that has hundreds or thousands of activities per day, filtering the results after they've been retrieved can be very inefficient. Bear in mind that the **Get Activity Events** operation is limited to 200 requests per hour.

To avoid API throttling (when you exceed the number of requests that you're allowed to make in a given time period), don't retrieve the original data more than you need to. You can continue to work with the filtered results without running the script to retrieve the results again. For ongoing needs, it's a better practice to extract all of the data once per day and then query it many times.

PowerShell

```
#Input values before running the script:
$ActivityDate = '2023-03-22'
$WorkspaceName = 'Sales Analytics'
#----------------------------------------------------------------------
#Run cmdlet to check activity events and store intermediate results:
$Events = Get-PowerBIActivityEvent `
    -StartDateTime ($ActivityDate+'T00:00:00.000') `
    -EndDateTime ($ActivityDate+'T23:59:59.999')

#Convert from JSON so we can parse the data:
$ConvertedResults = $Events | ConvertFrom-Json

#Obtain specific attributes and save to a PowerShell object:
$FilteredResults = $ConvertedResults `
    |
    Select-Object `
    @{Name="ActivityDateTime";Expression={$PSItem.CreationTime}}, ` #alias
name
    Activity, `
    UserId, `
    ArtifactName, `
    WorkspaceName `
    |
    #Filter the results:
    Where-Object {($PSItem.WorkspaceName -eq $WorkspaceName)}

#View the filtered results:
$FilteredResults

#Optional - Save back to JSON format:
#$FilteredResults = $FilteredResults | ConvertTo-Json -Depth 10
#$FilteredResults
```

Sample response 5

Here are the filtered results, which include a small subset of properties. The format is easier to read for occasional analysis. However, we recommend that you convert it back to JSON format if you plan to store the results.

> **Note**
>
> After converting the JSON results to a PowerShell object, time values are converted to local time. The original audit data is always recorded in Coordinated Universal Time (UTC) time, so we recommend that you get accustomed to using only UTC time.

Output

```
ActivityDateTime : 4/25/2023 3:18:30 PM
Activity         : ViewReport
UserId           : jordan@contoso.com
ArtifactName     : Gross Margin Analysis
WorkSpaceName    : Sales Analytics

CreationTime     : 4/25/2023 5:32:10 PM
Activity         : ShareReport
UserId           : ellis@contoso.com
ArtifactName     : Call Center Stats
WorkSpaceName    : Sales Analytics

CreationTime     : 4/25/2023 9:03:05 PM
Activity         : ViewReport
UserId           : morgan@contoso.com
ArtifactName     : Call Center Stats
WorkSpaceName    : Sales Analytics
```

> **Tip**
>
> You can use this technique to filter results by any property in the results. For example, you can use a specific event `RequestId` to analyze just one specific event.

Example 6: Export all activities for previous N days

Sometimes you might want to export all activity data to a file so that you can work with the data outside of PowerShell. This example retrieves all activities for all users for up to 28 days. It exports the data to one JSON file per day.

> **Important**
>
> Activity log data is available for a maximum of 28 days. It's important that you export and retain the data so you can do historical analysis. If you don't currently export and store the activity log data, we strongly recommend that you prioritize doing so.

Sample request 6

The script retrieves all activities for a series of days. It declares three variables:

- `$NbrDaysDaysToExtract`: How many days you're interested in exporting. It performs a loop, working backward from the previous day. The maximum value allowed is 28 days (because the earliest date that you can retrieve is 28 days before the current day).
- `$ExportFileLocation`: The folder path where you want to save the files. The folder must exist before running the script. Don't include a backslash (\) character at the end of the folder path (because it's automatically added at runtime). We recommend that you use a separate folder to store raw data files.
- `$ExportFileName`: The prefix for each file name. Because one file per day is saved, the script adds a suffix to indicate the data contained in the file, and the date and time the data was retrieved. For example, if you ran a script at 9am (UTC) on April 25, 2023 to extract activity data for April 23, 2023, the file name would be: PBIActivityEvents-20230423-202304250900. Although the folder structure where it's stored is helpful, each file name should be fully self-describing.

We recommend that you extract data that's at least one day before the current day. That way, you avoid retrieving partial day events, and you can be confident that each export file contains the full 24 hours of data.

The script gathers up to 28 days of data, through to the previous day. Timestamps for audited events are always in UTC. We recommend that you build all of your auditing processes based on UTC time rather than your local time.

The script produces one JSON file per day. The suffix of the file name includes the timestamp (in UTC format) of the extracted data. If you extract the same day of data more than once, the suffix in the file name helps you identify the newer file.

PowerShell

```
#Input values before running the script:
$NbrDaysDaysToExtract = 7
```

```
$ExportFileLocation = 'C:\Power-BI-Raw-Data\Activity-Log'
$ExportFileName = 'PBIActivityEvents'
#-----------------------------------------

#Start with yesterday for counting back to ensure full day results are
obtained:
[datetime]$DayUTC = (([datetime]::Today.ToUniversalTime()).Date).AddDays(-1)

#Suffix for file name so we know when it was written:
[string]$DateTimeFileWrittenUTCLabel =
([datetime]::Now.ToUniversalTime()).ToString("yyyyMMddHHmm")

#Loop through each of the days to be extracted (<Initilize> ; <Condition> ;
<Repeat>)
For($LoopNbr=0 ; $LoopNbr -lt $NbrDaysDaysToExtract ; $LoopNbr++)
{
    [datetime]$DateToExtractUTC=$DayUTC.AddDays(-$LoopNbr).ToString("yyyy-
MM-dd")

    [string]$DateToExtractLabel=$DateToExtractUTC.ToString("yyyy-MM-dd")

    #Create full file name:
    [string]$FullExportFileName = $ExportFileName `
    + '-' + ($DateToExtractLabel -replace '-', '') `
    + '-' + $DateTimeFileWrittenUTCLabel `
    + '.json'

    #Obtain activity events and store intermediary results:
    [psobject]$Events=Get-PowerBIActivityEvent `
        -StartDateTime ($DateToExtractLabel+'T00:00:00.000') `
        -EndDateTime ($DateToExtractLabel+'T23:59:59.999')

    #Write one file per day:
    $Events | Out-File "$ExportFileLocation\$FullExportFileName"

    Write-Verbose "File written: $FullExportFileName" -Verbose
}
Write-Verbose "Extract of Power BI activity events is complete." -Verbose
```

There are several advantages to using the Get-PowerBIActivityEvent PowerShell cmdlet rather than the Get Activity Events REST API operation.

- The cmdlet allows you to request one day of activity each time you make a call by using the cmdlet. Whereas when you communicate with the API directly, you can only request one hour per API request.
- The cmdlet handles continuation tokens for you. If you use the API directly, you need to check the continuation token to determine whether there are any more results to come. Some APIs need to use pagination and continuation tokens for performance reasons when they return a large amount of data. They return the first set of records, then with a continuation token you can make a subsequent API

call to retrieve the next set of records. You continue calling the API until a continuation token isn't returned. Using the continuation token is a way to consolidate multiple API requests so that you can consolidate a logical set of results. For an example of using a continuation token, see Activity Events REST API.

- The cmdlet handles Microsoft Entra ID access token expirations for you. After you've authenticated, your access token expires after one hour (by default). In this case, the cmdlet automatically requests a refresh token for you. If you communicate with the API directly, you need to request a refresh token.

For more information, see Choose APIs or PowerShell cmdlets.

> **Note**
>
> A sample response is omitted because it's an output similar to the responses shown in the previous examples.

Related content

For more information related to this article, check out the following resources:

- Track user activities in Power BI
- Power BI implementation planning: Tenant-level auditing
- Fabric adoption roadmap: Auditing and monitoring
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI migration overview

Article • 12/30/2024

Customers are increasingly standardizing on Power BI to drive a data culture, which involves enabling managed self-service business intelligence (SSBI), rationalizing the delivery of enterprise BI, and addressing economic pressures. The purpose of this series of Power BI migration articles is to provide you with guidance on how to plan and conduct a migration from a third-party BI tool to Power BI.

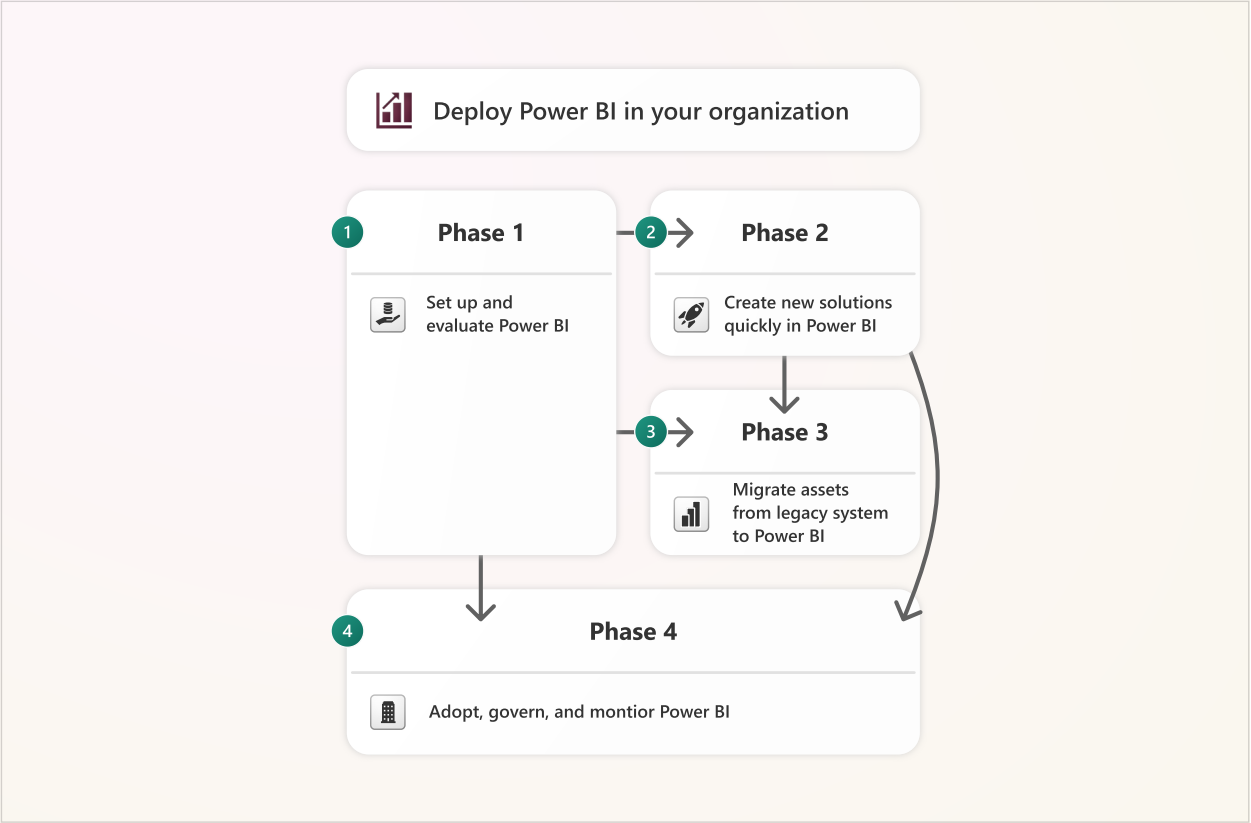
The articles in the Power BI migration series include:

1. Power BI migration overview (this article)
2. Prepare to migrate to Power BI
3. Gather requirements to migrate to Power BI (Stage 1)
4. Plan deployment to migrate to Power BI (Stage 2)
5. Conduct proof of concept to migrate to Power BI (Stage 3)
6. Create content to migrate to Power BI (Stage 4)
7. Deploy to Power BI (Stage 5)
8. Learn from customer Power BI migrations

> **Note**
>
> We also recommend that you thoroughly read the Fabric adoption roadmap and Power BI implementation planning articles.

There are two assumptions: Your organization has a legacy BI platform currently in place and the decision has been made to formally migrate content and users to Power BI. Migrating to the Power BI service is the primary focus of this series. Additional considerations could apply for national/regional cloud customers beyond what is discussed in this series of articles.

The following diagram shows four high-level phases for deploying Power BI in your organization.



Expand table

Phase	Description
1	**Set up and evaluate Power BI.** The first phase involves establishing the initial Power BI architecture. Preliminary deployment and governance planning are handled at this point, as well as Power BI evaluations including return on investment and/or cost benefit analysis.
2	**Create new solutions quickly in Power BI.** In the second phase, self-service BI authors can begin using and evaluating Power BI for their needs, and value can be obtained from Power BI quickly. Activities in Phase 2 place importance on agility and rapid business value, which is critical to gaining acceptance for the selection of a new BI tool such as Power BI. For this reason, the diagram depicts activities in Phase 2 happening side by side with the migration activities in Phase 3.
3	**Migrate BI assets from legacy platform to Power BI.** The third phase addresses the migration to Power BI. It's the focus of this series of Power BI migration articles. Five specific migration stages are discussed in the next section.
4	**Adopt, govern, and monitor Power BI.** The final phase comprises ongoing activities such as nurturing a data culture, communication, and training. These activities greatly impact on an effective Power BI implementation. It's important to have governance and security policies and processes that are appropriate for your organization, as well as auditing and monitoring to allow you to scale, grow, and continually improve.

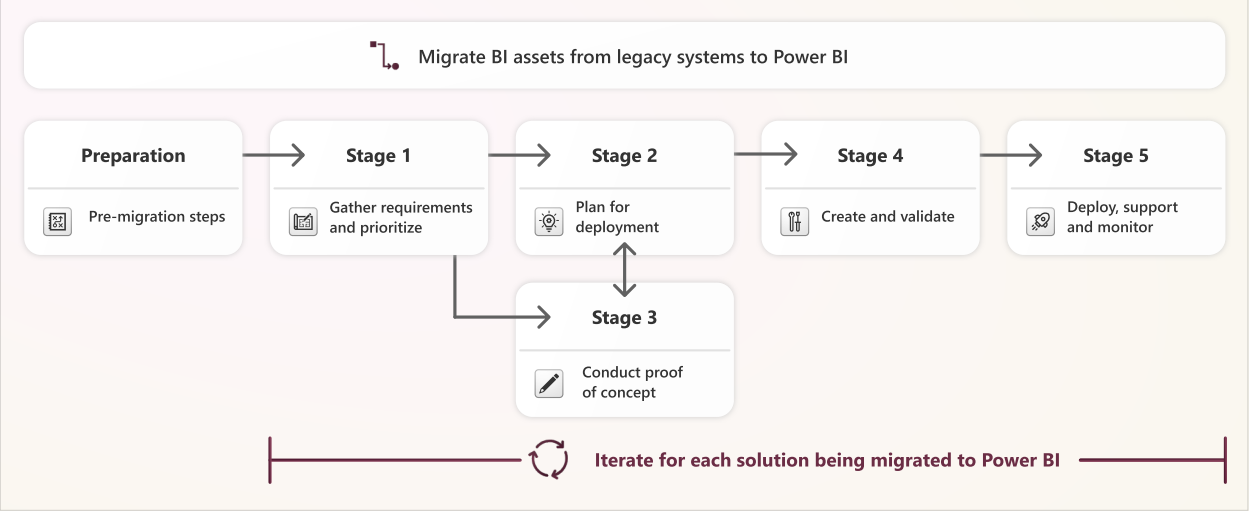
Important

A formal migration to Power BI almost always occurs in parallel with the development of a new Power BI solution. *Power BI solution* is generic term that encompasses the use of both data and reports. A single Power BI Desktop (pbix) file could contain a data model or report, or both. **Separating the data model from reports** is encouraged for data reusability purposes, but isn't required.

Using Power BI to author new requirements, while you plan and conduct the formal migration, will help gain buy-in. Simultaneous phases provide content authors with practical, real-world experience with Power BI.

Five stages of a Power BI migration

Phase 3 of the diagram addresses migration to Power BI. During this phase, there are five common stages.



The following stages shown in the previous diagram are:

- Pre-migration steps
- Stage 1: Gather requirements and prioritize
- Stage 2: Plan for deployment
- Stage 3: Conduct proof of concept
- Stage 4: Create and validate content
- Stage 5: Deploy, support, and monitor

Pre-migration steps

The pre-migration steps include actions you should consider prior to beginning a project to migrate content from a legacy BI platform to Power BI. It typically includes the initial tenant-level deployment planning. For more information about these activities, see Prepare to migrate to Power BI.

Stage 1: Gather requirements and prioritize

The emphasis of Stage 1 is on gathering information and planning for the migration of a single solution. This process should be iterative and scoped to a reasonable sized effort. The output for Stage 1 includes a prioritized inventory of reports and data that are to be migrated. Additional activities in Stages 2 and 3 are necessary to fully estimate the level of effort. For more information about the activities in Stage 1, see Gather requirements to migrate to Power BI.

Stage 2: Plan for deployment

The focus of Stage 2 is on how the requirements defined in Stage 1 are fulfilled for each specific solution. The output of Stage 2 includes as many specifics as possible to guide the process, though it is an iterative, non-linear process. Creation of a proof of concept (in Stage 3) can occur in parallel with this stage. Even while creating the solution (in Stage 4), additional information might come to light that influences deployment planning decisions. This type of deployment planning in Stage 2 focuses on the solution level, while respecting the decisions already made at the organizational level. For more information about the activities in Stage 2, see Plan deployment to migrate to Power BI.

Stage 3: Conduct proof of concept

The emphasis of Stage 3 is to address unknowns and mitigate risks as early as possible. A technical proof of concept (POC) is helpful for validating assumptions, and it can be done iteratively alongside deployment planning (Stage 2). The output of this stage is a Power BI solution that's narrow in scope. Note that we don't intend for the POC to be disposable work. However, it will likely require additional work in Stage 4 to make it production-ready. In this respect, in your organization, you might refer to this activity as either a prototype, pilot, mockup, quickstart, or minimally viable product (MVP). Conducting a POC isn't always necessary and it can be done informally. For more information about the activities in Stage 3, see
Conduct proof of concept to migrate to Power BI.

Stage 4: Create and validate content

Stage 4 is when the actual work to convert the POC to a production-ready solution is done. The output of this stage is a completed Power BI solution that's been validated in a development environment. It should be ready for deployment in Stage 5. For more information about the activities in Stage 4, see Create content to migrate to Power BI.

Stage 5: Deploy, support, and monitor

The primary focus of Stage 5 is to deploy the new Power BI solution to production. The output of this stage is a production solution that's actively used by business users. When using an agile methodology, it's acceptable to have some planned enhancements that will be delivered in a future iteration. Depending on your comfort level with Power BI, such as minimizing risk and user disruption, you might choose to do a staged deployment. Or, you might initially deploy to a smaller group of pilot users. Support and monitoring are also important at this stage, and on an ongoing basis. For more information about the activities in Stage 5, see Migrate to Power BI.

> **Tip**
>
> Most of the concepts discussed throughout this series of Power BI migration articles also apply to a standard Power BI implementation project.

Consider migration reasons

Enabling a productive and healthy data culture is a principal goal of many organizations. Power BI is an excellent tool to facilitate this objective. Three common reasons you might consider migrating to Power BI can be distilled down to:

- Enable managed self-service BI by introducing new capabilities that empower the self-service BI user community. Power BI makes access to information and decision-making more broadly available, while relying less on specialist skills that can be difficult to find.
- Rationalize the delivery of enterprise BI to meet requirements that aren't addressed by existing BI tools, while decreasing complexity level, reducing cost of ownership, and/or standardizing from multiple BI tools currently in use.
- Address economic pressures for increased productivity with fewer resources, time, and staffing.

Achieve Power BI migration success

Every migration is slightly different. It can depend on the organizational structure, data strategies, data management maturity, and organizational objectives. However, there are some practices we consistently see with our customers who achieve Power BI migration success.

- **Executive sponsorship**: Identify an executive sponsor early in the process. This person should be someone who actively supports BI in the organization and is personally invested in achieving a positive outcome for the migration. Ideally, the executive sponsor has ultimate authority and accountability for outcomes related to Power BI. For more information, see this article.
- **Training, support, and communication**: Recognize that it's more than just a technology initiative. Any BI or analytics project is also a people initiative, so consider investing early in user training and support. Also, create a communication plan that transparently explains to all stakeholders what is occurring, why, and sets realistic expectations. Be sure to include a feedback loop in your communication plan to capture input from stakeholders.
- **Quick wins**: Initially, prioritize high value items that have tangible business value and are pressing. Rather than strictly attempting to always migrate reports precisely as they appear in the legacy BI platform, focus on the business question the report is trying to answer—including action to be taken—when addressing the redesigned report.
- **Modernization and improvements**: Be willing to rethink how things have always been done. A migration can provide an opportunity to deliver improvements. For example, it could eliminate manual data preparation or relocate business rules that were confined to a single report. Consider refactoring, modernizing, and consolidating existing solutions when the effort can be justified. It can include consolidating multiple reports into one, or eliminating legacy items that haven't been used for some time.
- **Continual learning**: Be prepared to use a phased approach while continually learning and adapting. Work in short, iterative cycles to bring value quickly. Make a frequent practice of completing small POCs to minimize risk of unknowns, validate assumptions, and learn about new features. As Power BI is a cloud service that updates monthly, it's important to keep abreast of developments and adjust course when appropriate.
- **Resistance to change**: Understand there might be varying levels of resistance to change; some users will resist learning a new tool. Also, some professionals who have dedicated significant time and effort to gain expertise with a different BI tool might feel threatened by being displaced. Be prepared, because it can result in internal political struggles, particularly in highly decentralized organizations.
- **Constraints**: Be realistic with migration plans, including funding, time estimates, as well as roles and responsibilities for everyone involved.

Acknowledgments

This series of articles was written by Melissa Coates, Data Platform MVP and owner of Coates Data Strategies. Contributors and reviewers include Marc Reguera, Venkatesh Titte, Patrick Baumgartner, Tamer Farag, Richard Tkachuk, Matthew Roche, Adam Saxton, Chris Webb, Mark Vaillancourt, Daniel Rubiolo, David Iseminger, and Peter Myers.

Related content

In the next article in this Power BI migration series, learn about the pre-migration steps when migrating to Power BI.

Other helpful resources include:

- Fabric adoption roadmap
- Power BI implementation planning
- Microsoft's BI transformation
- Migrate SSRS reports to Power BI
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

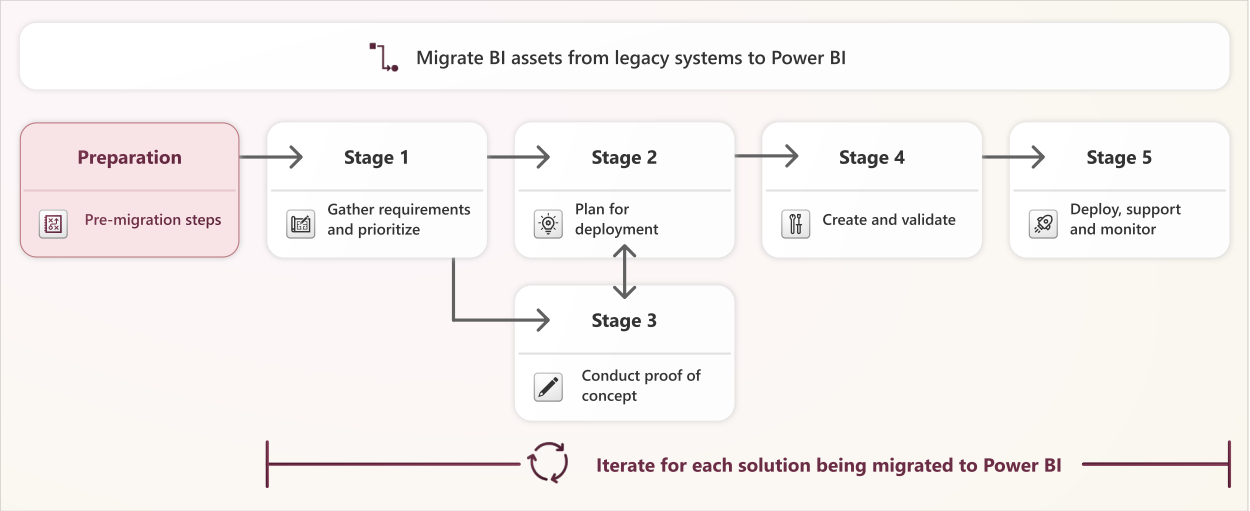
Was this page helpful? Yes No

Provide product feedback | Ask the community

Prepare to migrate to Power BI

Article • 12/30/2024

This article describes actions you can consider prior to migrating to Power BI.



> **Note**
>
> For a complete explanation of the above graphic, see Power BI migration overview.

The pre-migration steps emphasize up-front planning, which is important preparation before moving through the five migration stages. Most of the pre-migration steps will occur once, though for larger organizations some portions might be iterative for each business unit or departmental area.

The output from the pre-migration steps includes an initial governance model, initial high-level deployment planning, in addition to an inventory of the reports and data to be migrated. Additional information from activities in Stages 1, 2, and 3 will be necessary to fully estimate the level of effort for migrating individual solutions.

> **Tip**
>
> Most of the topics discussed in this article also apply to a standard Power BI implementation project.

Create cost/benefit analysis and evaluation

Several top considerations during the initial evaluation include obtaining:

- Clarity on the business case and BI strategy to reach a specific desired future state.
- Clarity on what success means, and how to measure progress and success for the migration initiative.
- Cost estimates and return on investment (ROI) calculation results.
- Successful results for several productive Power BI initiatives that are smaller in scope and complexity level.

Identify stakeholders and executive support

Several considerations for identifying stakeholders include:

- Ensure executive sponsorship is in place.
- Ensure alignment with stakeholders on the business case and BI strategy.
- Include representatives from throughout the business units—even if their content is slated for migration on a later timetable—to understand their motivations and concerns.
- Involve Power BI champions early.
- Create, and follow, a communication plan with stakeholders.

> **Tip**
>
> If you fear you're starting to overcommunicate, then it's probably just about right.

Generate initial governance model

Several key items to address early in a Power BI implementation include:

- Specific goals for adoption and where Microsoft Fabric and Power BI fit into the overall BI strategy for the organization.
- How the Fabric administrator role will be handled, particularly in decentralized organizations.
- Policies related to achieving trusted data: use of authoritative data sources, addressing data quality issues, and use of consistent terminology and common definitions.
- Security and data privacy strategy for data sources, data models, reports, and content delivery to internal and external users.
- How internal and external compliance, regulatory, and audit requirements will be met.

> **Important**

The most effective governance model strives to balance user empowerment with the necessary level of control. See more information, read about **discipline at the core** and **flexibility at the edge**.

Conduct initial deployment planning

Initial deployment planning involves defining standards, policies, and preferences for the organization's Power BI implementation.

Note that Stage 2 references solution-level deployment planning. The Stage 2 activities should respect the organizational-level decisions whenever possible.

Some critical items to address early in a Power BI implementation include:

- Fabric tenant setting decisions, which should be documented.
- Workspace management decisions, which should be documented.
- Considerations and preferences related to data and content distribution methods, such as apps, workspaces, sharing, subscriptions, and embedding of content.
- Preferences related to semantic model modes, such as use of Import mode, DirectQuery mode, or combining the two modes in a composite model.
- Securing data and access.
- Working with shared semantic models for reusability.
- Applying data certification to promote the use of authoritative and trustworthy data.
- Use of different report types, including Power BI reports, Excel reports, or paginated reports for different use cases or business units.
- Change management approaches for managing centralized BI items and business-managed BI items.
- Training plans for consumers, data modelers, report authors, and administrators.
- Support for content authors by using Power BI Desktop templates, custom visuals, and documented report design standards.
- Procedures and processes for managing user requirements, such as requesting new licenses, adding new gateway data sources, gaining permission to gateway data sources, requesting new workspaces, workspace permissions changes, and other common requirements that might be encountered on a regular basis.

Important

Deployment planning is an iterative process. Deployment decisions will be refined and augmented many times as your organization's experience with Power BI grows, and as Power BI evolves. The decisions made during this process will be used

during the solution-level deployment planning discussed in Stage 2 of the migration process.

Establish initial architecture

Your BI solution architecture will evolve and mature over time. Power BI setup tasks to handle right away include:

- Power BI tenant setup and integration with Microsoft Entra ID.
- Define Power BI administrators.
- Procure and assign initial user licenses.
- Configure and review Fabric tenant settings.
- Setup workspace roles and assign access to Microsoft Entra security groups and users.
- Configure an initial data gateway cluster—with a plan to update regularly.
- Procure initial Premium capacity subscription (if applicable).
- Configure Premium capacity workloads—with a plan to manage on an ongoing basis.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Define success criteria for migration

The first task is to understand what success looks like for migrating an individual solution. Questions you might ask include:

- **What are the specific motivations and objectives for this migration?** For more information, see Power BI migration overview (Consider migration reasons). This article describes the most common reasons for migrating to Power BI. Certainly, your objectives should be specified at the organizational level. Beyond that, migrating one legacy BI solution could benefit significantly from cost savings,

whereas migrating a different legacy BI solution could focus on gaining workflow optimization benefits.

- **What's the expected cost/benefit or ROI for this migration?** Having a clear understanding of expectations related to cost, increased capabilities, decreased complexity, or increased agility, is helpful in measuring success. It can provide guiding principles to help with decision-making during the migration process.
- **What key performance indicators (KPIs) will be used to measure success?** The following list presents some example KPIs:
  - Number of reports rendered from legacy BI platform, decreasing month over month.
  - Number of reports rendered from Power BI, increasing month over month.
  - Number of Power BI report consumers, increasing quarter over quarter.
  - Percentage of reports migrated to production by target date.
  - Cost reduction in licensing cost year over year.

> **Tip**
>
> The **Power BI activity log** can be used as a source for measuring KPI progress.

Prepare inventory of existing reports

Preparing an inventory of existing reports in the legacy BI platform is a critical step towards understanding what already exists. The outcome of this step is an input to assessing the migration effort level. Activities related to preparing an inventory can include:

1. **Inventory of reports**: Compile a list of reports and dashboards that are migration candidates.
2. **Inventory of data sources**: Compile a list of all data sources accessed by existing reports. It should include both enterprise data sources as well as departmental and personal data sources. This process could unearth data sources not previously known to the IT department, often referred to as *shadow IT*.
3. **Audit log**: Obtain data from the legacy BI platform audit log to understand usage patterns and assist with prioritization. Important information to obtain from the audit log includes:

   - Average number of times each report was executed per week/month/quarter.
   - Average number of consumers per report per week/month/quarter.
   - The consumers for each report, particularly reports used by executives.
   - Most recent date each report was executed.

> **Note**
>
> In many cases, the content isn't migrated to Power BI exactly as is. The migration represents an opportunity to redesign the data architecture and/or improve report delivery. Compiling an inventory of reports is crucial to understanding what currently exists so you can begin to assess what refactoring needs to occur. The remaining articles in this series describe possible improvements in more detail.

Explore automation options

It isn't possible to completely automate a Power BI conversion process end-to-end.

Compiling the existing inventory of data and reports is a possible candidate for automation when you have an existing tool that can do it for you. The extent to which automation can be used for some portions of the migration process—such as compiling the existing inventory—highly depends upon the tools you have.

Related content

In the next article in this Power BI migration series, learn about Stage 1, which is concerned with gathering and prioritizing requirements when migrating to Power BI.

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

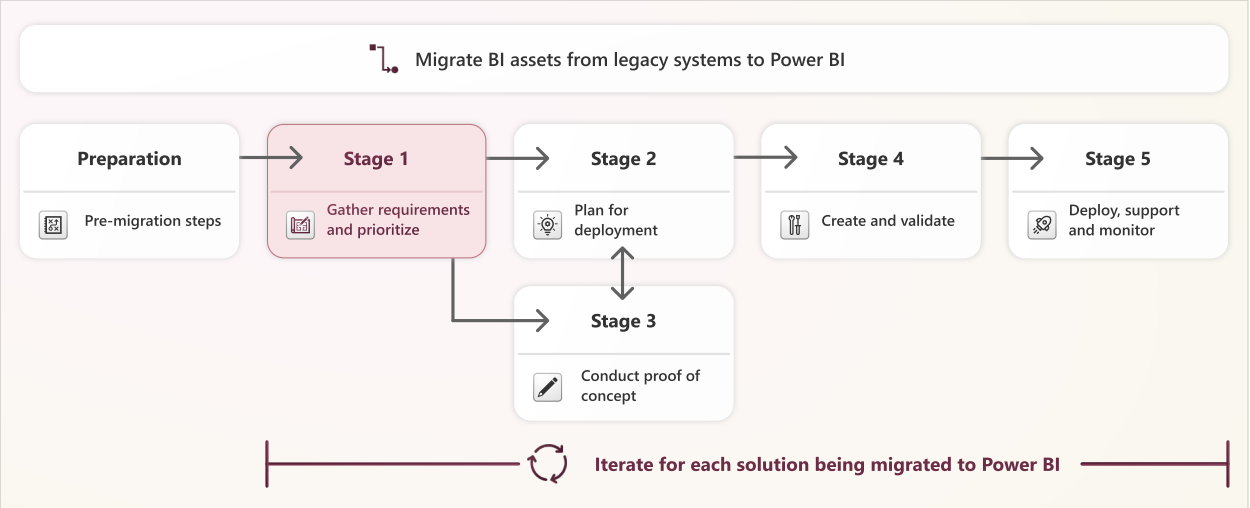
Was this page helpful? Yes No

Provide product feedback | Ask the community

Gather requirements to migrate to Power BI

Article • 12/30/2024

This article describes **Stage 1**, which is concerned with gathering and prioritizing requirements when migrating to Power BI.



> **Note**
>
> For a complete explanation of the above graphic, see Power BI migration overview.

The emphasis of Stage 1 is on information gathering and planning for an individual solution that will be migrated to Power BI.

The output from Stage 1 includes detailed requirements that have been prioritized. However, additional activities in Stages 2 and 3 must be completed to fully estimate the level of effort.

> **Important**
>
> Stages 1-5 represent activities related to one specific solution. There are decisions and activities at the organizational/tenant level which impact the process at the solution level. Some of those higher-level planning activities are discussed in the Power BI migration overview article. When appropriate, defer to the organizational-level decisions for efficiency and consistency.

The **Fabric adoption roadmap** describes these types of strategic and tactical considerations. It has an emphasis on **organizational adoption**.

> **Tip**
>
> Most of the topics discussed in this article also apply to a standard Power BI implementation project.

Compile requirements

The inventory of existing BI items, compiled in the pre-migration steps, become the input for the requirements of the new solution to be created in Power BI. Collecting requirements is about understanding the current state, as well as what items users would like changed or refactored when reports are redesigned in Power BI. Detailed requirements will useful for solution deployment planning in Stage 2, during creation of a proof of concept in Stage 3, and when creating the production-ready solution in Stage 4.

Gather report requirements

Compile thorough, easy-to-reference, information about reports, such as:

- **Purpose, audience, and expected action**: Identify the purpose and business process applicable to each report, as well as the audience, analytical workflow, and expected action to be taken by report consumers.
- **How consumers use the report**: Consider sitting with report consumers of the existing report to understand exactly what they do with it. You might learn that certain elements of the report can be eliminated or improved in the new Power BI version. This process involves additional time investment but it's valuable for critical reports or reports that are used often.
- **Owner and subject matter expert**: Identify the report owner and any subject matter expert(s) associated with the report or data domain. They might become the owners of the new Power BI report going forward. Include any specific change management requirements (which typically differ between IT-managed and business-managed solutions) as well as approvals and sign-offs, which will be required when changes are made in the future. For more information, see this article.
- **Content delivery method**: Clarify report consumer expectations for content delivery. It could be on-demand, interactive execution, embedded within a custom

application, or delivery on a schedule using an e-mail subscription. There might also be requirements to trigger alert notifications.

- **Interactivity needs**: Determine *must-have* and *nice-to-have* interactivity requirements, such as filters, drill-down actions, or drillthrough actions.
- **Data sources**: Ensure all data sources required by the report are discovered, and data latency needs (data freshness) are understood. Identify historical data, trending, and data snapshot requirements for each report so they can be aligned with the data requirements. Data source documentation can also be useful later on when performing data validation of a new report with its source data.
- **Security requirements**: Clarify security requirements (such as allowed viewers, allowed editors, and any row-level security needs), including any exceptions to normal organizational security. Document any data sensitivity level, data privacy, or regulatory/compliance needs.
- **Calculations, KPIs, and business rules**: Identify and document all calculations, KPIs, and business rules that are currently defined within the existing report so they can be aligned with the data requirements.
- **Usability, layout, and cosmetic requirements**: Identify specific usability, layout, and cosmetic needs related to data visualizations, grouping and sorting requirements, and conditional visibility. Include any specific considerations related to mobile device delivery.
- **Printing and exporting needs**: Determine whether there are any requirements specific to exporting or print-ready layout. These needs will influence which type of report will be most suitable (such as a Power BI, Excel, or paginated report). Be aware that report consumers tend to place a lot of importance on how they've always done things, so don't be afraid to challenge their way of thinking. Be sure to talk in terms of *enhancements* rather than *change*.
- **Risks or concerns**: Determine whether there are other technical or functional requirements for reports, as well as any risks or concerns regarding the information being presented in them.
- **Open issues and backlog items**: Identify any future maintenance, known issues, or deferred requests to add to the backlog at this time.

> **Tip**
>
> Consider ranking requirements by classifying them as *must have* or *nice to have*. Frequently consumers ask for everything they might possibly need up-front because they believe it's their only chance to make requests. Also, when addressing priorities in multiple iterations, make the backlog available to stakeholders. It helps with communication, decision-making, and the tracking of pending commitments.

Gather data requirements

Compile detailed information pertaining to data, such as:

- **Existing queries**: Identify whether there are existing report queries or stored procedures that can be used by a DirectQuery model or a Composite model, or can be converted to an Import model.
- **Types of data sources**: Compile the types of data sources that are necessary, including centralized data sources (such as an enterprise data warehouse) as well as non-standard data sources (such as flat files or Excel files that augment enterprise data sources for reporting purposes). Finding where data sources are located, for purposes of data gateway connectivity, is important too.
- **Data structure and cleansing needs**: Determine the data structure for each requisite data source, and to what extent data cleansing activities are necessary.
- **Data integration**: Assess how data integration will be handled when there are multiple data sources, and how relationships can be defined between each model table. Identify specific data elements needed to simplify the model and reduce its size.
- **Acceptable data latency**: Determine the data latency needs for each data source. It will influence decisions about which data storage mode to use. Data refresh frequency for Import model tables is important to know too.
- **Data volume and scalability**: Evaluate data volume expectations, which will factor into decisions about large model support and designing DirectQuery or Composite models. Considerations related to historical data needs are essential to know too. For larger semantic models, determining incremental data refresh will also be necessary.
- **Measures, KPIs, and business rules**: Assess needs for measures, KPIs, and business rules. They will impact decisions regarding where to apply the logic: in the semantic model or the data integration process.
- **Master data and data catalog**: Consider whether there are master data issues requiring attention. Determine if integration with an enterprise data catalog is appropriate for enhancing discoverability, accessing definitions, or producing consistent terminology accepted by the organization.
- **Security and data privacy**: Determine whether there are any specific security or data privacy considerations for semantic models, including row-level security requirements.
- **Open issues and backlog items**: Add any known issues, known data quality defects, future maintenance, or deferred requests to the backlog at this time.

> **Important**

Data reusability can be achieved with **shared semantic models**, which can optionally be **certified** to indicate trustworthiness and improve discoverability. Data preparation reusability can be achieved with **dataflows** to reduce repetitive logic in multiple semantic models. Dataflows can also significantly reduce the load on source systems because the data is retrieved less often—multiple semantic models can then import data from the dataflow.

Identify improvement opportunities

In most situations, some modifications and improvements occur. It's rare that a direct one-to-one migration occurs without any refactoring or enhancement. Three types of improvements you can consider include:

- **Consolidation of reports**: Similar reports might be consolidated using techniques such as filters, bookmarks, or personalization. Having fewer reports, which are each more flexible, can significantly improve the experience for report consumers. Consider optimizing semantic models for Q&A (natural language querying) to deliver even greater flexibility to report consumers, allowing them to create their own visualizations.
- **Efficiency improvements**: During requirements gathering, improvements can often be identified. For instance, when analysts compile numbers manually or when a workflow can be streamlined. Power Query can play a large role in replacing manual activities that are currently performed. If business analysts find themselves performing the same activities to cleanse and prepare data on a regular basis, repeatable Power Query data preparation steps can yield significant time savings and reduce errors.
- **Centralization of data model**: An authoritative and certified semantic model serves as the backbone for managed self-service BI. In this case, the data is managed once, and analysts have flexibility to use and augment that data to meet their reporting and analysis needs.

> ⓘ Note
>
> For more information about centralization of data models, read about discipline at the core and flexibility at the edge.

Prioritize and assess complexity

At this point, the initial inventory is available and might include specific requirements. When prioritizing the initial set of BI items ready for migration, reports and data should be considered collectively as well as independently of each other.

Identify high priority reports, which might include reports that:

- Bring significant value to the business.
- Are executed frequently.
- Are required by senior leadership or executives.
- Involve a reasonable level of complexity (to improve chances of success during the initial migration iterations).

Identify high priority data, which might include data that:

- Contains critical data elements.
- Is common organizational data that serves many use cases.
- Can be used to create a shared semantic model for reuse by reports and many report creators.
- Involves a reasonable level of complexity (to improve chances of success when in the initial migration iterations).

Related content

In the next article in this Power BI migration series, learn about Stage 2, which is concerned with planning the migration for a single Power BI solution.

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

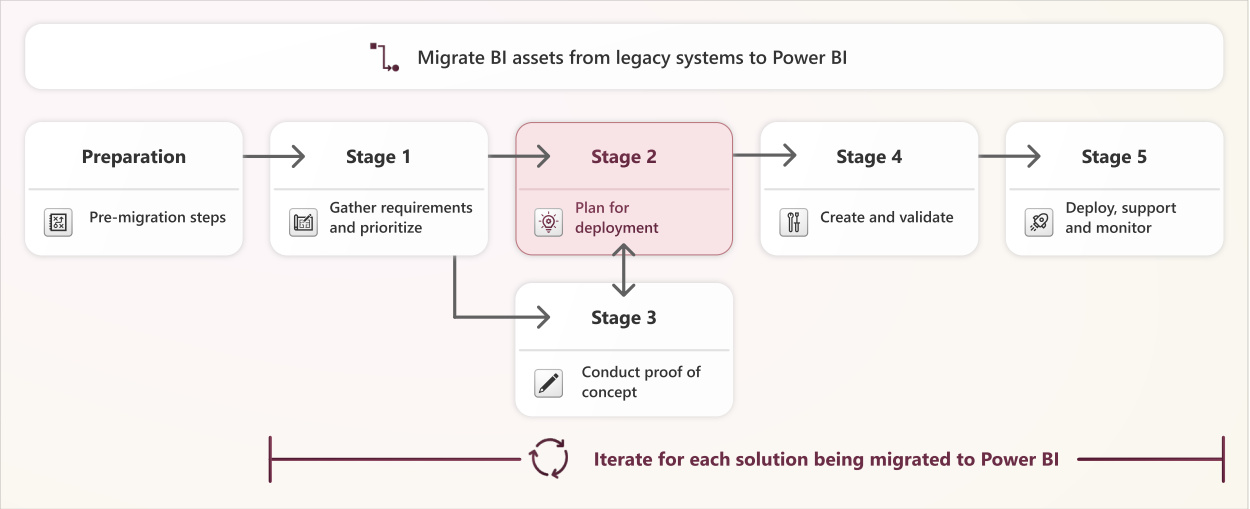
Feedback

Was this page helpful? Yes No

Plan deployment to migrate to Power BI

Article • 12/30/2024

This article describes **Stage 2**, which is concerned with planning the migration for a single Power BI solution.



> **Note**
>
> For a complete explanation of the above graphic, see Power BI migration overview.

The focus of Stage 2 is on defining how the requirements that were defined in Stage 1 are used to migrate a solution to Power BI.

The output from Stage 2 includes as many specific decisions as possible to guide the deployment process.

Decision-making of this nature is an iterative and non-linear process. Some planning will have already occurred in the pre-migration steps. Learnings from a proof of concept (described in Stage 3) can occur in parallel with deployment planning. Even while creating the solution (described in Stage 4), additional information might arise that influences deployment decisions.

> **Important**
>
> Stages 1-5 represent activities related to one specific solution. There are decisions and activities at the organizational/tenant level which impact the process at the solution level. Some of those higher-level planning activities are discussed in the

Power BI migration overview article. When appropriate, defer to the organizational-level decisions for efficiency and consistency.

> **Tip**
>
> The topics discussed in this article also apply to a standard Power BI implementation project.

Choose Power BI product

One of the first decisions is to choose the Power BI product. It's a decision between the Power BI service or Power BI Report Server. Once content has been published, many additional options become available, such as embedding, mobile delivery, and email subscriptions.

> **Caution**
>
> If you're tempted to rely on using Power BI Desktop files stored in a file system, be aware that it's not an optimal approach. Using the Power BI service (or Power BI Report Server) has significant advantages for security, content distribution, and collaboration. The ability to audit and monitor activities is also enabled by the Power BI service.

Decide on workspace management approach

Workspaces are a core concept of the Power BI service, which makes workspace management an important aspect of planning. Questions to ask include:

- Is a new workspace needed for this new solution?
- Will separate workspaces be needed to accommodate development, test, and production?
- Will separate workspaces be used for data and reports, or will a single workspace be sufficient? Separate workspaces have numerous advantages, especially for securing semantic models. When necessary, they can be managed separately from those users who publish reports.
- What are the security requirements for the workspace? It influences planning for workspace roles. If an app will be used by content consumers, audience permissions for the app are managed separately from the workspace. Distinct

permissions for app viewers allow additional flexibility in meeting security requirements for read-only consumers of reports or dashboards.

- Can existing groups be used for securing the new content? Both groups in Microsoft Entra ID and Microsoft 365 are supported. When aligned with existing processes, using groups makes permissions management easier than assignments to individual users.
- Are there any security considerations related to external guest users? You might need to work with your Microsoft Entra administrator and your Power BI administrator to set up guest user access.

> **Tip**
>
> Consider creating a workspace for a specific business activity or project. You might be tempted to start off structuring workspaces based on your organizational structure (such as a workspace per department), but this approach frequently ends up being too broad.

Determine how content will be consumed

It's helpful to understand how consumers of a solution prefer to view reports and dashboards. Questions to ask include:

- Will a Power BI app (which comprises reports and dashboards from a single workspace) be the best way to deliver content to consumers, or will direct access to a workspace be sufficient for content viewers?
- Will certain reports and dashboards be embedded elsewhere, such as Teams, SharePoint Online, or a secure portal or website?
- Will consumers access content using mobile devices? Requirements to deliver reports to small form factor devices will influence some report design decisions.

Decide whether other content can be created

There are several key decisions to be made related to allowing consumers to create new content, such as:

- Will consumers be allowed to create new reports from the published semantic model? This capability can be enabled by assigning semantic model Build permission to a user.
- If consumers want to customize a report, can they save a copy of it and personalize it to meet their needs?

> ⊗ **Caution**
>
> Although the *Save a copy* capability is a nice feature, it should be used with caution when the report includes certain graphics or header/footer messages. Since logos, icons, and textual messages often relate to branding requirements or regulatory compliance, it's important to carefully control how they're delivered and distributed. If *Save a copy* is used, but the original graphics or header/footer messages remain unchanged by the new author, it can result in confusion about who actually produced the report. It can also reduce the meaningfulness of the branding.

Evaluate needs for Premium capacity

> ⓘ **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Additional capabilities are available when a workspace is stored on a Premium capacity. Here are several reasons why workspaces on Premium capacity can be advantageous:

- Content can be accessed by consumers who don't have a Power BI Pro or Premium Per User (PPU) license.
- Support for large semantic models.
- Support for more frequent data refreshes.
- Support for using the full feature set of dataflows.
- Enterprise features, including deployment pipelines and the XMLA endpoint.

Determine data acquisition method

The data required by a report might influence several decisions. Questions to ask include:

- Can an existing Power BI shared semantic models be used, or is the creation of a new Power BI semantic model appropriate for this solution?
- Does an existing shared semantic model need to be augmented with new data or measures to meet additional needs?
- Which data storage mode will be most appropriate? Options include Import, DirectQuery, Composite, or Live Connection.
- Should aggregations be used to enhance query performance?
- Will creation of a dataflow be useful and can it serve as a source for numerous semantic models?
- Will a new gateway data source need to be registered?

Decide where original content will be stored

In addition to planning the target deployment destination, it's also important to plan where the original—or source—content will be stored, such as:

- Specify an approved location for storing the original Power BI Desktop (.pbix) files. Ideally, this location is available only to people who edit the content. It should align with how security is set up in the Power BI service.
- Use a location for original Power BI Desktop files that includes versioning history or source control. Versioning permits the content author to revert to a previous file version, if necessary. OneDrive for work or school or SharePoint work well for this purpose.
- Specify an approved location for storing non-centralized source data, such as flat files or Excel files. It should be a path that any of the semantic model creators can reach without error and is backed up regularly.
- Specify an approved location for content exported from the Power BI service. The goal is to ensure that security defined in the Power BI service isn't inadvertently circumvented.

> **Important**
>
> Specifying a protected location for original Power BI Desktop files is particularly important when they contain imported data.

Assess the level of effort

Once sufficient information is available from the requirements (which were described in Stage 1) and the solution deployment planning process, it's now possible to assess the

level of effort. It's then possible to formulate a project plan with tasks, timeline, and responsibility.

> **Tip**
>
> Labor costs—salaries and wages—are usually among the highest expenses in most organizations. Although it can be difficult to accurately estimate, productivity enhancements have an excellent return on investment (ROI).

Related content

In the next article in this Power BI migration series, learn about Stage 3, which is concerned with conducting a proof of concept to mitigate risk and address unknowns as early as possible when migrating to Power BI.

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

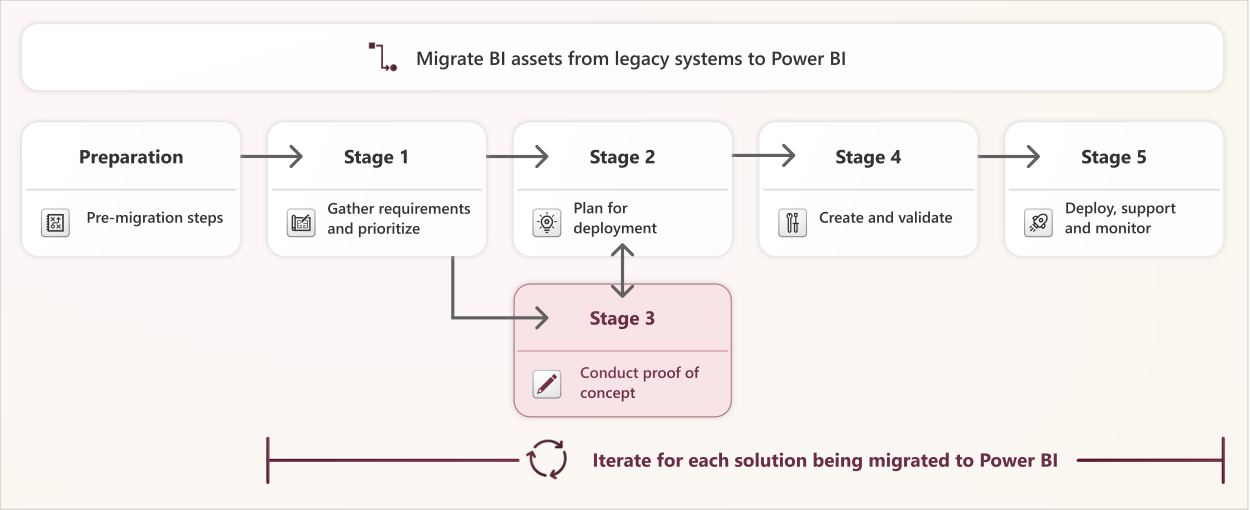
Was this page helpful? Yes No

Provide product feedback | Ask the community

Conduct proof of concept to migrate to Power BI

Article • 12/30/2024

This article describes **Stage 3**, which is concerned with conducting a proof of concept (POC) to mitigate risk and address unknowns as early as possible when migrating to Power BI.



> **Note**
>
> For a complete explanation of the above graphic, see Power BI migration overview.

The focus of Stage 3 is to address unknowns and mitigate risks as early as possible. A technical POC is helpful for validating assumptions. It can be done iteratively alongside solution deployment planning (described in Stage 2).

The output from this stage is a Power BI solution that's narrow in scope, addresses the initial open questions, and is ready for additional work in Stage 4 to make it production-ready.

> **Important**
>
> We don't intend for the POC to be disposable work. Rather, we expect it to be an early iteration of the production-ready solution. In your organization, you might refer to this activity as a prototype, pilot, mockup, quick start, or minimally viable product (MVP). Conducting a POC isn't always necessary and it could even happen informally.

> **Tip**
>
> Most of the topics discussed in this article also apply to a standard Power BI implementation project. As your organization becomes more experienced with Power BI, the need to conduct POCs diminishes. However, due to the fast release cadence with Power BI and the continual introduction of new features, you might regularly conduct technical POCs for learning purposes.

Set POC goals and scope

When conducting a POC, focus on the following goals:

- Verify your assumptions about how a feature works.
- Educate yourself on differences in how Power BI works compared with the legacy BI platform.
- Validate initial understandings of certain requirements with subject matter experts.
- Create a small semantic model with real data to understand and detect any issues with the data structure, relationships, data types, or data values.
- Experiment with, and validate, DAX syntax expressions used by model calculations.
- Test data source connectivity using a gateway (if it's to be a gateway source).
- Test data refresh using a gateway (if it's to be a gateway source).
- Verify security configurations, including row-level security when applicable.
- Experiment with layout and cosmetic decisions.
- Verify that all functionality in the Power BI service works as expected.

The POC scope is dependent on what the unknowns are, or which goals need to be validated with colleagues. To reduce complexity, keep a POC as narrow as possible in terms of scope.

Most often with a migration, requirements are well known because there's an existing solution to start from. However, depending on the extent of improvements to be made or existing Power BI skills, a POC still provides significant value. In addition, rapid prototyping with consumer feedback could be appropriate to quickly clarify requirements—especially if enhancements are made.

> **Important**
>
> Even if a POC includes only a subset of data, or includes only limited visuals, it's often important to take it from start to finish. That is, from development in Power BI Desktop to deployment to a development workspace in the Power BI service. It's

the only way to fully accomplish the POC objectives. It's particularly true when the Power BI service must deliver critical functionality that you haven't used before, like a DirectQuery semantic model that uses single sign-on. During the POC, focus your efforts on aspects you're uncertain about or need to verify with others.

Handle differences in Power BI

Power BI can be used as a *model-based tool* or as a *report-based tool*. A model-based solution involves developing a data model, whereas a report-based solution connects to an already-deployed data model.

Due to its extreme flexibility, there are some aspects about Power BI that might be fundamentally different from the legacy BI platform you're migrating from.

Consider redesigning the data architecture

If you're migrating from a legacy BI platform that has its own semantic layer, then the creation of an Import semantic model is likely to be a good option. Power BI functions best with a star schema table design. Therefore, if the legacy semantic layer is not a star schema, it's possible that some redesign might be required to fully benefit from Power BI. Putting effort into defining a semantic layer adhering to star schema design principles (including relationships, commonly used measures, and friendly organizational terminology) serves as an excellent starting point for self-service report authors.

If you're migrating from a legacy BI platform where reports reference relational data sources using SQL queries or stored procedures, and if you're planning to use Power BI in DirectQuery mode, you might be able to achieve close to a one-to-one migration of the data model.

> **⊗ Caution**
>
> If you see the creation of lots of Power BI Desktop files comprising a single imported table, it's usually an indicator that the design isn't optimal. Should you notice this situation, investigate whether the use of **shared semantic models** that are created using a **star schema** design could achieve a better result.

Decide how to handle dashboard conversions

In the BI industry, a dashboard is a collection of visuals that displays key metrics on a single page. However, in Power BI, a dashboard represents a specific visualization

feature that can only be created in the Power BI service. When migrating a dashboard from a legacy BI platform, you have two choices:

1. The legacy dashboard can be recreated as a Power BI *report*. Most reports are created with Power BI Desktop. Paginated reports and Excel reports are alternative options, too.
2. The legacy dashboard can be recreated as a Power BI *dashboard*. Dashboards are a visualization feature of the Power BI service. Dashboard visuals are often created by pinning visuals from one or more reports, Q&A, or Quick Insights.

> **Tip**
>
> Because dashboards are a Power BI content type, refrain from using the word *dashboard* in the report or dashboard name.

Focus on the big picture when recreating visuals

Every BI tool has its strengths and focus areas. For this reason, the exact report visuals you depended on in a legacy BI platform might not have a close equivalent in Power BI.

When recreating report visuals, focus more on the big picture business questions that are being addressed by the report. It removes the pressure to replicate the design of every visual in precisely the same way. While content consumers appreciate consistency when using migrated reports, it's important not to get caught up in time-consuming debates about small details.

Related content

In the next article in this Power BI migration series, learn about stage 4, which is concerned with creating and validating content when migrating to Power BI.

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

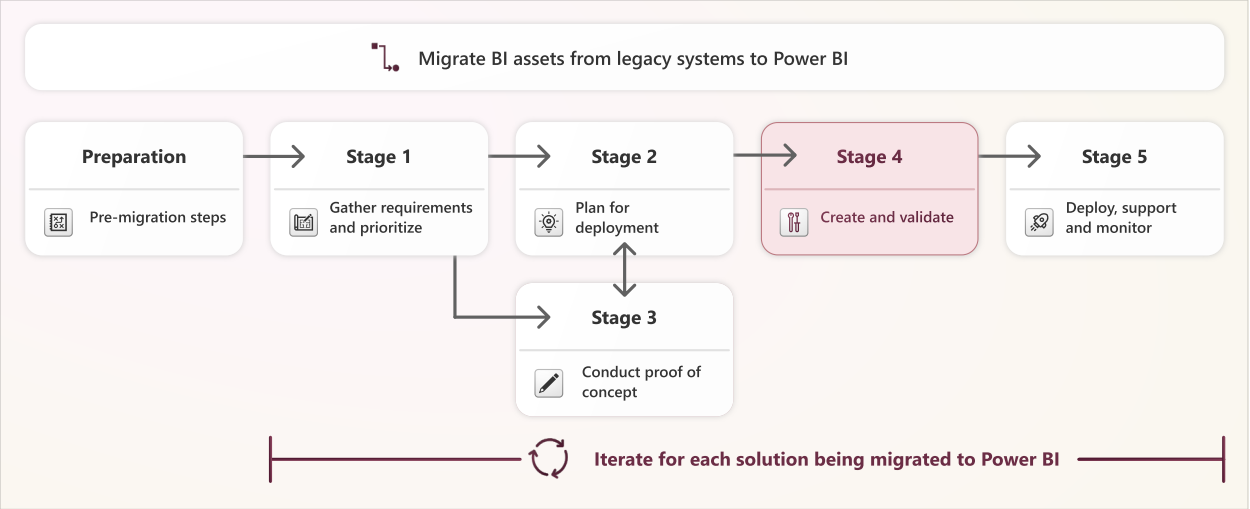
Was this page helpful? Yes No

Provide product feedback | Ask the community

Create content to migrate to Power BI

Article • 12/30/2024

This article describes **Stage 4**, which is concerned with creating and validating content when migrating to Power BI.



> **Note**
>
> For a complete explanation of the above graphic, see Power BI migration overview.

The focus of Stage 4 is performing the actual work to convert the proof of concept (POC) to a production-ready solution.

The output from this stage is a Power BI solution that has been validated in a development workspace and is ready for deployment to production.

> **Tip**
>
> Most of the topics discussed in this article also apply to a standard Power BI implementation project.

Create the production solution

At this juncture, the same person who performed the POC might carry on with producing the production-ready Power BI solution. Or, someone different could be involved. If timelines are not jeopardized, it's great to get people involved who will be responsible for Power BI development in the future. This way, they can actively learn.

> **Important**
>
> Reuse as much of the work from the POC as possible.

Develop new Import semantic model

You might choose to create a new Import semantic model when an existing Power BI semantic model doesn't already exist to meet your needs, or if it can't be enhanced to meet your needs.

Ideally, from the very beginning, consider decoupling the development work for data and reports. Decoupling data and reports will facilitate the separation of work, and permissions, when different people are responsible for data modeling and reports. It makes for a more scalable approach and encourages data reusability.

The essential activities related to development of an Import semantic model include:

- Acquire data from one or more data sources (which could be a Power BI dataflow).
- Shape, combine, and prepare data.
- Create the semantic model, including date tables.
- Create and verify model relationships.
- Define measures.
- Set up row-level security, if necessary.
- Configure synonyms and optimize Q&A.
- Plan for scalability, performance, and concurrency, which might influence your decisions about data storage modes, such as using a Composite model or aggregations.

> **Tip**
>
> If you have different development/test/production environments, consider **parameterizing** data sources. It will make deployment, described in **Stage 5**, significantly easier.

Develop new reports and dashboards

The essential activities related to development of a Power BI report or dashboard include:

- Decide on using a Live Connection to an existing data model, or creating a new data model
- When creating a new data model, decide on the data storage mode for model tables (Import, DirectQuery, or Composite).
- Decide on the best data visualization tool to meet requirements: Power BI Desktop, Paginated Report Builder, or Excel.
- Decide on the best visuals to tell the story the report needs to tell, and to address the questions the report needs to answer.
- Ensure all visuals present clear, concise, and business-friendly terminology.
- Address interactivity requirements.
- When using Live Connection, add report-level measures.
- Create a dashboard in the Power BI service, especially when consumers want an easy way to monitor key metrics.

> **Note**
>
> Many of these decisions will have been made in earlier stages of planning or in the technical POC.

Validate the solution

There are four main aspects to validation of a Power BI solution:

1. Data accuracy
2. Security
3. Functionality
4. Performance

Validate data accuracy

As a one-time effort during the migration, you'll need to ensure the data in the new report matches what's displayed in the legacy report. Or—if there's a difference—be able to explain why. It's more common than you might think to find an error in the legacy solution that gets resolved in the new solution.

As part of ongoing data validation efforts, the new report will typically need to be cross-checked with the original source system. Ideally, this validation occurs in a repeatable way every time you publish a report change.

Validate security

When validating security, there are two primary aspects to consider:

- Data permissions
- Access to semantic models, reports, and dashboards

In an Import semantic model, data permissions are applied by defining row-level security (RLS). It's also possible that data permissions are enforced by the source system when using DirectQuery storage mode (possibly with single sign-on).

The main ways to grant access to Power BI content are:

- Workspace roles (for content editors and viewers).
- App audience permissions applied to a packaged set of workspace content (for viewers).
- Sharing an individual report or dashboard (for viewers).

> **Tip**
>
> We recommend training content authors on how to manage security effectively. It's also important to have robust testing, auditing and monitoring in place.

Validate functionality

It's the time to double-check semantic model details like field names, formatting, sorting, and default summarization behavior. Interactive report features, such as slicers, drill-down actions, drillthrough actions, expressions, buttons, or bookmarks, should all be verified, too.

During the development process, the Power BI solution should be published to a development workspace in the Power BI service on a regular basis. Verify all functionality works as expected in the service, such as the rendering of custom visuals. It's also a good time to do further testing. Test scheduled refresh, Q&A, and how reports and dashboards look on a mobile device.

Validate performance

Performance of the Power BI solution is important for consumer experience. Most reports should present visuals in under 10 seconds. If you have reports that take longer to load, pause and reconsider what might be contributing to delays. Report performance should be assessed regularly in the Power BI service, in addition to Power BI Desktop.

Many performance issues arise from substandard DAX (Data Analysis eXpressions), poor semantic model design, or suboptimal report design (for instance, trying to render too many visuals on a single page). Technical environment issues, such as the network, an overloaded data gateway, or how a Premium capacity is configured can also contribute to performance issues. For more information, see the Optimization guide for Power BI and Troubleshoot report performance in Power BI.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Document the solution

There are two main types of documentation that are useful for a Power BI solution:

- Semantic model documentation
- Report documentation

Documentation can be stored wherever it's most easily accessed by the target audience. Common options include:

- **Within a SharePoint site**: A SharePoint site might exist for your Center of Excellence or an internal Power BI community site.
- **Within an app**: URLs can be set up when publishing a Power BI app to direct the consumer to more information.
- **Within individual Power BI Desktop files**: Model elements, like tables and columns, can define a description. These descriptions appear as tooltips in the **Data** pane when authoring reports.

> **Tip**
>
> If you create a site to serve as a hub for Power BI-related documentation, consider **customizing the Get Help menu** with its URL location.

Create semantic model documentation

Semantic model documentation is targeted at users who will be managing the semantic model in the future. It's useful to include:

- Design decisions made and reasons why.
- Who owns, maintains, and certifies semantic models.
- Data refresh requirements.
- Custom business rules defined in semantic models.
- Specific semantic model security or data privacy requirements.
- Future maintenance needs.
- Known open issues or deferred backlog items.

You might also elect to create a change log that summarizes the most important changes that have happened to the semantic model over time.

Create report documentation

Report documentation, which is typically structured as a walk-through targeted at report consumers, can help consumers get more value from your reports and dashboards. A short video tutorial often works well.

You may also choose to include additional report documentation on a hidden page of your report. It could include design decisions and a change log.

Related content

In the next article in this Power BI migration series, learn about stage 5, which is concerned with deploying, supporting, and monitoring content when migrating to Power BI.

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

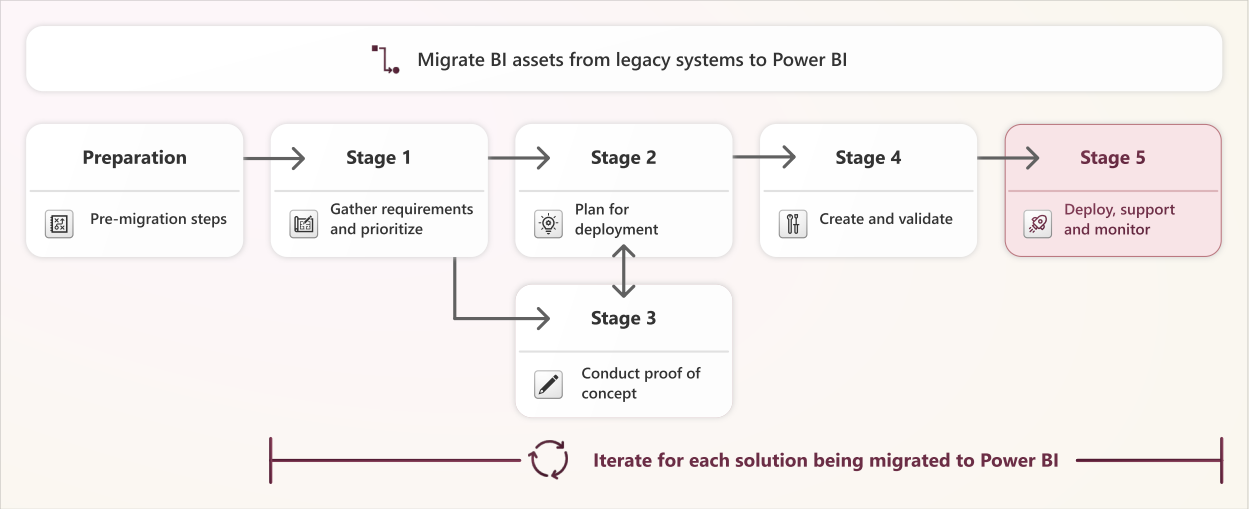
Was this page helpful? Yes No

Provide product feedback | Ask the community

Deploy to Power BI

Article • 12/30/2024

This article describes **Stage 5**, which is concerned with deploying, supporting, and monitoring content when migrating to Power BI.



> **Note**
>
> For a complete explanation of the above diagram, see Power BI migration overview.

The primary focus of Stage 5 is to deploy the new Power BI solution to production.

The output from this stage is a production solution ready for use by the business. When working with an agile method, it's acceptable to have some planned enhancements that will be delivered in a future iteration. Support and monitoring are also important at this stage, and on an ongoing basis.

> **Tip**
>
> Except for running in parallel and decommissioning the legacy reports, which are discussed below, the topics discussed in this article also apply to a standard Power BI implementation project.

Deploy to test environment

For IT-managed solutions, or solutions that are critical to business productivity, there's generally a test environment. A test environment sits between development and

production, and it's not necessary for all Power BI solutions. A test workspace can serve as a stable location, separated from development, for user acceptance testing (UAT) to occur before release to production.

If your content has been published to a workspace on Premium capacity, deployment pipelines can simplify the deployment process to development, test, and production workspaces. Alternatively, publishing can be done manually or with PowerShell scripts.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Deploy to test workspace

Key activities during a deployment to the test workspace typically include:

- **Connection strings and parameters**: Adjust semantic model connection strings if the data source differs between development and test. Parameterization can be used to effectively manage connection strings.
- **Workspace content**: Publish semantic models and reports to the test workspace, and create dashboards.
- **App.** Publish an app using the content from the test workspace, if it will form part of the UAT process. Usually, app permissions are restricted to a small number of people involved with UAT.
- **Data refresh**: Schedule the semantic model refresh for any Import semantic models for the period when UAT is actively occurring.
- **Security**: Update or verify workspace roles. Testing workspace access includes a small number of people who are involved with UAT.

Conduct user acceptance testing

Generally, UAT involves business users who are subject matter experts. Once verified, they provide their approval that the new content is accurate, meets requirements, and can be deployed for wider consumption by others.

The extent to which this UAT process is formal, including written sign-offs, will depend on your change management practices.

Deploy to production environment

There are several considerations for deploying to the production environment.

Conduct a staged deployment

If you're trying to minimize risk and user disruption, or if there are other concerns, you might opt to perform a staged deployment. The first deployment to production might involve a smaller group of pilot users. With a pilot, feedback can be actively requested from the pilot users.

Expand permissions in the production workspace, or the app, gradually until all target users have permission to the new Power BI solution.

> **Tip**
>
> Use the **Power BI Activity Log** to understand how consumers are adopting and using the new Power BI solution.

Handle additional components

During the deployment process, you might need to work with your Power BI administrators to address other requirements that are needed to support the entire solution, such as:

- **Gateway maintenance**: A new data source registration in the data gateway might be required.
- **Gateway drivers and connectors**: A new proprietary data source might require installation of a new driver or custom connector on each server in the gateway cluster.
- **Create a new Premium capacity**: You might be able to use an existing Premium capacity. Or, there might be situations when a new Premium capacity is warranted. It could be the case when you purposely wish to separate a departmental workload.
- **Set up a Power BI dataflow**: Data preparation activities can be set up once in a Power BI dataflow using Power Query Online. It helps avoid replicating data preparation work in many different Power BI Desktop files.

- **Register a new organizational visual**: Organizational visual registration can be done in the admin portal for custom visuals that didn't originate from AppSource.
- **Set featured content**: A tenant setting exists that controls who can feature content in the Power BI service home page.
- **Set up sensitivity labels**: All sensitivity labels are integrated with Microsoft Purview Information Protection.

Deploy to production workspace

Key activities during a deployment to the production workspace typically include:

- **Change management**: If necessary, obtain approval to deploy, and communicate deployment to the user population using your standard change management practices. There might be an approved change management window during which production deployments are allowed. Usually, it's applicable to IT-managed content and much less frequently applied to self-service content.
- **Rollback plan**: With a migration, the expectation is that it's the migration of a new solution for the first time. If content does already exist, it's wise to have a plan to revert to the previous version, should it become necessary. Having previous versions of the Power BI Desktop files (using SharePoint or OneDrive versioning) works well for this purpose.
- **Connection strings and parameters**: Adjust semantic model connection strings when the data source differs between test and production. Parameterization can be used effectively for this purpose.
- **Data refresh**: Schedule the semantic model refresh for any imported semantic models.
- **Workspace content**: Publish semantic models and reports to the production workspace, and create dashboards. Deployment pipelines can simplify the process to deploy to development, test, and production workspaces if your content has been published to workspaces on Premium capacity.
- **App**: If apps are part of your content distribution strategy, publish an app using the content from the production workspace.
- **Security**: Update and verify workspace roles based on your content distribution and collaboration strategy.
- **Semantic model settings**: Update and verify settings for each semantic model, including:
  - Endorsement (such as certified or promoted)
  - Gateway connection or data source credentials
  - Scheduled refresh
  - Featured Q&A questions

- **Report and dashboard settings**: Update and verify settings for each report and dashboard. The most important settings include:
  - Description
  - Contact person or group
  - Sensitivity label
  - Featured content
- **Subscriptions**: Set up report subscriptions, if necessary.

> **Important**
>
> At this point, you have reached a big milestone. Celebrate your accomplishment at completing the migration.

Communicate with users

Announce the new solution to consumers. Let them know where they can find the content, as well as associated documentation, FAQs, and tutorials. To introduce the new content, consider hosting a lunch-and-learn type of session or prepare some on-demand videos.

Be sure to include instructions on how to request help, as well as how to provide feedback.

Conduct a retrospective

Consider conducting a retrospective to examine what went well with the migration, and what could be done better with the next migration.

Run in parallel

In many situations, the new solution will run in parallel to the legacy solution for a predetermined time. Advantages of running in parallel include:

- Risk reduction, particularly if the reports are considered mission-critical.
- Allows time for users to become accustomed to the new Power BI solution.
- Allows for the information presented in Power BI to be cross-referenced to the legacy reports.

Decommission the legacy report

At some point, the reports migrated to Power BI should be disabled in the legacy BI platform. Decommissioning legacy reports can occur when:

- The predetermined time for running in parallel—which should have been communicated to the user population—has expired. It's commonly 30-90 days.
- All users of the legacy system have access to the new Power BI solution.
- Significant activity is no longer occurring on the legacy report.
- No issues have occurred with the new Power BI solution that could impact user productivity.

Monitor the solution

Events from the Power BI activity log can be used to understand usage patterns of the new solution (or the execution log for content deployed to Power BI Report Server). Analyzing the activity log can help determine whether actual use differs from expectations. It can also validate that the solution is adequately supported.

Here's some questions that can be addressed by reviewing the activity log:

- How frequently is the content being viewed?
- Who is viewing the content?
- Is the content typically viewed through an app or a workspace?
- Are most users using a browser or mobile application?
- Are subscriptions being used?
- Are new reports being created that are based on this solution?
- Is the content being updated frequently?
- How is security defined?
- Are problems occurring regularly, such as data refresh failures?
- Are concerning activities happening (for instance, significant export activity or numerous individual report shares) which could mean additional training might be warranted?

> **ⓘ Important**
>
> Be sure to have someone regularly review the activity log. Merely capturing it and storing the history does have value for auditing or compliance purposes. However, the real value is when proactive action can be taken.

Support the solution

Although the migration is complete, the post-migration period is vital for addressing issues and handling any performance concerns. Over time, the migrated solution will likely undergo changes as business needs evolve.

Support tends to happen a little differently depending on how self-service BI is managed across the organization. Power BI champions throughout the business units often informally act as first-line support. Although it's an informal role, it's a vital one that should be encouraged.

Having a formal support process, staffed by IT with support tickets, is also essential for handling routine system-oriented requests and for escalation purposes.

> **Note**
>
> The different types of **internal and external support** are described in the Fabric adoption roadmap.

You might also have a Center of Excellence (COE) that acts like internal consultants who support, educate, and govern Power BI in the organization. A COE can be responsible for curating helpful Power BI content in an internal portal.

Lastly, it should be clear to content consumers to know who to contact with questions about the content, and to have a mechanism for providing feedback on issues or improvements.

For more information about user support, with a focus on the resolution of issues, see User support in the Fabric adoption roadmap.

Related content

In the final article in this series, learn from customers when migrating to Power BI.

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Learn from customer Power BI migrations

Article • 12/30/2024

This article, which concludes the series on migrating to Power BI, shares key lessons learned by two customers who have successfully migrated to Power BI.

International consumer goods company

An international consumer goods company, which sells hundreds of products, made the decision in 2017 to pursue a cloud-first strategy. One of the major factors for selecting Power BI as its business intelligence (BI) platform is its deep integration with Azure and Microsoft 365.

Conduct a phased migration

In 2017, the company began using Power BI. The initial organizational objective was to introduce Power BI as an additional BI tool. The decision provided content authors, consumers, and IT with the time to adapt to new ways of delivering BI. It also allowed them to build expertise in Power BI.

During the second half of 2018, a formal announcement was made declaring that Power BI was the approved BI tool for the organization. And, accordingly, all new BI development work should take place in Power BI. The availability of Power BI Premium was a key driver for making this decision. At this time, the organization discouraged the use of the former BI platform, and planning for transition commenced.

Towards the end of 2019, work began to migrate existing content from the legacy BI platform to Power BI. Some early adopters migrated their content rapidly. That helped build even more momentum with Power BI around the organization. Content owners and authors were then asked to begin preparations to fully migrate to Power BI by the end of 2020. The organization does still face challenges related to skills, time, and funding—though none of their challenges are related to the technology platform itself.

> **Important**
>
> Power BI had already become successful and entrenched within the organization before the business units were asked to undergo a formal migration effort away from the former BI platform.

Prepare to handle varying responses

In this large decentralized organization, there were varying levels of receptiveness and willingness to move to Power BI. Beyond concerns related to time and budget, there were staff who had made significant investments in building their skills in the former BI platform. So, the announcement about standardizing on Power BI wasn't news welcomed by everyone. Since each business unit has its own budget, individual business units could challenge decisions such as this one. As IT tool decisions were made centrally, that resulted in some challenges for the executive sponsor and BI leaders to handle.

> **Important**
>
> Communication with leadership teams throughout the business units was critical to ensure they all understood the high-level organizational benefits of standardizing on Power BI. Effective communication became even more essential as the migration progressed and the decommissioning date of the legacy BI platform approached.

Focus on the bigger picture

The company found that while some migrated reports could closely replicate the original design, not every individual report could be faithfully replicated in Power BI. Although it's to be expected—since all BI platforms are different. It did bring to light that a different design mindset was required.

Guidance was provided to content authors: focus on creating fit-for-purpose reports in Power BI, rather than attempt an exact replica of the legacy report. For this reason, subject matter experts need to be actively available during the migration process for consultation and validation. Efforts were taken to consider the report design purpose and to improve it when appropriate.

> **Important**
>
> Sometimes the better approach is to take on improvements during the migration. At other times, the better choice is to deliver the exact same value as before—without significant improvements—so as not to jeopardize the migration timeline.

Cautiously assess priorities

An analysis of the former BI platform was conducted to fully understand its usage. The former BI platform had thousands of published reports, of which approximately half had been accessed in the previous year. That number could be cut in half once again when assessing which reports were deemed to deliver significant value to the organization. Those reports were prioritized first for the migration.

> **Important**
>
> It's very easy to overestimate how critical a report actually is. For reports that aren't used frequently, evaluate whether they can be decommissioned entirely. Sometimes, the cheapest and easiest thing to do is nothing.

Cautiously assess complexity

Of the first prioritized reports, time estimates were compiled based on estimated effort levels: simple, medium, or complex. Although it sounds like a relatively straightforward process, don't expect time estimates to be accurate on an individual report basis. You might find an estimate can be wildly inaccurate. For example, the company had a report that it deemed highly complex. It received a conversion estimate of 50 days by the consultants. However, the redesigned report in Power BI was completed in about 50 hours.

> **Important**
>
> Although time estimates are often necessary to obtain funding and personnel assignments, they're probably most valuable in the aggregate.

Decide how change management is handled

With such a high volume of BI assets, change management for the business-owned reports represented a challenge. IT-managed reports were handled according to standard change management practices. However, due to the high volume, driving change centrally for business-owned content wasn't possible.

> **Important**
>
> Additional responsibility falls to the business units when it's impractical to manage change from one central team.

Create an internal community

The company established a Center of Excellence (COE) to provide internal training classes and resources. The COE also serves as an internal consultancy group that's ready to assist content authors with technical issues, resolution of roadblocks, and best practice guidance.

There's also an internal Power BI community, which has been a massive success counting in excess of 1,600 members. The community is managed in Yammer. Members can ask internally relevant questions and receive answers adhering to best practices and framed within organizational constraints. This type of user-to-user interaction alleviates much of the support burden from the COE. However, the COE does monitor the questions and answers, and it gets involved in conversations when appropriate.

An extension of the internal community is the newer Power BI expert network. It includes a small number of pre-selected Power BI champions from within the organization. They are highly skilled Power BI practitioners from the business units, who are enthusiastic champions, and who actively want to solve challenges within the business. Members of the Power BI expert network are expected to abide by best practices and guidelines established by the COE, and help the broader internal Power BI community understand and implement them. Although the Power BI expert network collaborates with the COE, and can receive dedicated training, Power BI experts operate independently from the COE. Each Power BI expert can define the parameters for how they operate, bearing in mind they have other responsibilities and priorities in their official role.

> **ⓘ Important**
>
> Have a very well defined scope for what the COE does, such as: adoption, governance, guidance, best practices, training, support, and perhaps even hands-on development. While a COE is incredibly valuable, measuring its return on investment can be difficult.

Monitor migration progress and success

Key performance indicators (KPIs) are continually monitored during the migration to Power BI. They help the company to understand trends for metrics such as number of report visits, number of active reports, and distinct users per month. Increased usage of Power BI is measured alongside decreased usage of the former BI platform, with the goal of achieving an inverse relationship. Targets are updated each month to adapt to

changes. If usage isn't happening at the desired pace, bottlenecks are identified so appropriate action can be taken.

> **Important**
>
> Create a migration scorecard with actionable business intelligence to monitor the success of the migration effort.

Large transportation and logistics company

A large North American transportation and logistics company is actively investing in the modernization of its data infrastructure and analytical systems.

Allow a period of gradual growth

The company started using Power BI in 2018. By mid-2019, Power BI became the preferred platform for all new BI use cases. Then, in 2020, the company focused on phasing out their existing BI platform, in addition to a variety of custom developed ASP.NET BI solutions.

> **Important**
>
> Power BI had many active users across the organization before commencing the phase out of their legacy BI platform and solutions.

Balance centralized and distributed groups

In the company, there are two types of BI teams: a central BI team and analytics groups distributed throughout the organization. The central BI team has ownership responsibility for Power BI as a platform, but it doesn't own any of the content. This way, the central BI team is a technical enablement hub that supports the distributed analytics groups.

Each of the analytics groups is dedicated to a specific business unit or a shared services function. A small group might contain a single analyst, while a larger group could have 10-15 analysts.

> **Important**

The distributed analytics groups comprise subject matter experts who are familiar with the day-to-day business needs. This separation allows the central BI team to focus primarily on technical enablement and support of the BI services and tools.

Focus on semantic model reusability

Relying on custom ASP.NET BI solutions was a barrier to developing new BI solutions. The required skill set meant that the number of self-service content authors was small. Because Power BI is a much more approachable tool—specifically designed for self-service BI—it spread quickly throughout the organization once it was released.

The empowerment of data analysts within the company resulted in immediate positive outcomes. However, the initial focus with Power BI development was on visualization. While it resulted in valuable BI solutions, this focus resulted in a large number of Power BI Desktop files, each with a one-to-one relationship between the report and its semantic model. It resulted in many semantic models and duplication of data and business logic. To reduce duplication of data, logic, and effort, the company delivered training and provided support to content authors.

> **ⓘ Important**
>
> Include information about the importance of data reusability in your internal training efforts. Address important concepts as early as is practical.

Test data access multiple ways

The company's data warehouse platform is DB2. Based on the current data warehouse design, the company found that DirectQuery models—instead of Import models—worked best for their requirements.

> **ⓘ Important**
>
> Conduct a technical proof of concept to evaluate the model storage mode that works best. Also, teach data modelers about model storage modes and how they can choose an appropriate mode for their project.

Educate authors about Premium licensing

Because it was easier to get started with Power BI (compared with their legacy BI platform), many of the early adopters were people who didn't have a license to the previous BI tool. As expected, the number of content authors grew considerably. These content authors understandably wanted to share their content with others, resulting in a continual need for additional Power BI Pro licenses.

The company made a large investment in Premium workspaces, most notably to distribute Power BI content to many users with Fabric free licenses. The support team works with content authors to ensure they use Premium workspaces when appropriate. It avoids unnecessarily allocating Power BI Pro licenses when a user only needs to consume content.

> **Important**
>
> Licensing questions often arise. Be prepared to educate and help content authors to address licensing questions. Validate that user requests for Power BI Pro licenses are justified.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Understand the data gateways

Early on, the company had many personal gateways. Using an on-premises data gateway cluster shifts the management efforts to the central BI team, which allows the content author community to focus on producing content. The central BI team worked with the internal Power BI user community to reduce the number of personal gateways.

> **Important**

Have a plan for creating and managing on-premises data gateways. Decide who is permitted to install and use a personal gateway and enforce it with gateway policies. For more information, see Manage gateways.

Formalize your support plan

As the adoption of Power BI grew within the organization, the company found that a multi-tier support approach worked well:

- **Layer 1: Intra-team**: People learn from, and teach, each other on a day-to-day basis.
- **Layer 2: Power BI community**: People ask questions of the internal Teams community to learn from each other and communicate important information.
- **Layer 3: Central BI team and COE**: People submit email requests for assistance. *Office hour* sessions are held twice per week to collectively discuss problems and share ideas.

> **Important**
>
> Although the first two layers are less formal, they're equally important as the third layer of support. Experienced users tend to rely mostly on people they know, whereas newer users (or those who are the single data analyst for a business unit or shared service) tend to rely more on formal support.

Invest in training and governance

Over the past year, the company improved its internal training offerings and enhanced its data governance program. The governance committee includes key members from each of the distributed analytics groups, plus the COE.

There are now six internal Power BI courses in their internal catalog. The Dashboard in a Day course remains a popular course for beginners. To help users deepen their skills, they deliver a series of three Power BI courses and two DAX courses.

One of their most important data governance decisions related to management of Premium capacities. The company opted to align their capacity with key analytics areas in business units and shared services. Therefore, if inefficiencies exist, the impact is felt only within that one area, and the decentralized capacity administrators are empowered to manage the capacity as they see fit.

> **Important**
>
> Pay attention to how Premium capacities are used, and how workspaces are assigned to them.

Related content

Other helpful resources include:

- Microsoft's BI transformation
- Power BI implementation planning
- Dashboard in a Day
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap

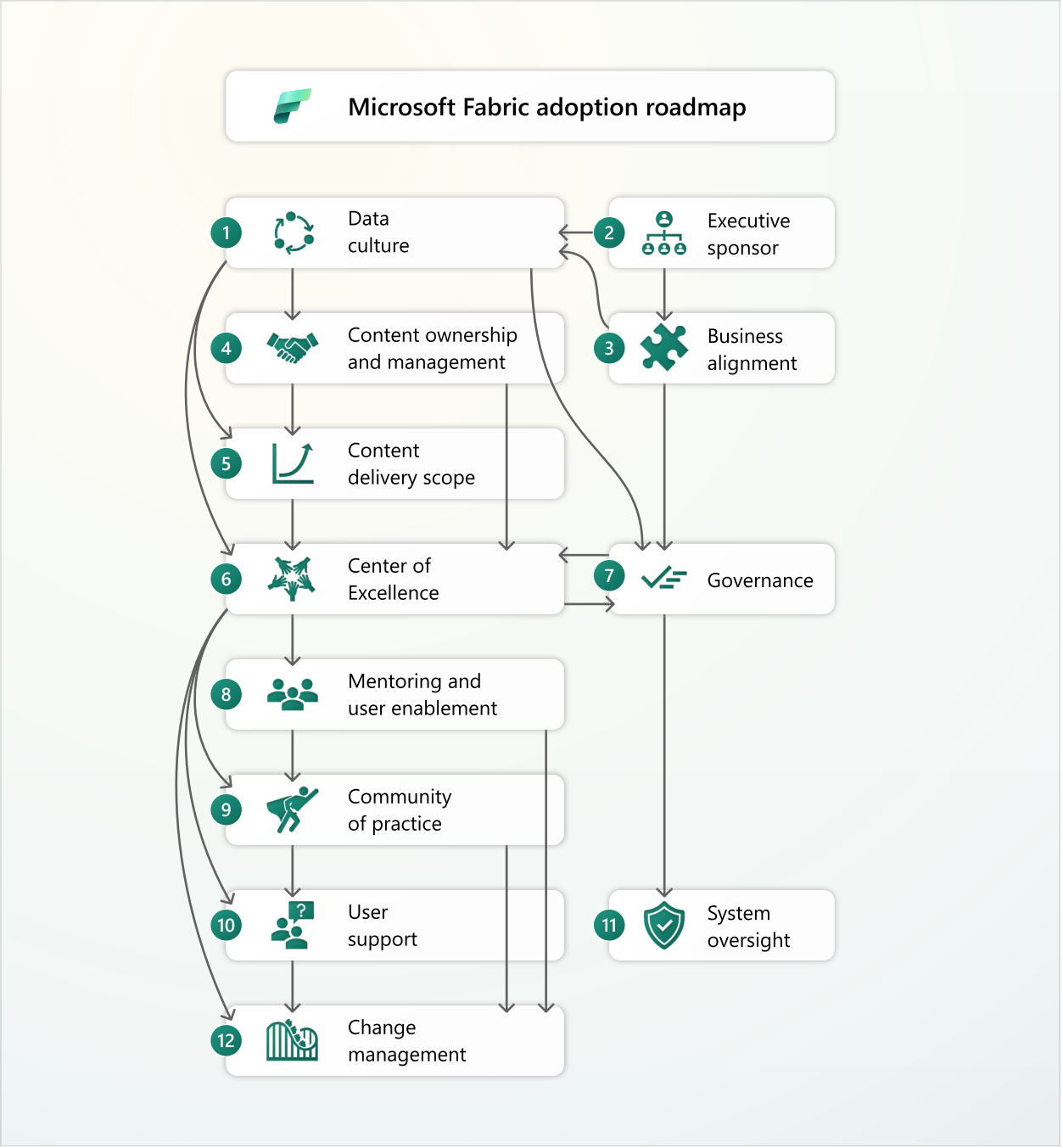
Article • 12/30/2024

The goal of this series of articles is to provide a roadmap. The roadmap presents a series of strategic and tactical considerations and action items that lead to the successful adoption of Microsoft Fabric, and help build a data culture in your organization.

Advancing adoption and cultivating a data culture is about more than implementing technology features. Technology can assist an organization in making the greatest impact, but a healthy data culture involves many considerations across the spectrum of people, processes, and technology.

> **Note**
>
> While reading this series of articles, we recommended that you also take into consideration **Power BI implementation planning** guidance. After you're familiar with the concepts in the Microsoft Fabric adoption roadmap, consider reviewing the **usage scenarios**. Understanding the diverse ways Power BI is used can influence your implementation strategies and decisions for all of Microsoft Fabric.

The diagram depicts the following areas of the Microsoft Fabric adoption roadmap.



The areas in the above diagram include:

Expand table

Area	Description
1	**Data culture**: Data culture refers to a set of behaviors and norms in the organization that encourages a data-driven culture. Building a data culture is closely related to adopting Fabric, and it's often a key aspect of an organization's digital transformation.
2	**Executive sponsor**: An executive sponsor is someone with credibility, influence, and authority throughout the organization. They advocate for building a data culture and adopting Fabric.
3	**Business Alignment**: How well the data culture and data strategy enable business users to achieve business objectives. An effective BI data strategy aligns with the business strategy.

Area	Description
4	**Content ownership and management**: There are three primary strategies for how business intelligence (BI) and analytics content is owned and managed: business-led self-service BI, managed self-service BI, and enterprise BI. These strategies have a significant influence on adoption, governance, and the Center of Excellence (COE) operating model.
5	**Content delivery scope**: There are four primary strategies for content and data delivery: personal, team, departmental, and enterprise. These strategies have a significant influence on adoption, governance, and the COE operating model.
6	**Center of Excellence**: A Fabric COE is an internal team of technical and business experts. These experts actively assist others who are working with data within the organization. The COE forms the nucleus of the broader community to advance adoption goals that are aligned with the data culture vision.
7	**Governance**: Data governance is a set of policies and procedures that define the ways in which an organization wants data to be used. When adopting Fabric, the goal of governance is to empower the internal user community to the greatest extent possible, while adhering to industry, governmental, and contractual requirements and regulations.
8	**Mentoring and user enablement**: A critical objective for adoption efforts is to enable users to accomplish as much as they can within the guardrails established by governance guidelines and policies. The act of mentoring users is one of the most important responsibilities of the COE. It has a direct influence on adoption efforts.
9	**Community of practice**: A community of practice comprises a group of people with a common interest, who interact with and help each other on a voluntary basis. An active community is an indicator of a healthy data culture. It can significantly advance adoption efforts.
10	**User support**: User support includes both informally organized and formally organized methods of resolving issues and answering questions. Both formal and informal support methods are critical for adoption.
11	**System oversight**: System oversight includes the day-to-day administration responsibilities to support the internal processes, tools, and people.
12	**Change management**: Change management involves procedures to address the impact of change for people in an organization. These procedures safeguard against disruption and productivity loss due to changes in solutions or processes. An effective data strategy describes who is responsible for managing this change and the practices and resources needed to realize it.

The relationships in the above diagram can be summarized as follows.

- Your organizational **data culture** vision will strongly influence the strategies that you follow for self-service and enterprise **content ownership and management** and **content delivery scope**.

- These strategies will, in turn, have a big impact on the operating model for your **Center of Excellence** and governance decisions.
- The established **governance** guidelines, policies, and processes affect the implementation methods used for **mentoring and enablement**, the **community of practice**, and **user support**.
- Governance decisions will dictate the day-to-day **system oversight** (administration) activities.
- Adoption and governance decisions are implemented alongside **change management** to mitigate the impact and disruption of change on existing business processes.
- All data culture and adoption-related decisions and actions are accomplished more easily with guidance and leadership from an **executive sponsor**, who facilitates **business alignment** between the business strategy and data strategy. This alignment in turn informs data culture and governance decisions.

Each individual article in this series discusses key topics associated with the items in the diagram. Considerations and potential action items are provided. Each article concludes with a set of maturity levels to help you assess your current state so you can decide what action to take next.

Microsoft Fabric adoption

Successful adoption of analytical tools like Fabric involves making effective processes, support, tools, and data available and integrated into regular ongoing patterns of usage for content creators, consumers, and stakeholders in the organization.

> **ⓘ Important**
>
> This series of adoption articles is focused on *organizational* adoption. See Microsoft Fabric adoption maturity levels for an introduction to the three types of adoption: organizational, user, and solution.

A common misconception is that adoption relates primarily to usage or the number of users. There's no question that usage statistics are an important factor. However, usage isn't the only factor. Adoption isn't just about *using* the technology regularly; it's about using it *effectively*. Effectiveness is much more difficult to define and measure.

Whenever possible, adoption efforts should be aligned across analytics platforms and BI services.

> **Note**
>
> Individuals—and the organization itself—are continually learning, changing, and improving. That means there's no formal end to adoption-related efforts.

The remaining articles in this Power BI adoption series discuss the following aspects of adoption.

- Adoption maturity levels
- Data culture
- Executive sponsorship
- Business alignment
- Content ownership and management
- Content delivery scope
- Center of Excellence
- Governance
- Mentoring and enablement
- Community of practice
- User support
- System oversight
- Change management
- Conclusion and additional resources

> **Important**
>
> You might be wondering how this Fabric adoption roadmap is different from the **Power BI adoption framework**. The adoption framework was created primarily to support Microsoft partners. It's a lightweight set of resources to help partners deploy Power BI solutions for their customers.
>
> This adoption series is more current. It's intended to guide any person or organization that is using—or considering using—Fabric. If you're seeking to improve your existing Power BI of Fabric implementation, or planning a new Power BI or Fabric implementation, this adoption roadmap is a great place to start.

Target audience

The intended audience of this series of articles is interested in one or more of the following outcomes.

- Improving their organization's ability to effectively use analytics.
- Increasing their organization's maturity level related to the delivery of analytics.
- Understanding and overcoming adoption-related challenges faced when scaling and growing.
- Increasing their organization's return on investment (ROI) in data and analytics.

This series of articles will be most helpful to those who work in an organization with one or more of the following characteristics.

- Power BI or other Fabric workloads are deployed with some successes.
- There are pockets of viral adoption, but analytics isn't being purposefully governed across the entire organization.
- Analytics solutions are deployed with some meaningful scale, but there remains a need to determine:
  - What is effective and what should be maintained.
  - What should be improved.
  - How future deployments could be more strategic.
- An expanded implementation of analytics is under consideration or is planned.

This series of articles will also be helpful for:

- Organizations that are in the early stages of an analytics implementation.
- Organizations that have had success with adoption and now want to evaluate their current maturity level.

Assumptions and scope

The primary focus of this series of articles is on the Microsoft Fabric platform.

To fully benefit from the information provided in these articles, you should have an understanding of Power BI foundational concepts and Fabric foundational concepts.

Related content

In the next article in this series, learn about the Fabric adoption maturity levels. The maturity levels are referenced throughout the entire series of articles. Also, see the conclusion article for additional adoption-related resources.

Other helpful resources include:

- Power BI implementation planning
- Questions? Try asking the Fabric Community

- Suggestions? Contribute ideas to improve Fabric

Experienced partners are available to help your organization succeed with adoption initiatives. To engage with a partner, visit the Power BI partner portal.

Acknowledgments

The Microsoft Fabric adoption roadmap articles are written by Melissa Coates, Kurt Buhler, and Peter Myers. Matthew Roche, from the Fabric Customer Advisory Team, provides strategic guidance and feedback to the subject matter experts. Reviewers include Cory Moore, James Ward, Timothy Bindas, Greg Moir, and Chuy Varela.

Feedback

Was this page helpful? Yes No

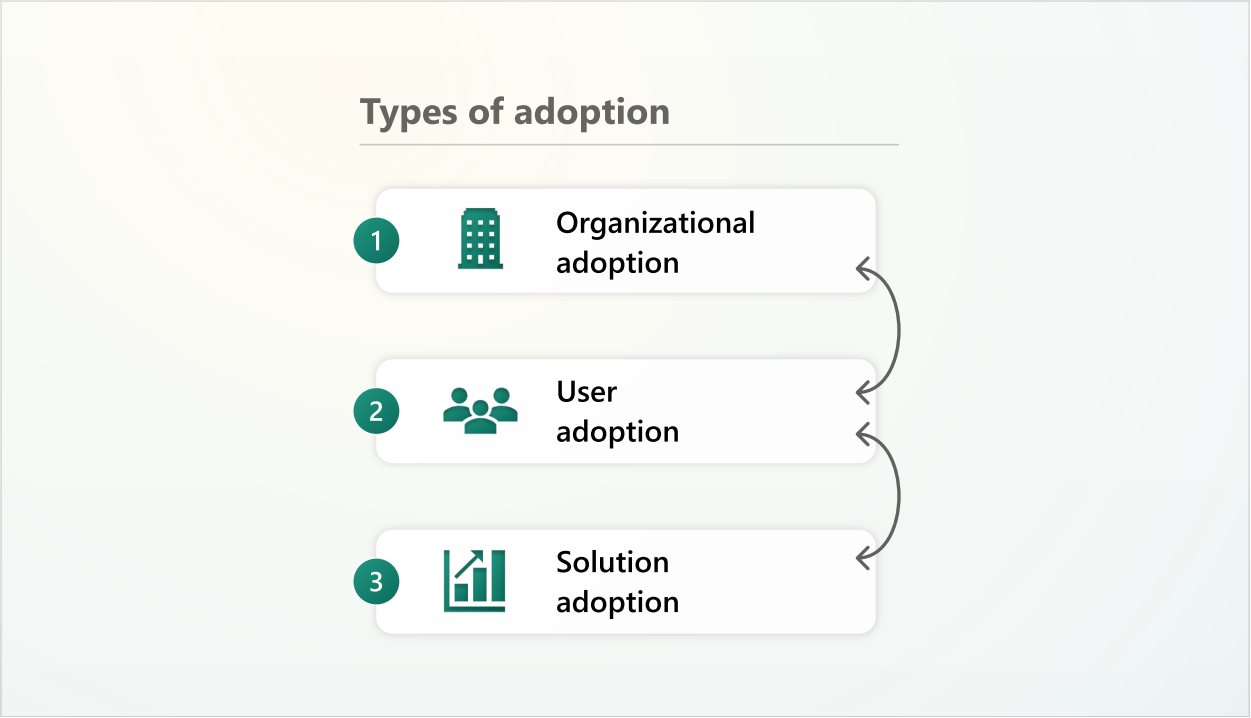
Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap maturity levels

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

There are three inter-related perspectives to consider when adopting an analytics technology like Microsoft Fabric.



The three types of adoption depicted in the above diagram include:

Expand table

Type	Description
1	**Organizational adoption** refers to the effectiveness of your analytics governance processes. It also refers to data management practices that support and enable analytics and business intelligence (BI) efforts.
2	**User adoption** is the extent to which consumers and creators continually increase their knowledge. It's concerned with whether they're actively using analytics tools, and whether they're using them in the most effective way.

Type	Description
3	**Solution adoption** refers to the impact and business value achieved for individual requirements and analytical solutions.

As the four arrows in the previous diagram indicate, the three types of adoption are all strongly inter-related:

- **Solution adoption affects user adoption.** A well-designed and well-managed solution—which could be many things, such as a set of reports, a Power BI app, a semantic model, or a Fabric lakehouse—impacts and guides users on how to use analytics in an optimal way.
- **User adoption impacts organizational adoption.** The patterns and practices used by individual users influence organizational adoption decisions, policies, and practices.
- **Organizational adoption influences user adoption.** Effective organizational practices—including mentoring, training, support, and community—encourage users to do the right thing in their day-to-day workflow.
- **User adoption affects solution adoption.** Stronger user adoption, because of the effective use of analytics by educated and informed users, contributes to stronger and more successful individual solutions.

The remainder of this article introduces the three types of adoption in more detail.

Organizational adoption maturity levels

Organizational adoption measures the state of analytics governance and data management practices. There are several organizational adoption goals:

- Effectively support the community of creators, consumers, and stakeholders
- Enable and empower users
- Right-sized governance of analytics, BI, and data management activities
- Oversee information delivery via enterprise BI and self-service BI with continuous improvement cycles

It's helpful to think about organizational adoption from the perspective of a maturity model. For consistency with the Power CAT adoption maturity model and the maturity model for Microsoft 365, this Microsoft Fabric adoption roadmap aligns with the five levels from the Capability Maturity Model, which were later enhanced by the Data Management Maturity (DMM) model from ISACA (note that the DMM was a paid resource that has since been retired).

Every organization has limited time, funding, and people. So, it requires them to be selective about where they prioritize their efforts. To get the most from your investment in analytics, seek to attain at least maturity level 300 or 400, as discussed below. It's common that different business units in the organization evolve and mature at different rates, so be conscious of the organizational state as well as progress for key business units.

> **Note**
>
> Organizational adoption maturity is a long journey. It takes time, effort, and planning to progress to the higher levels.

Maturity level 100 – Initial

Level 100 is referred to as *initial* or *performed*. It's the starting point for new data-related investments that are new, undocumented, and without any process discipline.

Common characteristics of maturity level 100 include:

- Pockets of success and experimentation with Fabric exist in one or more areas of the organization.
- Achieving quick wins has been a priority, and solutions have been delivered with some success.
- Organic growth has led to the lack of a coordinated strategy or governance approach.
- Practices are undocumented, with significant reliance on tribal knowledge.
- There are few formal processes in place for effective data management.
- Risk exists due to a lack of awareness of how data is used throughout the organization.
- The potential for a strategic investment with analytics is acknowledged. However, there's no clear path forward for purposeful, organization-wide execution.

Maturity level 200 – Repeatable

Level 200 is referred to as *repeatable* or *managed*. At this point on the maturity curve, data management is planned and executed. Defined processes exist, though these processes might not apply uniformly throughout the organization.

Common characteristics of maturity level 200 include:

- Certain analytics content is now critical in importance and/or it's broadly used by the organization.
- There are attempts to document and define repeatable practices. These efforts are siloed, reactive, and deliver varying levels of success.
- There's an over-reliance on individuals having good judgment and adopting healthy habits that they learned on their own.
- Analytics adoption continues to grow organically and produces value. However, it takes place in an uncontrolled way.
- Resources for an internal community are established, such as a Teams channel or Yammer group.
- Initial planning for a consistent analytics governance strategy is underway.
- There's recognition that a Center of Excellence (COE) can deliver value.

Maturity level 300 – Defined

Level 300 is referred to as *defined*. At this point on the maturity curve, a set of standardized data management processes are established and consistently applied across organizational boundaries.

Common characteristics of maturity level 300 include:

- Measurable success is achieved for the effective use of analytics.
- Progress is made on the standardization of repeatable practices. However, less-than-optimal aspects could still exist due to early uncontrolled growth.
- The COE is established. It has clear goals and scope of responsibilities.
- The internal community of practice gains traction with the participation of a growing number of users.
- Champions emerge in the internal user community.
- Initial investments in training, documentation, and resources (such as template files) are made.
- An initial governance model is in place.
- There's an active and engaged executive sponsor.
- Roles and responsibilities for all analytics stakeholders are well understood.

Maturity level 400 – Capable

Level 400 is known as *capable* or *measured*. At this point on the maturity curve, data is well-managed across its entire lifecycle.

Common characteristics of maturity level 400 include:

- Analytics and business intelligence efforts deliver significant value.

- Approved tools are commonly used for delivering critical content throughout the organization.
- There's an established and accepted governance model with cooperation from all key business units.
- Training, documentation, and resources are readily available for, and actively used by, the internal community of users.
- Standardized processes are in place for the oversight and monitoring of analytics usage and practices.
- The COE includes representation from all key business units.
- A champions network supports the internal community. The champions actively work with their colleagues as well as the COE.

Maturity level 500 – Efficient

Level 500 is known as *efficient* or *optimizing* because at this point on the maturity curve, the emphasis is now on automation and continuous improvement.

Common characteristics of maturity level 500 include:

- The value of analytics solutions is prevalent in the organization. Fabric is widely accepted throughout the organization.
- Analytics skillsets are highly valued in the organization, and they're recognized by leadership.
- The internal user community is self-sustaining, with support from the COE. The community isn't over-reliant on key individuals.
- The COE reviews key performance indicators regularly to measure success of implementation and adoption goals.
- Continuous improvement is a continual priority.
- Use of automation adds value, improves productivity, or reduces risk for error.

> **Note**
>
> The characteristics above are generalized. When considering maturity levels and designing a plan, you'll want to consider each topic or goal independently. In reality, it's probably not possible to reach level 500 maturity level for every aspect of Fabric adoption for the entire organization. So, assess maturity levels independently per goal. That way, you can prioritize your efforts where they will deliver the most value. The remainder of the articles in this Fabric adoption series present maturity levels on a per-topic basis.

Individuals—and the organization itself—continually learn, change, and improve. That means there's no formal end to adoption-related efforts. However, it's common that effort is reduced as higher maturity levels are reached.

The remainder of this article introduces the second and third types of adoption: user adoption and solution adoption.

> **Note**
>
> The remaining articles in this series focus primarily on organizational adoption.

User adoption stages

User adoption measures the extent to which content consumers and self-service content creators are actively and effectively using analytics tools such as Fabric. Usage statistics alone don't indicate successful user adoption. User adoption is also concerned with individual user behaviors and practices. The aim is to ensure users engage with solutions, tools, and processes in the correct way and to their fullest extent.

User adoption encompasses how consumers view content, as well as how self-service creators generate content for others to consume.

User adoption occurs on an individual user basis, but it's measured and analyzed in the aggregate. Individual users progress through the four stages of user adoption at their own pace. An individual who adopts a new technology will take some time to achieve proficiency. Some users will be eager; others will be reluctant to learn yet another tool, regardless of the promised productivity improvements. Advancing through the user adoption stages involves time and effort, and it involves behavioral changes to become aligned with organizational adoption objectives. The extent to which the organization supports users advancing through the user adoption stages has a direct correlation to the organizational-level adoption maturity.

User adoption stage 1 – Awareness

Common characteristics of stage 1 user adoption include:

- An individual has heard of, or been initially exposed to, analytics in some way.
- An individual might have access to a tool, such as Fabric, but isn't yet actively using it.

User adoption stage 2 – Understanding

Common characteristics of stage 2 user adoption include:

- An individual develops understanding of the benefits of analytics and how it can support decision-making.
- An individual shows interest and starts to use analytics tools.

User adoption stage 3 – Momentum

Common characteristics of stage 3 user adoption include:

- An individual actively gains analytics skills by attending formal training, self-directed learning, or experimentation.
- An individual gains basic competency by using or creating analytics relevant to their role.

User adoption stage 4 – Proficiency

Common characteristics of stage 4 user adoption include:

- An individual actively uses analytics regularly.
- An individual understands how to use analytic tools in the way in which they were intended, as relevant for their role.
- An individual modifies their behavior and activities to align with organizational governance processes.
- An individual's willingness to support organizational processes and change efforts is growing over time, and they become an advocate for analytics in the organization.
- An individual makes the effort to continually improve their skills and stay current with new product capabilities and features.

It's easy to underestimate the effort it takes to progress from stage 2 (understanding) to stage 4 (proficiency). Typically, it takes the longest time to progress from stage 3 (momentum) to stage 4 (proficiency).

> **Important**
>
> By the time a user reaches the momentum and proficiency stages, the organization needs to be ready to support them in their efforts. You can consider some proactive efforts to encourage users to progress through stages. For more information, see the community of practice and the user support articles.

Solution adoption phases

Solution adoption is concerned with measuring the impact of content that's been deployed. It's also concerned with the level of value solutions provide. The scope for evaluating solution adoption is for one set of requirements, like a set of reports, a lakehouse, or a single Power BI app.

> **Note**
>
> In this series of articles, *content* is synonymous with *solution*.

As a solution progresses to phases 3 or 4, expectations to operationalize the solution are higher.

> **Tip**
>
> The importance of scope on expectations for governance is described in the **content delivery scope** article. That concept is closely related to this topic, but this article approaches it from a different angle. It considers when you already have a solution that is operationalized and distributed to many users. That doesn't immediately equate to phase 4 solution adoption, as the concept of solution adoption focuses on how much value the content delivers.

Solution phase 1 – Exploration

Common characteristics of phase 1 solution adoption include:

- Exploration and experimentation are the main approaches to testing out new ideas. Exploration of new ideas can occur through informal self-service efforts, or through a formal proof of concept (POC), which is purposely narrow in scope. The goal is to confirm requirements, validate assumptions, address unknowns, and mitigate risks.
- A small group of users test the proof of concept solution and provide useful feedback.
- For simplicity, all exploration—and initial feedback—could occur within local user tools (such as Power BI Desktop or Excel) or within a single Fabric workspace.

Solution phase 2 – Functional

Common characteristics of phase 2 solution adoption include:

- The solution is functional and meets the basic set of user requirements. There are likely plans to iterate on improvements and enhancements.
- The solution is deployed to the Fabric portal.
- All necessary supporting components are in place (for example, a gateway to support scheduled data refresh).
- Target users are aware of the solution and show interest in using it. Potentially, it could be a limited preview release, and might not yet be ready to promote to a production workspace.

Solution phase 3 – Valuable

Common characteristics of phase 3 solution adoption include:

- Target users find the solution to be valuable and experience tangible benefits.
- The solution is promoted to a production workspace that's managed, secured, and audited.
- Validations and testing occur to ensure data quality, accurate presentation, accessibility, and acceptable performance.
- Content is endorsed, when appropriate.
- Usage metrics for the solution are actively monitored.
- User feedback loops are in place to facilitate suggestions and improvements that can contribute to future releases.
- Solution documentation is generated to support the needs of information consumers (such as data sources used or how metrics are calculated). The documentation helps future content creators (for example, for documenting any future maintenance or planned enhancements).
- Ownership and subject matter experts for the content are clear.
- Report branding and theming are in place and in line with governance guidelines.

Solution phase 4 – Essential

Common characteristics of phase 4 solution adoption include:

- Target users actively and routinely use the solution, and it's considered essential for decision-making purposes.
- The solution resides in a production workspace well separated from development and test content. Change management and release management are carefully controlled due to the impact of changes.
- A subset of users regularly provides feedback to ensure the solution continues to meet evolving requirements.
- Expectations for the success of the solution are clear and are measured.

- Expectations for support of the solution are clear, especially if there are service level agreements.
- The solution aligns with organizational governance guidelines and practices.
- Most content is certified due to its critical nature.
- Formal user acceptance testing for new changes might occur, particularly for IT-managed content.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about the organizational data culture and its impact on adoption efforts.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Data culture

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

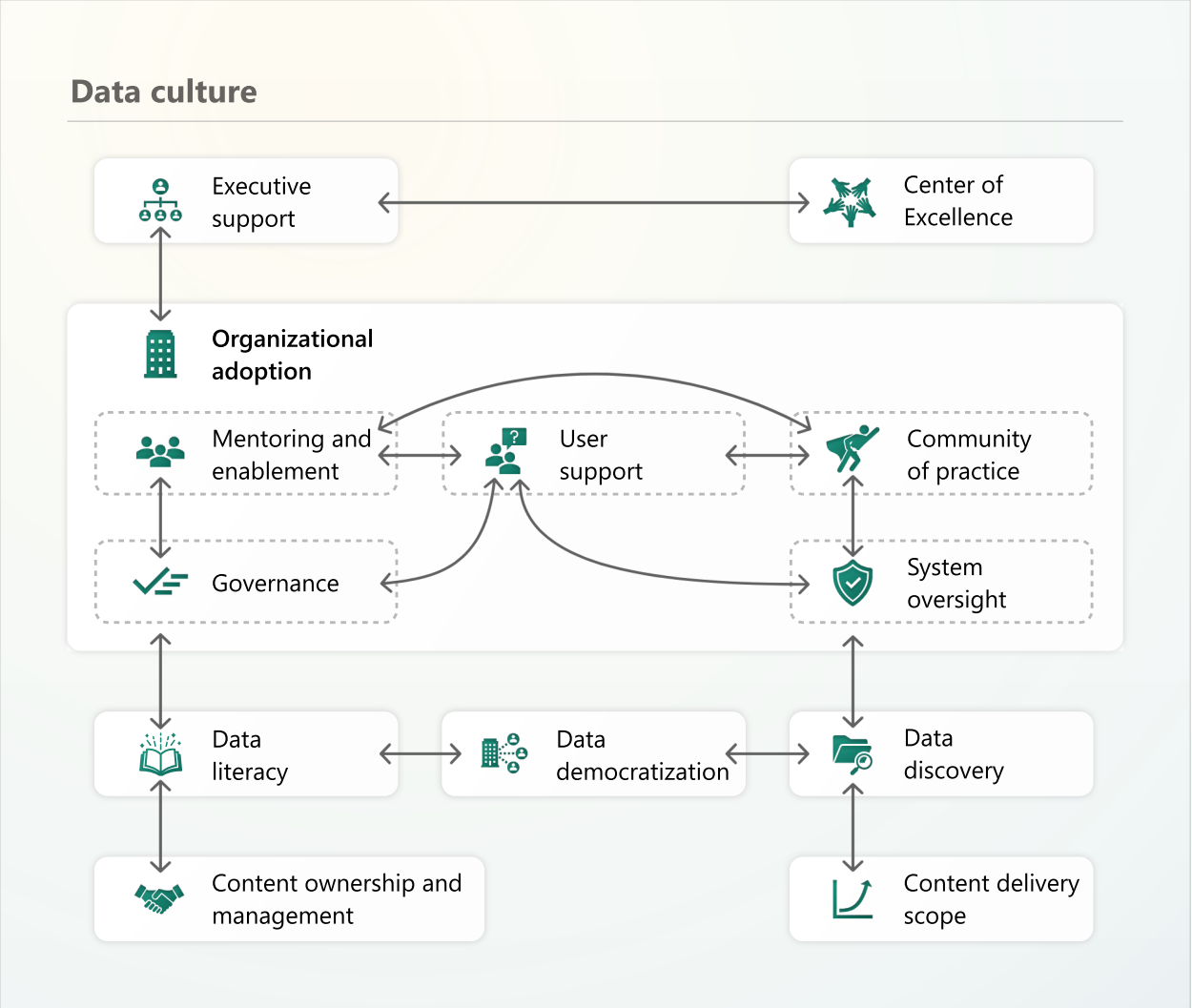
Building a data culture is closely related to adopting analytics, and it's often a key aspect of an organization's digital transformation. The term *data culture* can be defined in different ways by different organizations. In this series of articles, data culture means a set of behaviors and norms in an organization. It encourages a culture that regularly employs informed data decision-making:

- By more stakeholders throughout more areas of the organization.
- Based on analytics, not opinion.
- In an effective, efficient way that's based on best practices approved by the Center of Excellence (COE).
- Based on trusted data.
- That reduces reliance on undocumented tribal knowledge.
- That reduces reliance on hunches and gut decisions.

> **Important**
>
> Think of data culture as what you do, not what you say. Your data culture is not a set of rules (that's governance). So, data culture is a somewhat abstract concept. It's the behaviors and norms that are allowed, rewarded, and encouraged—or those that are disallowed and discouraged. Bear in mind that a healthy data culture motivates employees at all levels of the organization to generate and distribute actionable knowledge.

Within an organization, certain business units or teams are likely to have their own behaviors and norms for getting things done. The specific ways to achieve data culture objectives can vary across organizational boundaries. What's important is that they should all align with the organizational data culture objectives. You can think of this structure as *aligned autonomy*.

The following circular diagram conveys the interrelated aspects that influence your data culture:



The diagram depicts the somewhat ambiguous relationships among the following items:

- Data culture is the outer circle. All topics within it contribute to the state of the data culture.
- Organizational adoption (including the implementation aspects of mentoring and user enablement, user support, community of practice, governance, and system oversight) is the inner circle. All topics are major contributors to the data culture.
- Executive support and the Center of Excellence are drivers for the success of organizational adoption.
- Data literacy, data democratization, and data discovery are data culture aspects that are heavily influenced by organizational adoption.
- Content ownership and management, and content delivery scope, are closely related to data democratization.

The elements of the diagram are discussed throughout this series of articles.

Data culture vision

The concept of data culture can be difficult to define and measure. Even though it's challenging to articulate data culture in a way that's meaningful, actionable, and measurable, you need to have a well-understood definition of what a healthy data culture means to your organization. This vision of a healthy data culture should:

- Originate from the executive level.
- Align with organizational objectives.
- Directly influence your adoption strategy.
- Serve as the high-level guiding principles for enacting governance policies and guidelines.

Data culture outcomes aren't specifically mandated. Rather, the state of the data culture is the result of following the governance rules as they're enforced (or the lack of governance rules). Leaders at all levels need to actively demonstrate through their actions what's important to them, including how they praise, recognize, and reward staff members who take initiative.

> **Tip**
>
> If you can take for granted that your efforts to develop a data solution (such as a semantic model, a lakehouse, or a report) will be valued and appreciated, that's an excellent indicator of a healthy data culture. Sometimes, however, it depends on what your immediate manager values most.

The initial motivation for establishing a data culture often comes from a specific strategic business problem or initiative. It might be:

- A reactive change, such as responding to new agile competition.
- A proactive change, such as starting a new line of business or expanding into new markets to seize a "green field" opportunity. Being data driven from the beginning can be relatively easier when there are fewer constraints and complications, compared with an established organization.
- Driven by external changes, such as pressure to eliminate inefficiencies and redundancies during an economic downturn.

In each of these situations, there's often a specific area where the data culture takes root. The specific area could be a scope of effort that's smaller than the entire organization, even if it's still significant. After necessary changes are made at this smaller scope, they can be incrementally replicated and adapted for the rest of the organization.

Although technology can help advance the goals of a data culture, implementing specific tools or features isn't the objective. This series of articles covers a lot of topics that contribute to adoption of a healthy data culture. The remainder of this article addresses three essential aspects of data culture: data discovery, data democratization, and data literacy.

Data discovery

A successful data culture depends on users working with the right data in their day-to-day activities. To achieve this goal, users need to find and access data sources, reports, and other items.

Data discovery is the ability to effectively locate relevant data assets across the organization. Primarily, data discovery is concerned with improving awareness that data exists, which can be particularly challenging when data is siloed in departmental systems.

Data discovery is a slightly different concept from *search*, because:

- Data discovery allows users to see metadata for an item, like the name of a semantic model, even if they don't currently have access to it. After a user is aware of its existence, that user can go through the standard process to request access to the item.
- Search allows users to locate an existing item when they already have security access to the item.

> **Tip**
>
> It's important to have a clear and simple process so users can request access to data. Knowing that data exists—but being unable to access it within the guidelines and processes that the domain owner has established—can be a source of frustration for users. It can force them to use inefficient workarounds instead of requesting access through the proper channels.

Data discovery contributes to adoption efforts and the implementation of governance practices by:

- Encouraging the use of trusted high-quality data sources.
- Encouraging users to take advantage of existing investments in available data assets.

- Promoting the use and enrichment of existing data items (such as a lakehouse, data warehouse, data pipeline, dataflow, or semantic model) or reporting items (such as reports, dashboards, or metrics).
- Helping people understand who owns and manages data assets.
- Establishing connections between consumers, creators, and owners.

The OneLake catalog and the use of endorsements are key ways to promote data discovery in your organization.

Furthermore, data catalog solutions are extremely valuable tools for data discovery. They can record metadata tags and descriptions to provide deeper context and meaning. For example, Microsoft Purview can scan and catalog items from a Fabric tenant (as well as many other sources).

Questions to ask about data discovery

Use questions like those found below to assess data discovery.

- Is there a data catalog where business users can search for data?
- Is there a metadata catalog that describes definitions and data locations?
- Are high-quality data sources endorsed by certifying or promoting them?
- To what extent do redundant data sources exist because people can't find the data they need? What roles are expected to create data items? What roles are expected to create reports or perform ad hoc analysis?
- Can end users find and use existing reports, or do they insist on data exports to create their own?
- Do end users know which reports to use to address specific business questions or find specific data?
- Are people using the appropriate data sources and tools, or resisting them in favor of legacy ones?
- Do analysts understand how to enrich existing certified semantic models with new data—for example, by using a Power BI composite model?
- How consistent are data items in their quality, completeness, and naming conventions?
- Can data item owners follow data lineage to perform impact analysis of data items?

Maturity levels of data discovery

The following maturity levels can help you assess your current state of data discovery.

Expand table

Level	State of Fabric data discovery
100: Initial	• Data is fragmented and disorganized, with no clear structures or processes to find it. • Users struggle to find and use data they need for their tasks.
200: Repeatable	• Scattered or organic efforts to organize and document data are underway, but only in certain teams or departments. • Content is occasionally endorsed, but these endorsements aren't defined and the process isn't managed. Data remains siloed and fragmented, and it's difficult to access.
300: Defined	• A central repository, like the OneLake catalog, is used to make data easier to find for people who need it. • An explicit process is in place to endorse quality data and content. • Basic documentation includes catalog data, definitions, and calculations, as well as where to find them.
400: Capable	• Structured, consistent processes guide users how to endorse, document, and find data from a central hub. Data silos are the exception instead of the rule. • Quality data assets are consistently endorsed and easily identified. • Comprehensive data dictionaries are maintained and improve data discovery.
500: Efficient	• Data and metadata is systematically organized and documented with a full view of the data lineage. • Quality assets are endorsed and easily identified. • Cataloging tools, like Microsoft Purview, are used to make data discoverable for both use and governance.

Data democratization

Data democratization refers to putting data into the hands of more users who are responsible for solving business problems. It's about enabling more users to make better data-driven decisions.

> **Note**
>
> The concept of data democratization doesn't imply a lack of security or a lack of justification based on job role. As part of a healthy data culture, data democratization helps reduce *shadow IT* by providing semantic models that:
>
> - Are secured, governed, and well managed.
> - Meet business needs in cost-effective and timely ways.

Your organization's position on data democratization will have a wide-reaching impact on adoption and governance-related efforts.

> **Warning**
>
> If access to data or the ability to perform analytics is limited to a select number of individuals in the organization, that's typically a warning sign because the ability to work with data is a key characteristic of a healthy data culture.

Questions to ask about data democratization

Use questions like those found below to assess data democratization.

- Is data and analytics readily accessible, or restricted to limited roles and individuals?
- Is an effective process in place for people to request access to new data and tools?
- Is data readily shared between teams and business units, or is it siloed and closely guarded?
- Who is permitted to have Power BI Desktop installed?

- Who is permitted to have Power BI Pro or Power BI Premium Per User (PPU) licenses?
- Who is permitted to create assets in Fabric workspaces?
- What's the desired level of self-service analytics and business intelligence (BI) user enablement? How does this level vary depending on business unit or job role?
- What's the desired balance between enterprise and self-service analytics, and BI?
- What data sources are strongly preferred for what topics and business domains? What's the allowed use of unsanctioned data sources?
- Who can manage content? Is this decision different for data versus reports? Is the decision different for enterprise BI users versus decentralized users? Who can own and manage self-service BI content?
- Who can consume content? Is this decision different for external partners, customers, or suppliers?

Maturity levels of data democratization

The following maturity levels can help you assess your current state of data democratization.

Expand table

Level	State of data democratization
100: Initial	• Data and analytics are limited to a small number of roles, who gatekeep access to others. • Business users must request access to data or tools to complete tasks. They struggle with delays or bottlenecks. • Self-service initiatives are taking place with some success in various areas of the organization. These activities are occurring in a somewhat chaotic manner, with few formal processes and no strategic plan. There's a lack of oversight and visibility into these self-service activities. The success or failure of each solution isn't well understood. • The enterprise data team can't keep up with the needs of the business. A significant backlog of requests exists for this team.

Level	State of data democratization
200: Repeatable	• There are limited efforts underway to expand access to data and tools. • Multiple teams have had measurable success with self-service solutions. People in the organization are starting to pay attention. • Investments are being made to identify the ideal balance of enterprise and self-service solutions.
300: Defined	• Many people have access to the data and tools they need, although not all users are equally enabled or held accountable for the content they create. • Effective self-service data practices are incrementally and purposely replicated throughout more areas of the organization.
400: Capable	• Healthy partnerships exist among enterprise and self-service solution creators. Clear, realistic user accountability and policies mitigate risk of self-service analytics and BI. • Clear and consistent processes are in place for users to request access to data and tools. • Individuals who take initiative in building valuable solutions are recognized and rewarded.
500: Efficient	• User accountability and effective governance give central teams confidence in what users do with data. • Automated, monitored processes enable people to easily request access to data and tools. Anyone with the need or interest to use data can follow these processes to perform analytics.

Data literacy

Data literacy refers to the ability to interpret, create, and communicate with data and analytics accurately and effectively.

Training efforts, as described in the mentoring and user enablement article, often focus on how to use the technology itself. Technology skills are important to producing high-quality solutions, but it's also important to consider how to purposely advance data literacy throughout the organization. Put another way, successful adoption takes a lot more than merely providing software and licenses to users.

How you go about improving data literacy in your organization depends on many factors, such as current user skillsets, complexity of the data, and the types of analytics

that are required. You might choose to focus on these types of activities related to data literacy:

- Interpreting charts and graphs
- Assessing the validity of data
- Performing root cause analysis
- Discerning correlation from causation
- Understanding how context and outliers affect how results are presented
- Using storytelling to help consumers quickly understand and act

> **Tip**
>
> If you're struggling to get data culture or governance efforts approved, focusing on tangible benefits that you can achieve with data discovery ("find the data"), data democratization ("use the data"), or data literacy ("understand the data") can help. It can also be helpful to focus on specific problems that you can solve or mitigate through data culture advancements.
>
> Getting the right stakeholders to agree on the problem is usually the first step. Then, it's a matter of getting the stakeholders to agree on the strategic approach to a solution, along with the solution details.

Questions to ask about data literacy

Use questions like those found below to assess data literacy.

- Does a common analytical vocabulary exist in the organization to talk about data and BI solutions? Alternatively, are definitions fragmented and different across silos?
- How comfortable are people with making decisions based on data and evidence compared to intuition and subjective experience?
- When people who hold an opinion are confronted with conflicting evidence, how do they react? Do they critically appraise the data, or do they dismiss it? Can they alter their opinion, or do they become entrenched and resistant?
- Do training programs exist to support people in learning about data and analytical tools?

- Is there significant resistance to visual analytics and interactive reporting in favor of static spreadsheets?
- Are people open to new analytical methods and tools to potentially address their business questions more effectively? Alternatively, do they prefer to keep using existing methods and tools to save time and energy?
- Are there methods or programs to assess or improve data literacy in the organization? Does leadership have an accurate understanding of the data literacy levels?
- Are there roles, teams, or departments where data literacy is particularly strong or weak?

Maturity levels of data literacy

The following maturity levels can help you assess your current state of data literacy.

Expand table

Level	State of data literacy
100: Initial	• Decisions are frequently made based on intuition and subjective experience. When confronted with data that challenges existing opinions, data is often dismissed. • Individuals have low confidence to use and understand data in decision-making processes or discussions. • Report consumers have a strong preference for static tables. These consumers dismiss interactive visualizations or sophisticated analytical methods as "fancy" or unnecessary.
200: Repeatable	• Some teams and individuals inconsistently incorporate data into their decision making. There are clear cases where misinterpretation of data has led to flawed decisions or wrong conclusions. • There's some resistance when data challenges pre-existing beliefs. • Some people are skeptical of interactive visualizations and sophisticated analytical methods, though their use is increasing.

Level	State of data literacy
300: Defined	• The majority of teams and individuals understand data relevant to their business area and use it implicitly to inform decisions. • When data challenges pre-existing beliefs, it produces critical discussions and sometimes motivates change. • Visualizations and advanced analytics are more widely accepted, though not always used effectively.
400: Capable	• Data literacy is recognized explicitly as a necessary skill in the organization. Some training programs address data literacy. Specific efforts are taken to help departments, teams, or individuals that have particularly weak data literacy. • Most individuals can effectively use and apply data to make objectively better decisions and take actions. • Visual and analytical best practices are documented and followed in strategically important data solutions.
500: Efficient	• Data literacy, critical thinking, and continuous learning are strategic skills and values in the organization. Effective programs monitor progress to improve data literacy in the organization. • Decision making is driven by data across the organization. Decision intelligence or prescriptive analytics are used to recommend key decisions and actions. • Visual and analytical best practices are seen as essential to generate business value with data.

Considerations and key actions

Checklist - Here are some considerations and key actions that you can take to strengthen your data culture.

- ✓ **Align your data culture goals and strategy**: Give serious consideration to the type of data culture that you want to cultivate. Ideally, it's more from a position of user empowerment than a position of command and control.
- ✓ **Understand your current state**: Talk to stakeholders in different business units to understand which analytics practices are currently working well and which practices

aren't working well for data-driven decision-making. Conduct a series of workshops to understand the current state and to formulate the desired future state.

- ✓ **Speak with stakeholders**: Talk to stakeholders in IT, BI, and the COE to understand which governance constraints need consideration. These conversations can present an opportunity to educate teams on topics like security and infrastructure. You can also use the opportunity to educate stakeholders on the features and capabilities included in Fabric.
- ✓ **Verify executive sponsorship**: Verify the level of executive sponsorship and support that you have in place to advance data culture goals.
- ✓ **Make purposeful decisions about your data strategy**: Decide what the ideal balance of business-led self-service, managed self-service, and enterprise data, analytics and BI use cases should be for the key business units in the organization (covered in the content ownership and management article). Also consider how the data strategy relates to the extent of published content for personal, team, departmental, and enterprise analytics and BI (described in the content delivery scope article). Define your high-level goals and priorities for this strategic planning. Determine how these decisions affect your tactical planning.
- ✓ **Create a tactical plan**: Begin creating a tactical plan for immediate, short-term, and long-term action items. Identify business groups and problems that represent "quick wins" and can make a visible difference.
- ✓ **Create goals and metrics**: Determine how you'll measure effectiveness for your data culture initiatives. Create key performance indicators (KPIs) or objectives and key results (OKRs) to validate the results of your efforts.

Questions to ask about data culture

Use questions like those found below to assess data culture.

- Is data regarded as a strategic asset in the organization?
- Is there a vision of a healthy data culture that originates from executive leadership and aligns with organizational objectives?
- Does the data culture guide creation of governance policies and guidelines?
- Are organizational data sources trusted by content creators and consumers?
- When justifying an opinion, decision, or choice, do people use data as evidence?
- Is knowledge about analytics and data use documented or is there a reliance on undocumented tribal knowledge?

- Are efforts to develop a data solution valued and appreciated by the user community?

Maturity levels of data culture

The following maturity levels will help you assess the current state of your data culture.

Expand table

Level	State of data culture
100: Initial	• Enterprise data teams can't keep up with the needs of the business. A significant backlog of requests exists. • Self-service data and BI initiatives are taking place with some success in various areas of the organization. These activities occur in a somewhat chaotic manner, with few formal processes and no strategic plan. • There's a lack of oversight and visibility into self-service BI activities. The successes or failures of data and BI solutions aren't well understood.
200: Repeatable	• Multiple teams have had measurable successes with self-service solutions. People in the organization are starting to pay attention. • Investments are being made to identify the ideal balance of enterprise and self-service data, analytics, and BI.
300: Defined	• Specific goals are established for advancing the data culture. These goals are implemented incrementally. • Learnings from what works in individual business units is shared. • Effective self-service practices are incrementally and purposely replicated throughout more areas of the organization.
400: Capable	• The data culture goals to employ informed decision-making are aligned with organizational objectives. They're actively supported by the executive sponsor, the COE, and they have a direct impact on adoption strategies. • A healthy and productive partnership exists between the executive sponsor, COE, business units, and IT. The teams are working towards shared goals.

Level	State of data culture
	• Individuals who take initiative in building valuable data solutions are recognized and rewarded.
500: Efficient	• The business value of data, analytics, and BI solutions is regularly evaluated and measured. KPIs or OKRs are used to track data culture goals and the results of these efforts. • Feedback loops are in place, and they encourage ongoing data culture improvements. • Continual improvement of organizational adoption, user adoption, and solution adoption is a top priority.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about the importance of an executive sponsor.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Executive sponsorship

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

When planning to advance the data culture and the state of organizational adoption for data and analytics, it's crucial to have executive support. An executive sponsor is imperative because analytics adoption is far more than just a technology project.

Although some successes can be achieved by a few determined individual contributors, the organization is in a significantly better position when a senior leader is engaged, supportive, informed, and available to assist with the following activities.

- Formulating a strategic vision, goals, and priorities for data, analytics, and business intelligence (BI).
- Providing top-down guidance and reinforcement for the data strategy by regularly promoting, motivating, and investing in strategic and tactical planning.
- Leading by example by actively using data and analytics in a way that's consistent with data culture and adoption goals.
- Allocating staffing and prioritizing resources.
- Approving funding (for example, Fabric licenses).
- Removing barriers to enable action.
- Communicating announcements that are of critical importance, to help them gain traction.
- Decision-making, particularly for strategic-level governance decisions.
- Dispute resolution (for escalated issues that can't be resolved by operational or tactical personnel).
- Supporting organizational change initiatives (for example, creating or expanding the Center of Excellence).

> **Important**
>
> The ideal executive sponsor has sufficient credibility, influence, and authority throughout the organization. They also have an invested stake in data efforts and

the data strategy. When the BI strategy is successful, the ideal executive sponsor also experiences success in their role.

Identifying an executive sponsor

There are multiple ways to identify an executive sponsor.

Top-down pattern

An executive sponsor might be selected by a more senior executive. For example, the Chief Executive Officer (CEO) could hire a Chief Data Officer (CDO) or Chief Analytics Officer (CAO) to explicitly advance the organization's data culture objectives or lead digital transformation efforts. The CDO or CAO then becomes the ideal candidate to serve as the executive sponsor for Fabric (or for data and analytics in general).

Here's another example: The CEO might empower an existing executive, such as the Chief Financial Officer (CFO), because they have a good track record leading data and analytics in their organization. As the new executive sponsor, the CFO could then lead efforts to replicate the finance team's success to other areas of the organization.

> **Note**
>
> Having an executive sponsor at the C-level is an excellent leading indicator. It indicates that the organization recognizes the importance of data as a strategic asset and is advancing its data culture in a positive direction.

Bottom-up pattern

Alternatively, a candidate for the executive sponsor role could emerge due to the success they've experienced with creating data solutions. For example, a business unit within the organization, such as Finance, has organically achieved great success with their use of data and analytics. Essentially, they've successfully formed their own data culture on a smaller scale. A junior-level leader who hasn't reached the executive level (such as a director) might then grow into the executive sponsor role by sharing successes with other business units across the organization.

The bottom-up approach is more likely to occur in smaller organizations. It might be because the return on investment and strategic imperative of a data culture (or digital transformation) isn't yet an urgent priority for C-level executives.

The success for a leader using the bottom-up pattern depends on being recognized by senior leadership.

With a bottom-up approach, the sponsor might be able to make some progress, but they won't have formal authority over other business units. Without clear authority, it's only a matter of time until challenges occur that are beyond their level of authority. For this reason, the top-down approach has a higher probability of success. However, initial successes with a bottom-up approach can convince leadership to increase their level of sponsorship, which might start a healthy competition across other business units in the adoption of data and BI.

Considerations and key actions

Checklist - Here's a list of considerations and key actions you can take to establish or strengthen executive support for analytics.

- ✓ **Identify an executive sponsor with broad authority**: Find someone in a sufficient position of influence and authority (across organizational boundaries) who understands the value and impact of BI. It is important that the individual has a vested interest in the success of analytics in the organization.
- ✓ **Involve your executive sponsor**: Consistently involve your executive sponsor in all strategic-level governance decisions involving data management, BI, and analytics. Also involve your sponsor in all governance data culture initiatives to ensure alignment and consensus on goals and priorities.
- ✓ **Establish responsibilities and expectation**: Formalize the arrangement with documented responsibilities for the executive sponsor role. Ensure that there's no uncertainty about expectations and time commitments.
- ✓ **Identify a backup for the sponsor**: Consider naming a backup executive sponsor. The backup can attend meetings in the sponsor's absence and make time-sensitive decisions when necessary.
- ✓ **Identify business advocates**: Find influential advocates in each business unit. Determine how their cooperation and involvement can help you to accomplish your objectives. Consider involving advocates from various levels in the organization chart.

Questions to ask

Use questions like those found below to assess data literacy.

- Has an executive sponsor of Fabric or other analytical tools been identified?
- If so, who is the executive sponsor?
- If not, is there an informal executive sponsor? Who is the closest to this role? Can you define the business impact of having no executive sponsor?
- To what extent is the strategic importance of Fabric and analytics understood and endorsed by executives?
- Are executives using Fabric and the results of data and BI initiatives? What's the sentiment among executives for the effectiveness of data solutions?
- Is the executive sponsor leading by example in the effective use of data and BI tools?
- Does the executive sponsor provide the appropriate resources for data initiatives?
- Is the executive sponsor involved in dispute resolution and change management?
- Does the executive sponsor engage with the user community?
- Does the executive sponsor have sufficient credibility and healthy relationships across organizational boundaries (particularly the business and IT)?

Maturity levels

The following maturity levels will help you assess your current state of executive support.

Expand table

Level	State of executive support
100: Initial	• There might be awareness from at least one executive about the strategic importance of how analytics can advance the organization's data culture goals. However, neither a sponsor nor an executive-level decision-maker is identified.
200:	• Informal executive support exists for analytics through informal channels and

Level	State of executive support
Repeatable	relationships.
300: Defined	• An executive sponsor is identified. Expectations are clear for the role.
400: Capable	• An executive sponsor is well established with someone with sufficient authority across organizational boundaries. • A healthy and productive partnership exists between the executive sponsor, COE, business units, and IT. The teams are working towards shared data culture goals.
500: Efficient	• The executive sponsor is highly engaged. They're a key driver for advancing the organization's data culture vision. • The executive sponsor is involved with ongoing organizational adoption improvements. KPIs (key performance indicators) or OKRs (objectives and key results) are used to track data culture goals and the results of data, analytics, and BI efforts.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about the importance of business alignment with organizational goals.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Business alignment

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

Business intelligence (BI) activities and solutions have the best potential to deliver value when they're well aligned to organizational business goals. In general, effective business alignment helps to improve adoption. With effective business alignment, the data culture and data strategy enable business users to achieve their business objectives.

You can achieve effective business alignment with data activities and solutions by having:

- An understanding of the strategic importance of data and analytics in achieving measurable progress toward business goals.
- A shared awareness of the business strategy and key business objectives among content owners, creators, consumers, and administrators. A common understanding should be integral to the data culture and decision-making across the organization.
- A clear and unified understanding of the business data needs, and how meeting these needs helps content creators and content consumers achieve their objectives.
- A governance strategy that effectively balances user enablement with risk mitigation.
- An engaged executive sponsor who provides top-down guidance to regularly promote, motivate, and support the data strategy and related activities and solutions.
- Productive and solution-oriented discussions between business teams and technical teams that address business data needs and problems.
- Effective and flexible requirements gathering processes to design and plan solutions.
- Structured and consistent processes to validate, deploy, and support solutions.
- Structured and sustainable processes to regularly update existing solutions so that they remain relevant and valuable, despite changes in technology or business objectives.

Effective business alignment brings significant benefits to an organization. Here are some benefits of effective business alignment.

- Improved adoption, because content consumers are more likely to use solutions that enable them to achieve their objectives.
- Increased business return on investment (ROI) for analytics initiatives and solutions, because these initiatives and solutions will be more likely to directly advance progress toward business goals.
- Less effort and fewer resources spent on change management and changing business requirements, due to an improved understanding of business data needs.

Achieve business alignment

There are multiple ways to achieve business alignment of data activities and initiatives.

Communication alignment

Effective and consistent communication is critical to aligning processes. Consider the following actions and activities when you want to improve communication for successful business alignment.

- Make and follow a plan for central teams and the user community to follow.
- Plan regular alignment meetings between different teams and groups. For example, central teams can plan regular planning and priority alignments with business units. Another example is when central teams schedule regular meetings to mentor and enable self-service users.
- Set up a centralized portal to consolidate communication and documentation for user communities. For strategic solutions and initiatives, consider using a communication hub.
- Limit complex business and technical terminology in cross-functional communications.
- Strive for concise communication and documentation that's formatted and well organized. That way, people can easily find the information that they need.
- Consider maintaining a visible roadmap that shows the planned solutions and activities relevant to the user community in the next quarter.
- Be transparent when communicating policies, decisions, and changes.
- Create a process for people to provide feedback, and review that feedback regularly as part of regular planning activities.

> **Important**

To achieve effective business alignment, you should make it a priority to identify and dismantle any communication barriers between business teams and technical teams.

Strategic alignment

Your business strategy should be well aligned with your data and BI strategy. To incrementally achieve this alignment, we recommend that you commit to following structured, iterative planning processes.

- **Strategic planning**: Define data, analytics, and BI goals and priorities based on the business strategy and current state of adoption and implementation. Typically, strategic planning occurs every 12-18 months to iteratively define high-level desired outcomes. You should synchronize strategic planning with key business planning processes.
- **Tactical planning**: Define objectives, action plans, and a backlog of solutions that help you to achieve your data and BI goals. Typically, tactical planning occurs quarterly to iteratively re-evaluate and align the data strategy and activities to the business strategy. This alignment is informed by business feedback and changes to business objectives or technology. You should synchronize tactical planning with key project planning processes.
- **Solution planning**: Design, develop, test, and deploy solutions that support content creators and consumers in achieving their business objectives. Both centralized content creators and self-service content creators conduct solution planning to ensure that the solutions they create are well aligned with business objectives. You should synchronize solution planning with key adoption and governance planning processes.

> **Important**
>
> Effective business alignment is a key prerequisite for a successful **data strategy**.

Governance and compliance alignment

A key aspect of effective business alignment is balancing user enablement and risk mitigation. This balance is an important aspect of your governance strategy, together with other activities related to compliance, security and privacy, that can include:

- Transparently document and justify compliance criteria, key governance decisions, and policies so that content creators and consumers know what's expected of

them.

- Regularly audit and assess activities to identify risk areas or strong deviations from the desired behaviors.
- Provide mechanisms for content owners, content creators, and content consumers to request clarification or provide feedback about existing policies.

> **⊗ Caution**
>
> A governance strategy that's poorly aligned with business objectives can result in more conflicts and compliance risk, because users will often pursue workarounds to complete their tasks.

Executive alignment

Executive leadership plays a key role in defining the business strategy and business goals. To this end, executive engagement is an important part of achieving top-down business alignment.

To achieve executive alignment, consider the following key considerations and activities.

- Work with your executive sponsor to organize short, quarterly executive feedback sessions about the use of data in the organization. Use this feedback to identify changes in business objectives, re-assess the data strategy, and inform future actions to improve business alignment.
- Schedule regular alignment meetings with the executive sponsor to promptly identify any potential changes in the business strategy or data needs.
- Deliver monthly executive summaries that highlight relevant information, including:
  - Key performance indicators (KPIs) that measure progress toward data, analytics, and BI goals.
  - Fabric adoption and implementation milestones.
  - Technology changes that might impact organizational business goals.

> **ⓘ Important**
>
> Don't underestimate the importance of the role your **executive sponsor** has in achieving and maintaining effective business alignment.

Maintain business alignment

Business alignment is a continual process. To maintain business alignment, consider the following factors.

- **Assign a responsible team**: A working team reviews feedback and organizes re-alignment sessions. This team is responsible for the alignment of planning and priorities between the business and data strategy.
- **Create and support a feedback process**: Your user community requires the means to provide feedback. Examples of feedback can include requests to change existing solutions, or to create new solutions and initiatives. This feedback is essential for bottom-up business user alignment, and it drives iterative and continuous improvement cycles.
- **Measure the success of business alignment**: Consider using surveys, sentiment analysis, and usage metrics to assess the success of business alignment. When combined with other concise feedback mechanisms, this can provide valuable input to help define future actions and activities to improve business alignment and Fabric adoption.
- **Schedule regular re-alignment sessions**: Ensure that data strategic planning and tactical planning occur alongside relevant business strategy planning (when business leadership review business goals and objectives).

> **Note**
>
> Because business objectives continually evolve, you should understand that solutions and initiatives will change over time. Don't assume that requirements for data and BI projects are rigid and can't be altered. If you struggle with changing requirements, it might be an indication that your requirements-gathering process is ineffective or inflexible, or that your development workflows don't sufficiently incorporate regular feedback.

> **Important**
>
> To effectively maintain business alignment, it's essential that user feedback be promptly and directly addressed. Regularly review and analyze feedback, and consider how you can integrate it into iterative strategic planning, tactical planning, and solution planning processes.

Questions to ask

Use questions like those found below to assess business alignment.

- Can people articulate the goals of the organization and the business objectives of their team?
- To what extent do descriptions of organizational goals align across the organization? How do they align between the business user community and leadership community? How do they align between business teams and technical teams?
- Does executive leadership understand the strategic importance of data in achieving business objectives? Does the user community understand the strategic importance of data in helping them succeed in their jobs?
- Are changes in the business strategy reflected promptly in changes to the data strategy?
- Are changes in business user data needs addressed promptly in data and BI solutions?
- To what extent do data policies support or conflict with existing business processes and the way that users work?
- Do solution requirements focus more on technical features than addressing business questions? Is there a structured requirements gathering process? Do content owners and creators interact effectively with stakeholders and content consumers during requirements gathering?
- How are decisions about data or BI investments made? Who makes these decisions?
- How well do people trust existing data and BI solutions? Is there a single version of truth, or are there regular debates about who has the correct version?
- How are data and BI initiatives and strategy communicated across the organization?

Maturity levels

A business alignment assessment evaluates integration between the business strategy and data strategy. Specifically, this assessment focuses on whether or not data and BI initiatives and solutions support business users to achieve business strategic objectives.

The following maturity levels will help you assess your current state of business alignment.

Expand table

Level	ate of data and business alignment**
100: Initial	• Business and data strategies lack formal alignment, which leads to reactive implementation and misalignment between data teams and business users. • Misalignment in priorities and planning hinders productive discussions and effectiveness. • Executive leadership doesn't recognize data as a strategic asset.
200: Repeatable	• There are efforts to align data and BI initiatives with specific data needs without a consistent approach or understanding of their success. • Alignment discussions focus on immediate or urgent needs and focus on features, solutions, tools or data, rather than strategic alignment. • People have a limited understanding of the strategic importance of data in achieving business objectives.
300: Defined	• Data and BI initiatives are prioritized based on their alignment with strategic business objectives. However, alignment is siloed and typically focuses on local needs. • Strategic initiatives and changes have a clear, structured involvement of both the business and data strategic decision makers. Business teams and technical teams can have productive discussions to meet business and governance needs.
400: Capable	• There's a consistent, organization-wide view of how data initiatives and solutions support business objectives. • Regular and iterative strategic alignments occur between the business and technical teams. Changes to the business strategy result in clear actions that are reflected by changes to the data strategy to better support business needs. • Business and technical teams have healthy, productive relationships.
500: Efficient	• The data strategy and the business strategy are fully integrated. Continuous improvement processes drive consistent alignment, and they are themselves data driven.

Level	ate of data and business alignment**
	• Business and technical teams have healthy, productive relationships.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn more about content ownership and management, and its effect on business-led self-service BI, managed self-service BI, and enterprise BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Content ownership and management

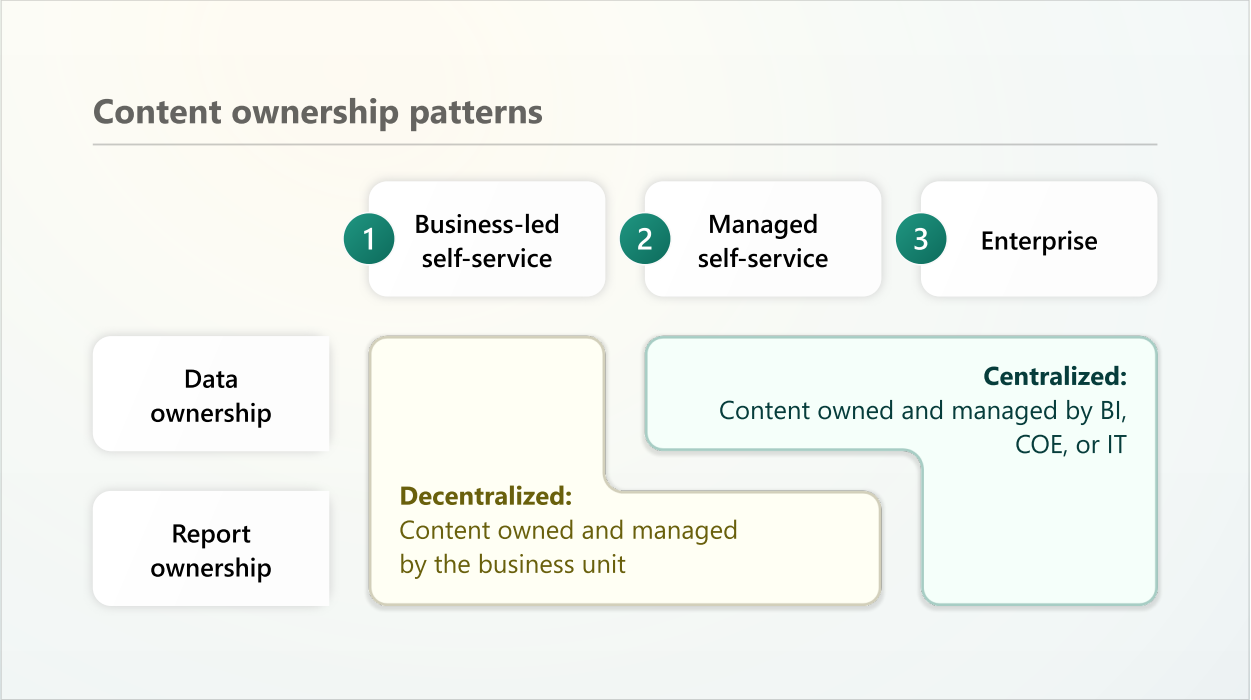
Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

> **Note**
>
> The Power BI implementation planning usage scenarios explore many concepts discussed in this article, focusing on the Power BI workload in Microsoft Fabric. The usage scenario articles include detailed diagrams that you might find helpful to support your planning and decision making.

There are three primary strategies for how data, analytics, and business intelligence (BI) content is owned and managed: business-led self-service, managed self-service, and enterprise. For the purposes of this series of articles, the term *content* refers to any type of data item (like a notebook, semantic model, report, or dashboard).

The organization's data culture is the driver for why, how, and by whom each of these three content ownership strategies is implemented.



The areas in the above diagram include:

Expand table

Area	Description
1	**Business-led self-service**: All content is owned and managed by the creators and subject matter experts within a business unit. This ownership strategy is also known as a *decentralized* or *bottom-up* strategy.
2	**Managed self-service**: The data is owned and managed by a centralized team, whereas business users take responsibility for reports and dashboards. This ownership strategy is also known as *discipline at the core and flexibility at the edge*.
3	**Enterprise**: All content is owned and managed by a centralized team such as IT, enterprise BI, or the Center of Excellence (COE).

It's unlikely that an organization operates exclusively with one content ownership and management strategy. Depending on your data culture, one strategy might be far more dominant than the others. The choice of strategy could differ from solution to solution, or from team to team. In fact, a single team can actively use multiple strategies if it's both a consumer of enterprise content and a producer of its own self-service content. The strategy to pursue depends on factors such as:

- Requirements for a solution (such as a collection of reports, a Power BI app, or a lakehouse).
- User skills.
- Ongoing commitment for training and skills growth.
- Flexibility required.
- Complexity level.
- Priorities and leadership commitment level.

The organization's data culture—particularly its position on data democratization—has considerable influence on the extent of which of the three content ownership strategies are used. While there are common patterns for success, there's no one-size-fits-all approach. Each organization's governance model and approach to content ownership and management should reflect the differences in data sources, applications, and business context.

How content is owned and managed has a significant effect on governance, the extent of mentoring and user enablement, needs for user support, and the COE operating model.

As discussed in the governance article, the level of governance and oversight depends on:

- Who owns and manages the content.
- The scope of content delivery.
- The data subject area and sensitivity level.
- The importance of the data, and whether it's used for critical decision making.

In general:

- Business-led self-service content is subject to the least stringent governance and oversight controls. It often includes personal BI and team BI solutions.
- Managed self-service content is subject to moderately stringent governance and oversight controls. It frequently includes team BI and departmental BI solutions.
- Enterprise solutions are subject to more rigorous governance controls and oversight.

As stated in the adoption maturity levels article, organizational adoption measures the state of data management processes and governance. The choices made for content ownership and management significantly affect how organizational adoption is achieved.

Ownership and stewardship

There are many roles related to data management. Roles can be defined in many ways and can be easily misunderstood. The following table presents possible ways you might conceptually define these roles:

Expand table

Role	Description
Data steward	Responsible for defining and/or managing acceptable data quality levels as well as master data management (MDM).
Subject matter expert (SME)	Responsible for defining what the data means, what it's used for, who might access it, and how the data is presented to others. Collaborates with domain owner as needed and supports colleagues in their use of data.
Technical owner	Responsible for creating, maintaining, publishing, and securing access to data and reporting items.
Domain owner	Higher-level decision-maker who collaborates with governance teams on data management policies, processes, and requirements. Decision-maker for defining appropriate and inappropriate uses of the data. Participates on the data governance board, as described in the governance article.

Assigning ownership for a data domain tends to be more straightforward when managing transactional source systems. In analytics and BI solutions, data is integrated from multiple domain areas, then transformed and enriched. For downstream analytical solutions, the topic of ownership becomes more complex.

> **Note**
>
> Be clear about who is responsible for managing data items. It's crucial to ensure a good experience for content consumers. Specifically, clarity on ownership is helpful for:
>
> - Who to contact with questions.
> - Feedback.
> - Enhancement requests.
> - Support requests.
>
> In the Fabric portal, content owners can set the **contact list property** for many types of items. The contact list is also used in security workflows. For example, when a user is sent a URL to open a **Power BI app** but they don't have permission, they will be presented with an option to make a request for access.

Guidelines for being successful with ownership:

- Define how ownership and stewardship terminology is used in your organization, including expectations for these roles.
- Set contacts for each workspace and for individual items to communicate ownership and/or support responsibilities.
- Specify between two and four workspace administrators and conduct an audit of workspace admins regularly (perhaps twice a year). Workspace admins might be directly responsible for managing workspace content, or it could be that those tasks are assigned to colleagues who do the hands-on work. In all cases, workspace admins should be able to easily contact owners of specific content.
- Include consistent branding on reports to indicate who produced the content and who to contact for help. A small image or text label located in the report footer is valuable, especially when the report is exported from the Fabric portal. A standard template file can encourage and simplify the consistent use of branding.
- Make use of best practices reviews and co-development projects with the COE.

The remainder of this article covers considerations related to the three content ownership and management strategies.

Business-led self-service

With a business-led self-service approach to data and BI, all content is owned and managed by creators and subject matter experts. Because responsibility is retained within a business unit, this strategy is often described as the *bottom-up*, or *decentralized*, approach. Business-led self-service is often a good strategy for personal BI and team BI solutions.

> **Important**
>
> The concept of business-led self-service isn't the same as shadow IT. In both scenarios, data and BI content is created, owned, and managed by business users. However, shadow IT implies that the business unit is circumventing IT and so the solution is not sanctioned. With business-led self-service BI solutions, the business unit has full authority to create and manage content. Resources and support from the COE are available to self-service content creators. It's also expected that the business unit will comply with all established data governance guidelines and policies.

Business-led self-service is most suitable when:

- Decentralized data management aligns with the organization's data culture, and the organization is prepared to support these efforts.
- Data exploration and freedom to innovate is a high priority.
- The business unit wants to have the most involvement and retain the highest level of control.
- The business unit has skilled users capable of—and fully committed to—supporting solutions through the entire lifecycle. It covers all types of items, including the data (such as a lakehouse, data warehouse, data pipeline, dataflow, or semantic model), the visuals (such as reports and dashboards), and Power BI apps.
- The flexibility to respond to changing business conditions and react quickly outweighs the need for stricter governance and oversight.

Here are some guidelines to help become successful with business-led self-service data and BI.

- Teach your creators to use the same techniques that IT would use, like shared semantic models and dataflows. Make use of a well-organized OneLake. Centralize data to reduce maintenance, improve consistency, and reduce risk.

- Focus on providing mentoring, training, resources, and documentation (described in the Mentoring and user enablement article). The importance of these efforts can't be overstated. Be prepared for skill levels of self-service content creators to vary significantly. It's also common for a solution to deliver excellent business value yet be built in such a way that it won't scale or perform well over time (as historic data volumes increase). Having the COE available to help when these situations arise is very valuable.
- Provide guidance on the best way to use endorsements. The promoted endorsement is for content produced by self-service creators. Consider reserving use of the certified endorsement for enterprise BI content and managed self-service BI content (described next).
- Analyze the activity log to discover situations where the COE could proactively contact self-service owners to offer helpful information. It's especially useful when a suboptimal usage pattern is detected. For example, log activity could reveal overuse of individual item sharing when Power BI app audiences or workspace roles might be a better choice. The data from the activity log allows the COE to offer support and advice to the business units. In turn, this information can help increase the quality of solutions, while allowing the business to retain full ownership and control of their content. For more information, see Auditing and monitoring.

Managed self-service

Managed self-service BI is a blended approach to data and BI. The data is owned and managed by a centralized team (such as IT, enterprise BI, or the COE), while responsibility for reports and dashboards belongs to creators and subject matter experts within the business units. Managed self-service BI is frequently a good strategy for team BI and departmental BI solutions.

This approach is often called_discipline at the core and flexibility at the edge_. It's because the data architecture is maintained by a single team with an appropriate level of discipline and rigor. Business units have the flexibility to create reports and dashboards based on centralized data. This approach allows report creators to be far more efficient because they can remain focused on delivering value from their data analysis and visuals.

Managed self-service BI is most suitable when:

- Centralized data management aligns with the organization's data culture.
- The organization has a team of BI experts who manage the data architecture.

- There's value in the reuse of data by many self-service report creators across organizational boundaries.
- Self-service report creators need to produce analytical content at a pace faster than the centralized team can accommodate.
- Different users are responsible for handling data preparation, data modeling, and report creation.

Here are some guidelines to help you become successful with managed self-service BI.

- Teach users to separate model and report development. They can use live connections to create reports based on existing semantic models. When the semantic model is decoupled from the report, it promotes data reuse by many reports and many authors. It also facilitates the separation of duties.
- Use dataflows to centralize data preparation logic and to share commonly used data tables—like date, customer, product, or sales—with many semantic model creators. Refine the dataflow as much as possible, using friendly column names and correct data types to reduce the downstream effort required by semantic model authors, who consume the dataflow as a source. Dataflows are an effective way to reduce the time involved with data preparation and improve data consistency across semantic models. The use of dataflows also reduces the number of data refreshes on source systems and allows fewer users who require direct access to source systems.
- When self-service creators need to augment an existing semantic model with departmental data, educate them to create composite models. This feature allows for an ideal balance of self-service enablement while taking advantage of the investment in data assets that are centrally managed.
- Use the certified endorsement for semantic models and dataflows to help content creators identify trustworthy sources of data.
- Include consistent branding on all reports to indicate who produced the content and who to contact for help. Branding is particularly helpful to distinguish content that is produced by self-service creators. A small image or text label in the report footer is valuable when the report is exported from the Fabric portal.
- Consider implementing separate workspaces for storing data and reports. This approach allows for better clarity on who is responsible for content. It also allows for more restrictive workspace roles assignments. That way, report creators can only publish content to their reporting workspace; and, read and build semantic model permissions allow creators to create new reports with row-level security (RLS) in effect, when applicable. For more information, see Workspace-level planning. For more information about RLS, see Content creator security planning.
- Use the Power BI REST APIs to compile an inventory of Power BI items. Analyze the ratio of semantic models to reports to evaluate the extent of semantic model

reuse.

Enterprise

Enterprise is a centralized approach to delivering data and BI solutions in which all solution content is owned and managed by a centralized team. This team is usually IT, enterprise BI, or the COE.

Enterprise is the most suitable when:

- Centralizing content management with a single team aligns with the organization's data culture.
- The organization has data and BI expertise to manage all items end-to-end.
- The content needs of consumers are well-defined, and there's little need to customize or explore data beyond the reporting solution that's delivered.
- Content ownership and direct access to data needs to be limited to a small number of experts and owners.
- The data is highly sensitive or subject to regulatory requirements.

Here are some guidelines to help you become successful with enterprise data and BI.

- Implement a rigorous process for use of the certified endorsement for content. Not all enterprise content needs to be certified, but much of it probably should be. Certified content should indicate that data quality has been validated. Certified content should also follow change management rules, have formal support, and be fully documented. Because certified content has passed rigorous standards, the expectations for trustworthiness are higher.
- Include consistent branding on enterprise BI reports to indicate who produced the content, and who to contact for help. A small image or text label in the report footer is valuable when the report is exported by a user.
- If you use specific report branding to indicate enterprise BI content, be careful with the *save a copy* functionality that would allow a user to download a copy of a report and personalize it. Although this functionality is an excellent way to bridge enterprise BI with managed self-service BI, it dilutes the value of the branding. A more seamless solution is to provide a separate Power BI Desktop template file for self-service authors. The template defines a starting point for report creation with a live connection to an existing semantic model, and it doesn't include branding. The template file can be shared as a link within a Power BI app, or from the community portal.

Ownership transfers

Occasionally, the ownership of a particular solution might need to be transferred to another team. An ownership transfer from a business unit to a centralized team can happen when:

- A business-led solution is used by a significant number of users, or it now supports critical business decisions. In these cases, the solution should be managed by a team with processes in place to implement higher levels of governance and support.
- A business-led solution is a candidate to be used far more broadly throughout the organization, so it needs to be managed by a team who can set security and deploy content widely throughout the organization.
- A business unit no longer has the expertise, budget, or time available to continue managing the content, but the business need for the content remains.
- The size or complexity of a solution has grown to a point where a different data architecture or redesign is required.
- A proof of concept is ready to be operationalized.

The COE should have well-documented procedures for identifying when a solution is a candidate for ownership transfer. It's very helpful if help desk personnel know what to look for as well. Having a customary pattern for self-service creators to build and grow a solution, and hand it off in certain circumstances, is an indicator of a productive and healthy data culture. A simple ownership transfer could be addressed during COE office hours; a more complex transfer could warrant a small project managed by the COE.

> **Note**
>
> There's potential that the new owner will need to do some refactoring and data validations before they're willing to take full ownership. Refactoring is most likely to occur with the less visible aspects of data preparation, data modeling, and calculations. If there are any manual steps or flat file sources, now is an ideal time to apply those enhancements. The branding of reports and dashboards might also need to change (for example, if there's a footer indicating report contact or a text label indicating that the content is certified).

It's also possible for a centralized team to transfer ownership to a business unit. It could happen when:

- The team with domain knowledge is better equipped to own and manage the content going forward.
- The centralized team has created the solution for a business unit that doesn't have the skills to create it from scratch, but it can maintain and extend the solution

going forward.

> **Tip**
>
> Don't forget to recognize and reward the work of the original creator, particularly if ownership transfers are a common occurrence.

Considerations and key actions

Checklist - Here's a list of considerations and key actions you can take to strengthen your approach to content ownership and management.

- ✓ **Gain a full understanding of what's currently happening**: Ensure you deeply understand how content ownership and management is happening throughout the organization. Recognize that there likely won't be a one-size-fits-all approach to apply uniformly across the entire organization. Review the implementation planning usage scenarios to understand how Power BI and Fabric can be used in diverse ways.
- ✓ **Conduct discussions**: Determine what is currently working well, what isn't working well, and what the desired balance is between the three ownership strategies. If necessary, schedule discussions with specific people on various teams. Develop a plan for moving from the current state to the desired state.
- ✓ **Perform an assessment**: If your enterprise data team currently has challenges related to scheduling and priorities, do an assessment to determine if a managed self-service strategy can be put in place to empower more content creators throughout the organization. Managed self-service data and BI can be extremely effective on a global scale.
- ✓ **Clarify terminology**: Clarify terms used in your organization for owner, data steward, and subject matter expert.
- ✓ **Assign clear roles and responsibilities**: Make sure roles and responsibilities for owners, stewards, and subject matter experts are documented and well understood by everyone involved. Include backup personnel.
- ✓ **Ensure community involvement**: Ensure that all your content owners—from both the business and IT—are part of your community of practice.
- ✓ **Create user guidance for owners and contacts in Fabric**: Determine how you will use the contacts feature in Fabric. Communicate with content creators about how it

should be used, and why it's important.

- ✓ **Create a process for handling ownership transfers**: If ownership transfers occur regularly, create a process for how it will work.
- ✓ **Support your advanced content creators**: Determine your strategy for using external tools for advanced authoring capabilities and increased productivity.

Questions to ask

Use questions like those found below to assess content ownership and management.

- Do central teams that are responsible for Fabric have a clear understanding of who owns what BI content? Is there a distinction between report and data items, or different item types (like Power BI semantic models, data science notebooks, or lakehouses)?
- Which usage scenarios are in place, such as personal BI, team BI, departmental BI, or enterprise BI? How prevalent are they in the organization, and how do they differ between key business units?
- What activities do business analytical teams perform (for example, data integration, data modeling, or reporting)?
- What kinds of roles in the organizations are expected to create and own content? Is it limited to central teams, analysts, or also functional roles, like sales?
- Where does the organization sit on the spectrum of business-led self-service, managed self-service, or enterprise? Does it differ between key business units?
- Do strategic data and BI solutions have ownership roles and stewardship roles that are clearly defined? Which are missing?
- Are content creators and owners also responsible for supporting and updating content once it's released? How effective is the ownership of content support and updates?
- Is a clear process in place to transfer ownership of solutions (where necessary)? An example is when an external consultant creates or updates a solution.
- Do data sources have data stewards or subject matter experts (SMEs) who serve as a special point of contact?
- If your organization is already using Fabric or Power BI, does the current workspace setup comply with the content ownership and delivery strategies that are in place?

Maturity levels

The following maturity levels will help you assess the current state of your content ownership and management.

Expand table

Level	State of content ownership and management
100: Initial	• Self-service content creators own and manage content in an uncontrolled way, without a specific strategy. • A high ratio of semantic models to reports exists. When many semantic models exist only support one report, it indicates opportunities to improve data reusability, improve trustworthiness, reduce maintenance, and reduce the number of duplicate semantic models. • Discrepancies between different reports are common, causing distrust of content produced by others.
200: Repeatable	• A plan is in place for which content ownership and management strategy to use and in which circumstances. • Initial steps are taken to improve the consistency and trustworthiness levels for self-service efforts. • Guidance for the user community is available that includes expectations for self-service versus enterprise content. • Roles and responsibilities are clear and well understood by everyone involved.
300: Defined	• Managed self-service is a priority and an area of investment to further advance the data culture. The priority is to allow report creators the flexibility they need while using well-managed, secure, and trustworthy data sources. • Report branding is consistently used to indicate who produced the content. • A mentoring program exists to educate self-service content creators on how to apply best practices and make good decisions.
400: Capable	• Criteria are defined to align governance requirements for self-service versus enterprise content.

Level	State of content ownership and management
	• There's a plan in place for how to request and handle ownership transfers. • Managed self-service—and techniques for the reuse of data—are commonly used and well-understood.
500: Efficient	• Proactive steps to communicate with users occur when any concerning activities are detected in the activity log. Education and information are provided to make gradual improvements or reduce risk. • Third-party tools are used by highly proficient content creators to improve productivity and efficiency.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about the scope of content delivery.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Content delivery scope

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

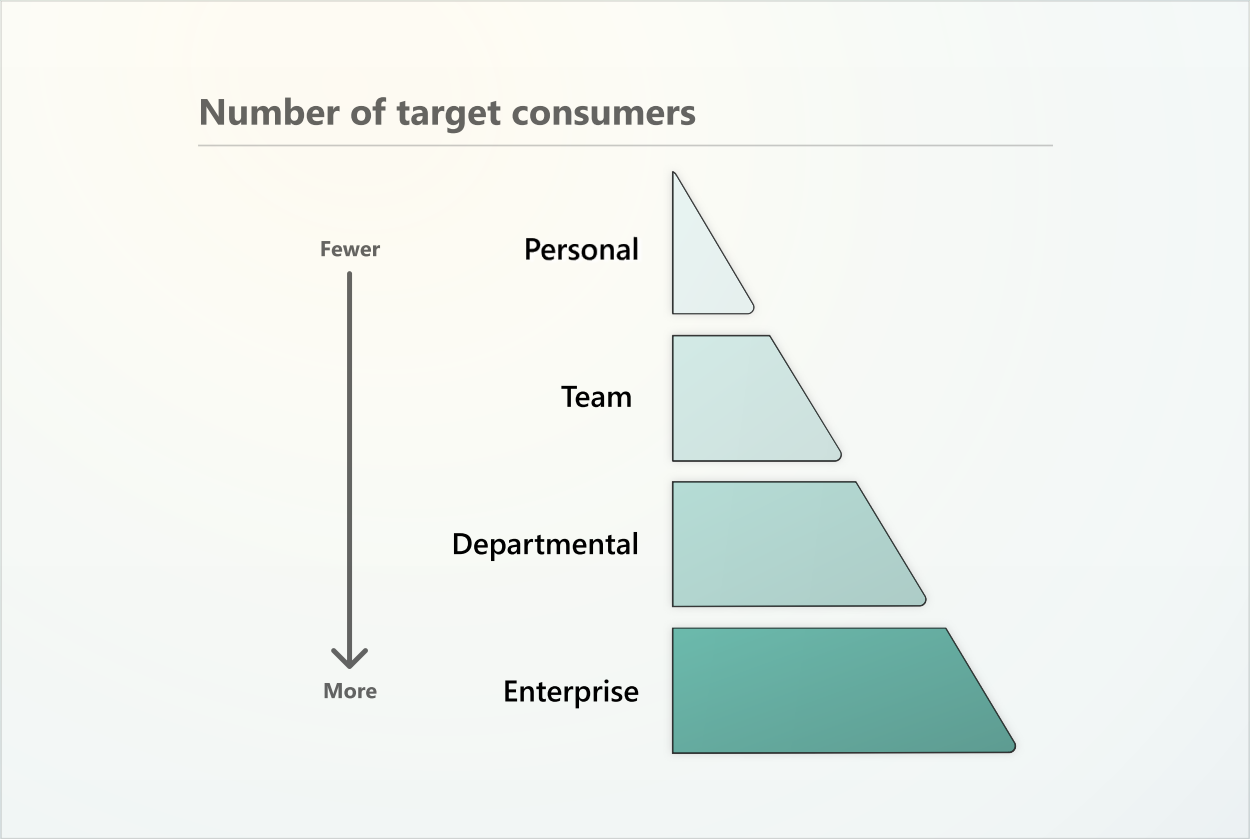
The four delivery scopes described in this article include personal, team, departmental, and enterprise. To be clear, focusing on the scope of a delivered data and business intelligence (BI) solution does refer to the number of people who might view the solution, though the impact is much more than that. The scope strongly influences best practices for not only content distribution, but also content management, security, and information protection. The scope has a direct correlation to the level of governance (such as requirements for change management, support, or documentation), the extent of mentoring and user enablement, and needs for user support. It also influences user licensing decisions.

The related content ownership and management article makes similar points. Whereas the focus of that article was on the content creator, the focus of this article is on the target content usage. Both inter-related aspects need to be considered to arrive at governance decisions and the Center of Excellence (COE) operating model.

> **Important**
>
> Not all data and solutions are equal. Be prepared to apply different levels of data management and governance to different teams and various types of content. Standardized rules are easier to maintain. However, flexibility or customization is often necessary to apply the appropriate level of oversight for particular circumstances. Your **executive sponsor** can prove invaluable by reaching consensus across stakeholder groups when difficult situations arise.

Scope of content delivery

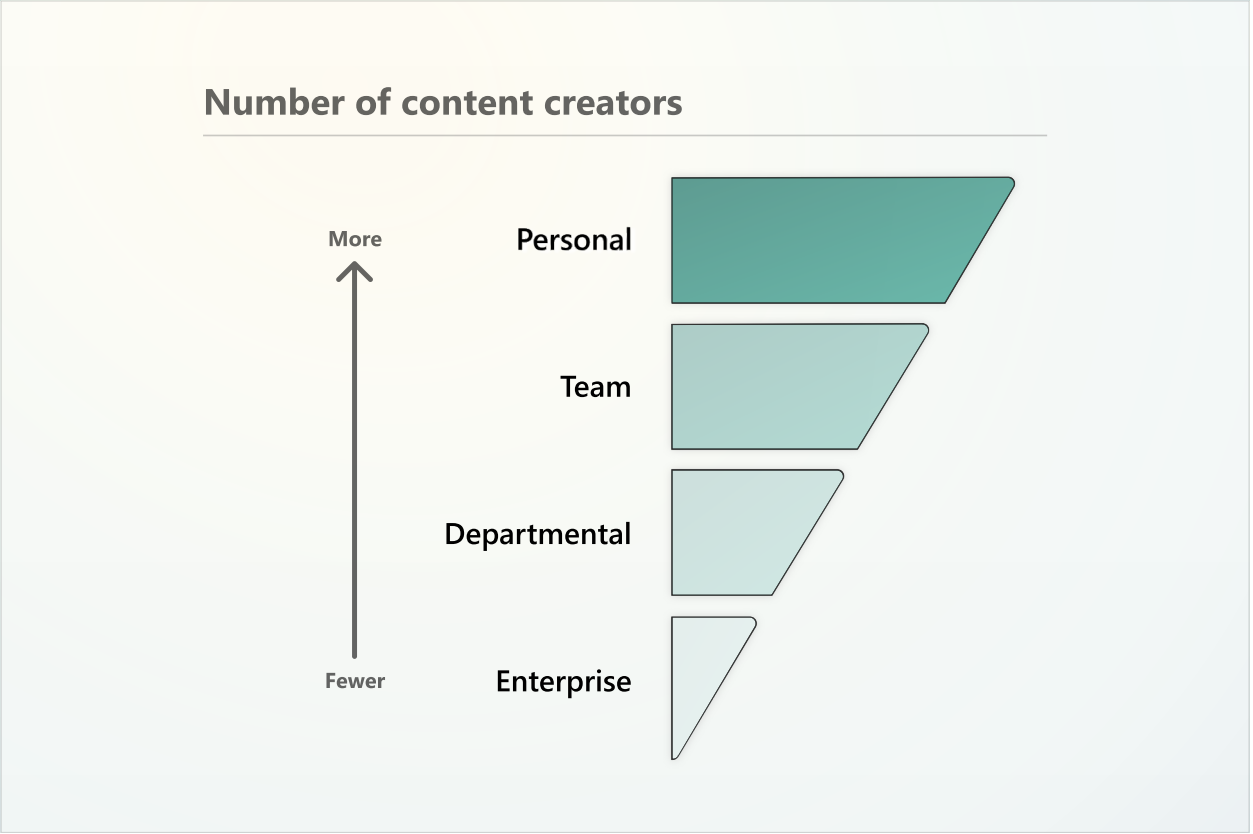
The following diagram focuses on the number of *target consumers* who will consume the content.



The four scopes of content delivery shown in the above diagram include:

- **Personal**: Personal solutions are, as the name implies, intended for use by the creator. Sharing content with others isn't an objective. Therefore, a personal data and BI solution has the fewest number of target consumers.
- **Team**: Collaborates and shares content with a relatively small number of colleagues who work closely together.
- **Departmental**: Delivers content to a large number of consumers, who can belong to a department or business unit.
- **Enterprise**: Delivers content broadly across organizational boundaries to the largest number of target consumers. Enterprise content is most often managed by a centralized team and is subject to additional governance requirements.

Contrast the above four scopes of content delivery with the following diagram, which has an inverse relationship with respect to the number of *content creators*.



The four scopes of content creators shown in the above diagram include:

- **Personal**: Represents the largest number of creators because the data culture encourages any user to work with data using business-led self-service data and BI methods. Although managed self-service BI methods can be used, it's less common with personal data and BI efforts.
- **Team**: Colleagues within a team collaborate and share with each other by using business-led self-service patterns. It has the next largest number of creators in the organization. Managed self-service patterns could also begin to emerge as skill levels advance.
- **Departmental**: Involves a smaller population of creators. They're likely to be considered power users who are using sophisticated tools to create sophisticated solutions. Managed self-service practices are very common and highly encouraged.
- **Enterprise**: Involves the smallest number of content creators because it typically includes only professional data and BI developers who work in the BI team, the COE, or in IT.

The content ownership and management article introduced the concepts of business-led self-service, managed self-service, and enterprise. The most common alignment between ownership and delivery scope is:

- **Business-led self-service ownership**: Commonly deployed as personal and team solutions.
- **Managed self-service ownership**: Can be deployed as personal, team, or departmental solutions.

- **Enterprise ownership**: Typically deployed as enterprise-scoped solutions.

Some organizations also equate self-service content with community-based support. It's the case when self-service content creators and owners are responsible for supporting the content they publish. The user support article describes multiple informal and formal levels for support.

> **Note**
>
> The term *sharing* can be interpreted two ways: It's often used in a general way related to sharing content with colleagues, which could be implemented multiple ways. It can also reference a specific feature in Fabric, which is a specific implementation where a user or group is granted access to a single item. In this article, the term *sharing* is meant in a general way to describe sharing content with colleagues. When the per-item permissions are intended, this article will make a clear reference to that feature. For more information, see Report consumer security planning.

Personal

The Personal delivery scope is about enabling an individual to gain analytical value. It's also about allowing them to more efficiently perform business tasks through the effective personal use of data, information, and analytics. It could apply to any type of information worker in the organization, not just data analysts and developers.

Sharing content with others isn't the objective. Personal content can reside in Power BI Desktop or in a personal workspace in the Fabric portal.

Here are the characteristics of creating content for a personal delivery scope.

- The creator's primary intention is data exploration and analysis, rather than report delivery.
- The content is intended to be analyzed and consumed by one person: the creator.
- The content might be an exploratory proof of concept that may, or may not, evolve into a project.

Here are a few guidelines to help you become successful with content developed for personal use.

- Consider personal data and BI solutions to be like an *analytical sandbox* that has little formal governance and oversight from the governance team or COE. However, it's still appropriate to educate content creators that some general

governance guidelines could still apply to personal content. Valid questions to ask include: Can the creator export the personal report and email it to others? Can the creator store a personal report on a non-organizational laptop or device? What limitations or requirements exist for content that contains sensitive data?

- See the techniques described for business-led self-service, and managed self-service in the content ownership and management article. They're highly relevant techniques that help content creators create efficient and personal data and BI solutions.
- Analyze data from the activity log to discover situations where personal solutions appear to have expanded beyond the original intended usage. It's usually discovered by detecting a significant amount of content sharing from a personal workspace.

> **Tip**
>
> For information about how users progress through the stages of user adoption, see the **Microsoft Fabric adoption roadmap maturity levels**. For more information about using the activity log, see **Tenant-level auditing**.

Team

The Team delivery scope is focused on a team of people who work closely together, and who are tasked with solving closely related problems using the same data. Collaborating and sharing content with each other in a workspace is usually the primary objective.

Content is often shared among the team more informally as compared to departmental or enterprise content. For instance, the workspace is often sufficient for consuming content within a small team. It doesn't require the formality of publishing the workspace to distribute it as an app. There isn't a specific number of users when team-based delivery is considered too informal; each team can find the right number that works for them.

Here are the characteristics of creating content for a team delivery scope.

- Content is created, managed, and viewed among a group of colleagues who work closely together.
- Collaboration and co-management of content is the highest priority.
- Formal delivery of content might occur for report viewers (especially for managers of the team), but it's usually a secondary priority.
- Reports aren't always highly sophisticated or attractive; functionality and accessing the information is what matters most.

Here are some guidelines to help you become successful with content developed for team use.

- Ensure the COE is prepared to support the efforts of self-service creators publishing content for their team.
- Make purposeful decisions about how workspace management will be handled. The workspace is a place to organize related content, a permissions boundary, and the scope for a Power BI app. It's tempting to start with one workspace per team, but that might not be flexible enough to satisfy all needs.
- See the techniques described for business-led self-service and managed self-service in the content ownership and management article. They're highly relevant techniques that help content creators create efficient and effective team data and BI solutions.

> **Tip**
>
> For more information, see **Workspace-level planning**.

Departmental

Content is delivered to members of a department or business unit. Content distribution to a larger number of consumers is a priority for departmental delivery scopes.

Here are the characteristics of departmental content delivery.

- A few content creators typically publish content for colleagues to consume.
- Formal delivery of reports by using Power BI apps is a high priority to ensure consumers have the best experience.
- Additional effort is made to deliver more sophisticated and polished reports. Following best practices for data preparation and higher quality data modeling is also expected.
- Needs for change management and lifecycle management begin to emerge to ensure release stability and a consistent experience for consumers.

Here are a few guidelines to help you become successful with departmental BI delivery.

- Ensure that the COE is prepared to support the efforts of self-service creators. Creators who publish content used throughout their department or business unit might emerge as candidates to become champions. Or, they might become candidates to join the COE as a satellite member.

- Make purposeful decisions about how workspace management will be handled. The workspace is a place to organize related content, a permissions boundary, and the scope for an app. Several workspaces will likely be required to meet all the needs of a large department or business unit.
- Plan how Power BI apps will distribute content to the enterprise. An app can provide a significantly better user experience for consuming content. In many cases, content consumers can be granted permissions to view content via the app only, reserving workspace permissions management for content creators and reviewers only. The use of app audience groups allows you to *mix and match* content and target audience in a flexible way.
- Be clear about what data quality validations have occurred. As the importance and criticality level grows, expectations for trustworthiness grow too.
- Ensure that adequate training, mentoring, and documentation is available to support content creators. Best practices for data preparation, data modeling, and data presentation will result in better quality solutions.
- Provide guidance on the best way to use the promoted endorsement, and when the certified endorsement could be permitted for departmental solutions.
- Ensure that the owner is identified for all departmental content. Clarity on ownership is helpful, including who to contact with questions, feedback, enhancement requests, or support requests. In the Fabric portal, content owners can set the contact list property for many types of items (like reports and dashboards). The contact list is also used in security workflows. For example, when a user is sent a URL to open an app but they don't have permission, they'll be presented with an option to make a request for access.
- Consider using deployment pipelines in conjunction with separate workspaces. Deployment pipelines can support development, test, and production environments, which provide more stability for consumers.
- Consider enforcing the use of sensitivity labels to implement information protection on all content.
- Include consistent branding on reports by:
  - Using departmental colors and styling to indicate who produced the content. For more information, see Content ownership and management.
  - Adding a small image or text label to the report footer, which is valuable when the report is exported from the Fabric portal.
  - Using a standard Power BI Desktop template file. For more information, see Mentoring and user enablement.
- Apply the techniques described for business-led self-service and managed self-service content delivery in the Content ownership and management article. They're highly relevant techniques that can help content creators to create efficient and effective departmental solutions.

Enterprise

Enterprise content is typically managed by a centralized team and is subject to additional governance requirements. Content is delivered broadly across organizational boundaries.

Here are the characteristics of enterprise content delivery.

- A centralized team of experts manages the content end-to-end and publishes it for others to consume.
- Formal delivery of data solutions like reports, lakehouses, and Power BI apps is a high priority to ensure consumers have the best experience.
- The content is highly sensitive, subject to regulatory requirements, or is considered extremely critical.
- Published enterprise-level semantic models and dataflows might be used as a source for self-service creators, thus creating a chain of dependencies to the source data.
- Stability and a consistent experience for consumers are highly important. Application lifecycle management, such as deployment pipelines and DevOps techniques, is commonly used. Change management processes to review and approve changes before they're deployed are commonly used for enterprise content, for example, by a change review board or similar group.
- Processes exist to gather requirements, prioritize efforts, and plan for new projects or enhancements to existing content.
- Integration with other enterprise-level data architecture and management services could exist, possibly with other Azure services and Power Platform products.

Here are some guidelines to help you become successful with enterprise content delivery.

- Governance and oversight techniques described in the governance article are relevant for managing an enterprise solution. Techniques primarily include change management and lifecycle management.
- Plan for how to effectively use Premium Per User or Fabric capacity licensing per workspace. Align your workspace management strategy, like how workspaces will be organized and secured, to the planned licensing strategy.
- Plan how Power BI apps will distribute enterprise content to consumers. An app can provide a significantly better user experience for consuming content. Align the app distribution strategy with your workspace management strategy.
- Consider enforcing the use of sensitivity labels to implement information protection on all content.

- Implement a rigorous process for use of the certified endorsement for enterprise reports and apps. Data assets can be certified, too, when there's the expectation that self-service creators will build solutions based on them. Not all enterprise content needs to be certified, but much of it probably will be.
- Make it a common practice to announce when changes will occur. For more information, see the community of practice article for a description of communication types.
- Include consistent branding on reports, by:
  - Using specific colors and styling, which can also indicate who produced the content. For more information, see Content ownership and management.
  - Adding a small image or text label to the report footer, which can be valuable when the report is exported from the Fabric portal.
  - Using a standard Power BI Desktop template file. For more information, see Mentoring and user enablement.
- Actively use the lineage view to understand dependencies, perform impact analysis, and communicate to downstream content owners when changes will occur.
- See the techniques described for enterprise content delivery in the content ownership and management article. They're highly relevant techniques that help content creators create efficient and effective enterprise solutions.
- See the techniques described in the system oversight article for auditing, governing, and the oversight of enterprise content.

Considerations and key actions

Checklist - Considerations and key actions you can take to strengthen your approach to content delivery.

- ✓ **Align goals for content delivery**: Ensure that guidelines, documentation, and other resources align with the strategic goals defined for Fabric adoption.
- ✓ **Clarify the scopes for content delivery in your organization**: Determine who each scope applies to, and how each scope aligns with governance decisions. Ensure that decisions and guidelines are consistent with how content ownership and management is handled.
- ✓ **Consider exceptions**: Be prepared for how to handle situations when a smaller team wants to publish content for an enterprise-wide audience.

  - Will it require the content be owned and managed by a centralized team? For more information, see the Content ownership and management article, which describes an inter-related concept with content delivery scope.
  - Will there be an approval process? Governance can become more complicated when the content delivery scope is broader than the owner of the content. For example, when an app that's owned by a divisional sales team is distributed to the entire organization.
- ✓ **Create helpful documentation**: Ensure that you have sufficient training documentation and support so that your content creators understand when it's appropriate to use workspaces, apps, or per-item sharing (direct access or link).
- ✓ **Create a licensing strategy**: Ensure that you have a specific strategy in place to handle Fabric licensing considerations. Create a process for how workspaces could be assigned each license type, and the prerequisites required for the type of content that could be assigned to Premium.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Questions to ask

Use questions like those found below to assess content delivery scope.

- Do central teams that are responsible for Fabric have a clear understanding of who creates and delivers content? Does it differ by business area, or for different content item types?
- Which usage scenarios are in place, such as personal BI, team BI, departmental BI, or enterprise BI? How prevalent are they in the organization? Are there advanced scenarios, like advanced data preparation or advanced data model management, or niche scenarios, like self-service real-time analytics?

- For the identified content delivery scopes in place, to what extent are guidelines being followed?
- Are there trajectories for helpful self-service content to be "promoted" from personal to team content delivery scopes and beyond? What systems and processes enable sustainable, bottom-up scaling and distribution of useful self-service content?
- What are the guidelines for publishing content to, and using, personal workspaces?
- Are personal workspaces assigned to dedicated Fabric capacity? In what circumstances are personal workspaces intended to be used?
- On average, how many reports does someone have access to? How many reports does an executive have access to? How many reports does the CEO have access to?
- If your organization is using Fabric or Power BI today, does the current workspace setup comply with the content ownership and delivery strategies that are in place?
- Is there a clear licensing strategy? How many licenses are used today? How many tenants and capacities exist, who uses them, and why?
- How do central teams decide what gets published to Premium (or Fabric) dedicated capacity, and what uses shared capacity? Do development workloads use separate Premium Per User (PPU) licensing to avoid affecting production workloads?

Maturity levels

The following maturity levels will help you assess the current state of your content delivery.

Expand table

Level	State of content delivery
100: Initial	• Content is published for consumers by self-service creators in an uncontrolled way, without a specific strategy.
200: Repeatable	• Pockets of good practices exist. However, good practices are overly dependent on the knowledge, skills, and habits of the content creator.

Level	State of content delivery
300: Defined	• Clear guidelines are defined and communicated to describe what can and can't occur within each delivery scope. These guidelines are followed by some—but not all—groups across the organization.
400: Capable	• Criteria are defined to align governance requirements for self-service versus enterprise content. • Guidelines for content delivery scope are followed by most, or all, groups across the organization. • Change management requirements are in place to approve critical changes for content that's distributed to a larger-sized audience. • Changes are announced and follow a communication plan. Content creators are aware of the downstream effects on their content. Consumers are aware of when reports and apps are changed.
500: Efficient	• Proactively take steps to communicate with users occur when any concerning activities are detected in the activity log. Education and information are provided to make gradual improvements or reduce risk. • The business value that's achieved for deployed solutions is regularly evaluated.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about the Center of Excellence (COE).

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Center of Excellence

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

A data or analytics Center of Excellence (COE) is an internal team of technical and business experts. The team actively assists others within the organization who are working with data. The COE forms the nucleus of the broader community to advance adoption goals, which align with the data culture vision.

A COE might also be known as *competency center*, *capability center*, or a *center of expertise*. Some organizations use the term *squad*. Many organizations perform the COE responsibilities within their data, analytics, or business intelligence (BI) team.

> **Note**
>
> Having a COE team formally recognized in your organizational chart is recommended, but not required. What's most important is that the COE roles and responsibilities are identified, prioritized, and assigned. It's common for a centralized data or analytics team to take on many of the COE responsibilities; some responsibilities might also reside within IT. For simplicity, in this series of articles, COE means a *specific group of people*, although you might implement it differently. It's also very common to implement the COE with a scope broader than Fabric or Power BI alone: for instance, a Power Platform COE, a data COE, or an analytics COE.

Goals for a COE

Goals for a COE include:

- Evangelizing a data-driven culture.
- Promoting the adoption of analytics.
- Nurturing, mentoring, guiding, and educating internal users to increase their skills and level of self-reliance.

- Coordinating efforts and disseminating knowledge across organizational boundaries.
- Creating consistency and transparency for the user community, which reduces friction and pain points related to finding relevant data and analytics content.
- Maximizing the benefits of self-service BI, while reducing the risks.
- Reducing technical debt by helping users make good decisions that increase consistency and result in fewer inefficiencies.

> **Important**
>
> One of the most powerful aspects of a COE is the cross-departmental insight into how analytics tools like Fabric are used by the organization. This insight can reveal which practices work well and which don't, that can facilitate a bottom-up approach to governance. A primary goal of the COE is to learn which practices work well, share that knowledge more broadly, and replicate best practices across the organization.

Scope of COE responsibilities

The scope of COE responsibilities can vary significantly between organizations. In a way, a COE can be thought of as a consultancy service because its members routinely provide expert advice to the internal community of users. To varying degrees, most COEs handle hands-on work too.

Common COE responsibilities include:

- Mentoring and facilitating knowledge sharing within the internal Fabric community.
- Holding office hours to engage with the internal Fabric community.
- Conducting co-development projects and best practices reviews in order to actively help business units deliver solutions.
- Managing the centralized portal.
- Producing, curating, and promoting training materials.
- Creating documentation and other resources, such as template files, to encourage consistent use of standards and best practices.
- Applying, communicating, and assisting with governance guidelines.
- Handling and assisting with system oversight and Fabric administration.
- Responding to user support issues escalated from the help desk.
- Developing solutions and/or proofs of concept.
- Establishing and maintaining the BI platform and data architecture.

- Communicating regularly with the internal community of users.

Staffing a COE

People who are good candidates as COE members tend to be those who:

- Understand the analytics vision for the organization.
- Have a desire to continually improve analytics practices for the organization.
- Have a deep interest in, and expertise with, analytics tools such as Fabric.
- Are interested in seeing Fabric used effectively and adopted successfully throughout the organization.
- Take the initiative to continually learn, adapt, and grow.
- Readily share their knowledge with others.
- Are interested in repeatable processes, standardization, and governance with a focus on user enablement.
- Are hyper-focused on collaboration with others.
- Are comfortable working in an agile fashion.
- Have an inherent interest in being involved and helping others.
- Can effectively translate business needs into solutions.
- Communicate well with both technical and business colleagues.

> **Tip**
>
> If you have self-service content creators in your organization who constantly push the boundaries of what can be done, they might be a great candidate to become a recognized **champion**, or perhaps even a satellite member of the COE.

When recruiting for the COE, it's important to have a mix of complementary analytical skills, technical skills, and business skills.

Roles and responsibilities

Very generalized roles within a COE are listed below. It's common for multiple people to overlap roles, which is useful from a backup and cross-training perspective. It's also common for the same person to serve multiple roles. For instance, most COE members also serve as a coach or mentor.

Expand table

Role	Description
COE leader	Manages the day-to-day operations of the COE. Interacts with the executive sponsor and other organizational teams, such as the data governance board, as necessary. For an overview of additional roles and responsibilities, see the Governance article.
Coach	Coaches and educates others on data and BI skills via office hours (community engagement), best practices reviews, or co-development projects. Oversees and participates in the discussion channel of the internal community. Interacts with, and supports, the champions network.
Trainer	Develops, curates, and delivers internal training materials, documentation, and resources.
Data analyst	Domain-specific subject matter expert. Acts as a liaison between the COE and the business unit. Content creator for the business unit. Assists with content certification. Works on co-development projects and proofs of concept.
Data modeler	Creates and manages data assets (such as shared semantic model and dataflows) to support other self-service content creators.
Report creator	Creates and publishes reports, dashboards, and metrics.
Data engineer	Plans for deployment and architecture, including integration with other services and data platforms. Publishes data assets which are utilized broadly across the organization (such as a lakehouse, data warehouse, data pipeline, dataflow, or semantic model).
User support	Assists with the resolution of data discrepancies and escalated help desk support issues.

As mentioned previously, the scope of responsibilities for a COE can vary significantly between organizations. Therefore, the roles found for COE members can vary too.

Structuring a COE

The selected COE structure can vary among organizations. It's also possible for multiple structures to exist inside of a single large organization. That's particularly true when there are subsidiaries or when acquisitions have occurred.

> **Note**
>
> The following terms might differ to those defined for your organization, particularly the meaning of *federated*, which tends to have many different IT-related meanings.

Centralized COE

A centralized COE comprises a single shared services team.

Pros:

- There's a single point of accountability for a single team that manages standards, best practices, and delivery end-to-end.
- The COE is one group from an organizational chart perspective.
- It's easy to start with this approach and then evolve to the unified or federated model over time.

Cons:

- A centralized team might have an authoritarian tendency to favor one-size-fits-all decisions that don't always work well for all business units.
- There can be a tendency to prefer IT skills over business skills.
- Due to the centralized nature, it might be more difficult for the COE members to sufficiently understand the needs of all business units.

Unified COE

A unified COE is a single, centralized, shared services team that has been expanded to include embedded team members. The embedded team members are dedicated to supporting a specific functional area or business unit.

Pros:

- There's a single point of accountability for a single team that includes cross-functional involvement from the embedded COE team members. The embedded COE team members are assigned to various areas of the business.
- The COE is one group from an organizational chart perspective.
- The COE understands the needs of business units more deeply due to dedicated members with domain expertise.

Cons:

- The embedded COE team members, who are dedicated to a specific business unit, have a different organizational chart responsibility than the people they serve directly within the business unit. The organizational structure could potentially lead to complications, differences in priorities, or necessitate the involvement of the executive sponsor. Preferably, the executive sponsor has a scope of authority that includes the COE and all involved business units to help resolve conflicts.

Federated COE

A federated COE comprises a shared services team (the core COE members) plus satellite members from each functional area or major business unit. A federated team works in coordination, even though its members reside in different business units. Typically, satellite members are primarily focused on development activities to support their business unit while the shared services personnel support the entire community.

Pros:

- There's cross-functional involvement from satellite COE members who represent their specific functional area and have domain expertise.
- There's a balance of centralized and decentralized representation across the core and satellite COE members.
- When distributed data ownership situations exist—as could be the case when business units take direct responsibility for data management activities—this model is effective.

Cons:

- Since core and satellite members span organizational boundaries, the federated COE approach requires strong leadership, excellent communication, robust project management, and ultra-clear expectations.
- There's a higher risk of encountering competing priorities due to the federated structure.
- This approach typically involves part-time people and/or *dotted line* organizational chart accountability that can introduce competing time pressures.

> **Tip**
>
> Some organizations have success by using a *rotational program*. It involves federated members joining the core COE for a period of time, such as six months. This type of program allows federated members to learn best practices and understand more deeply how and why things are done. Although each federated member remains focused on their specific business unit, they gain a deeper understanding of the organization's challenges. This deeper understanding leads to a more productive partnership over time.

Decentralized COE

Decentralized COEs are independently managed by business units.

Pros:

- A specialized data culture exists that's focused on the business unit, making it easier to learn quickly and adapt.
- Policies and practices are tailored to each business unit.
- Agility, flexibility, and priorities are focused on the individual business unit.

Cons:

- There's a risk that decentralized COEs operate in isolation. As a result, they might not share best practices and lessons learned outside of their business unit.
- Collaboration with a centralized team might be informal and/or inconsistent.
- Inconsistent policies are created and applied across business units.
- It's difficult to scale a decentralized model.
- There's potential rework to bring one or more decentralized COEs in alignment with organizational-wide policies.
- Larger business units with significant funding might have more resources available to them, which might not serve cost optimization goals from an organizational-wide perspective.

> **ⓘ Important**
>
> A highly centralized COE tends to be more *authoritarian*, while highly decentralized COEs tend to be more *siloed*. Each organization will need to weigh the pros and cons that apply to them to determine the best choice. For most organizations, the most effective approach tends to be the unified or federated, which bridges organizational boundaries.

Funding the COE

The COE might obtain its operating budget in multiple ways:

- Cost center.
- Profit center with project budget(s).
- A combination of cost center and profit center.

When the COE operates as a cost center, it absorbs the operating costs. Generally, it involves an approved annual budget. Sometimes this is called a *push* engagement model.

When the COE operates as a profit center (for at least part of its budget), it could accept projects throughout the year based on funding from other business units. Sometimes

this is called a *pull* engagement model.

Funding is important because it impacts the way the COE communicates and engages with the internal community. As the COE experiences more and more successes, they might receive more requests from business units for help. It's especially the case as awareness grows throughout the organization.

> **Tip**
>
> The choice of funding model can determine how the COE actively grows its influence and ability to help. The funding model can also have a big impact on where authority resides and how decision-making works. Further, it impacts the types of services a COE can offer, such as co-development projects and/or best practices reviews. For more information, see the **Mentoring and user enablement** article.

Some organizations cover the COE operating costs with chargebacks to business units based on the usage goals of Fabric. For a shared capacity, this could be based on number of active users. For Premium capacity, chargebacks could be allocated based on which business units are using the capacity. Ideally, chargebacks are directly correlated to the business value gained.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Considerations and key actions

Checklist - Considerations and key actions you can take to establish or improve your COE.

- ✓ **Define the scope of responsibilities for the COE**: Ensure that you're clear on what activities the COE can support. Once the scope of responsibilities is known, identify the skills and competencies required to fulfill those responsibilities.
- ✓ **Identify gaps in the ability to execute**: Analyze whether the COE has the required systems and infrastructure in place to meet its goals and scope of responsibilities.
- ✓ **Determine the best COE structure**: Identify which COE structure is most appropriate (centralized, unified, federated, or decentralized). Verify that staffing, roles and responsibilities, and appropriate organizational chart relationships (HR reporting) are in place.
- ✓ **Plan for future growth**: If you're starting out with a centralized or decentralized COE, consider how you will scale the COE over time by using the unified or federated approach. Plan for any actions that you can take now that'll facilitate future growth.
- ✓ **Identify customers**: Identify the internal community members, and any external customers, to be served by the COE. Decide how the COE will generally engage with those customers, whether it's a push model, pull model, or both models.
- ✓ **Verify the funding model for the COE**: Decide whether the COE is purely a cost center with an operating budget, whether it will operate partially as a profit center, and/or whether chargebacks to other business units will be required.
- ✓ **Create a communication plan**: Create you communications strategy to educate the internal community of users about the services the COE offers, and how to engage with the COE.
- ✓ **Create goals and metrics**: Determine how you'll measure effectiveness for the COE. Create KPIs (key performance indicators) or OKRs (objectives and key results) to validate that the COE consistently provides value to the user community.

Questions to ask

Use questions like those found below to assess the effectiveness of a COE.

- Is there a COE? If so, who is in the COE and what's the structure?
- If there isn't a COE, is there a central team that performs a similar function? Do data decision makers in the organization understand what a COE does?
- If there isn't a COE, does the organization aspire to create one? Why or why not?
- Are there opportunities for federated or decentralized COE models due to a mix of enterprise and departmental solutions?

- Are there any missing roles and responsibilities from the COE?
- To what extent does the COE engage with the user community? Do they mentor users? Do they curate a centralized portal? Do they maintain centralized resources?
- Is the COE recognized in the organization? Does the user community consider them to be credible and helpful?
- Do business users see central teams as enabling or restricting their work with data?
- What's the COE funding model? Do COE customers financially contribute in some way to the COE?
- How consistent and transparent is the COE with their communication?

Maturity levels

The following maturity levels will help you assess the current state of your COE.

Expand table

Level	State of the Center of Excellence
100: Initial	• One or more COEs exist, or the activities are performed within the data team, BI team, or IT. There's no clarity on the specific goals nor expectations for responsibilities. • Requests for assistance from the COE are handled in an unplanned manner.
200: Repeatable	• The COE is in place with a specific charter to mentor, guide, and educate self-service users. The COE seeks to maximize benefits of self-service approaches to data and BI while reducing the risks. • The goals, scope of responsibilities, staffing, structure, and funding model are established for the COE.
300: Defined	• The COE operates with active involvement from all business units in a unified or federated mode.
400: Capable	• The goals of the COE align with organizational goals, and they are reassessed regularly. • The COE is well-known throughout the organization, and consistently proves its value to the internal user community.

Level	State of the Center of Excellence
500: Efficient	• Regular reviews of KPIs or OKRs evaluate COE effectiveness in a measurable way. • Agility and implementing continual improvements from lessons learned (including scaling out methods that work) are top priorities for the COE.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about implementing governance guidelines, policies, and processes.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Governance

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

Data governance is a broad and complex topic. This article introduces key concepts and considerations. It identifies important actions to take when adopting Microsoft Fabric, but it's not a comprehensive reference for data governance.

As defined by the Data Governance Institute, data governance is "a system of decision rights and accountabilities for information-related processes, executed according to agreed-upon models which describe who can take what actions, with what information, and when, under what circumstances, using what methods."

The term *data governance* is a misnomer. The primary focus for governance isn't on the data itself. The focus is on governing *what users do with the data*. Put another way: the true focus is on governing user's behavior to ensure organizational data is well managed.

When focused on self-service data and business intelligence (BI), the primary goals of governance are to achieve the proper balance of:

- **User empowerment**: Empower the internal user community to be productive and efficient, within requisite guardrails.
- **Regulatory compliance**: Comply with the organization's industry, governmental, and contractual regulations.
- **Internal requirements**: Adhere to the organization's internal requirements.

The optimal balance between control and empowerment will differ between organizations. It's also likely to differ among different business units within an organization. You'll be most successful with a platform like Fabric when you put as much emphasis on user empowerment as on clarifying its practical usage within established guardrails.

> **Tip**

Think of governance as a set of established guidelines and formalized policies. All governance guidelines and policies should align with your organizational **data culture** and adoption objectives. Governance is enacted on a day-to-day basis by your **system oversight** (administration) activities.

Governance strategy

When considering data governance in any organization, the best place to start is by defining a governance strategy. By focusing first on the strategic goals for data governance, all detailed decisions when implementing governance policies and processes can be informed by the strategy. In turn, the governance strategy will be defined by the organization's data culture.

Governance decisions are implemented with documented guidance, policies, and processes. Objectives for governance of a self-service data and BI platform, such as Fabric, include:

- Empowering users throughout the organization to use data and make decisions, within the defined boundaries.
- Improving the user experience by providing clear and transparent guidance (with minimal friction) on what actions are permitted, why, and how.
- Ensuring that the data usage is appropriate for the needs of the business.
- Ensuring that content ownership and stewardship responsibilities are clear. For more information, see the Content ownership and management article.
- Enhancing the consistency and standardization of working with data across organizational boundaries.
- Reducing risk of data leakage and misuse of data. For more information, see the information protection and data loss prevention series of articles article.
- Meeting regulatory, industry, and internal requirements for the proper use of data.

Tip

A well-executed data governance strategy makes it easier for more users to work with data. When governance is approached from the perspective of user empowerment, users are more likely to follow the documented processes. Accordingly, the users become a trusted partner too.

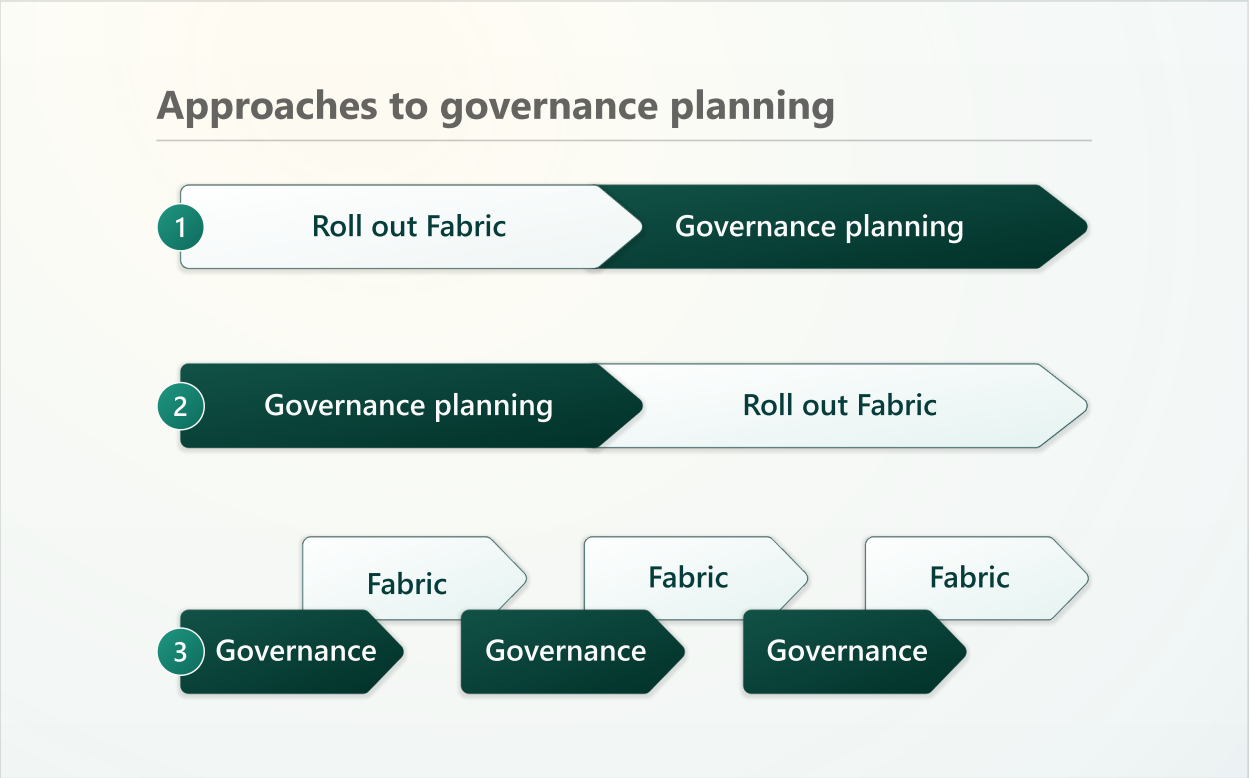
Governance success factors

Governance isn't well-received when it's enacted with top-down mandates that are focused more on control than empowerment. Governing Fabric is most successful when:

- The most lightweight governance model that accomplishes required objectives is used.
- Governance is approached on an iterative basis and doesn't significantly impede productivity.
- A bottom-up approach to formulating governance guidelines is used whenever practical. The Center of Excellence (COE) and/or the data governance team observes successful behaviors that are occurring within a business unit. The COE then takes action to scale out to other areas of the organization.
- Governance decisions are co-defined with input from different business units before they're enacted. Although there are times when a specific directive is necessary (particularly in heavily regulated industries), mandates should be the exception rather than the rule.
- Governance needs are balanced with flexibility and the ability to be productive.
- Governance requirements can be satisfied as part of users' regular workflow, making it easier for users to do the right thing in the right way with little friction.
- The answer to new requests for data isn't "no" by default, but rather "yes and" with clear, simple, transparent rules for what governance requirements are for data access, usage, and sharing.
- Users that need access to data have incentive to do so through normal channels, complying with governance requirements, rather than circumventing them.
- Governance decisions, policies, and requirements for users to follow are in alignment with organizational data culture goals as well as other existing data governance initiatives.
- Decisions that affect what users can—and can't—do aren't made solely by a system administrator.

Introduce governance to your organization

There are three primary timing methods organizations take when introducing Fabric governance to an organization.



The methods in the above diagram include:

Expand table

Method	Strategy followed
1	**Roll out Fabric first, then introduce governance**: Fabric is made widely available to users in the organization as a new self-service data and BI tool. Then, at some time in the future, a governance effort begins. This method prioritizes agility.
2	**Full governance planning first, then roll out Fabric**: Extensive governance planning occurs prior to permitting users to begin using Fabric. This method prioritizes control and stability.
3	**Iterative governance planning with rollouts of Fabric in stages**: Just enough governance planning occurs initially. Then Fabric is iteratively rolled out in stages to individual teams while iterative governance enhancements occur. This method equally prioritizes agility and governance.

Choose method 1 when Fabric is already used for self-service scenarios, and you're ready to start working in a more efficient manner.

Choose method 2 when your organization already has a well-established approach to governance that can be readily expanded to include Fabric.

Choose method 3 when you want to have a balance of control agility. This balanced approach is the best choice for most organizations and most scenarios.

Each method is described in the following sections.

Method 1: Roll out Fabric first

Method 1 prioritizes agility and speed. It allows users to quickly get started creating solutions. This method occurs when Fabric has been made widely available to users in the organization as a new self-service data and BI tool. Quick wins and some successes are achieved. At some point in the future, a governance effort begins, usually to bring order to an unacceptable level of chaos since the self-service user population didn't receive sufficient guidance.

Pros:

- Fastest to get started
- Highly capable users can get things done quickly
- Quick wins are achieved

Cons:

- Higher effort to establish governance once Fabric is used prevalently throughout the organization
- Resistance from self-service users who are asked to change what they've been doing
- Self-service users need to figure out things on their own, which is inefficient and results in inconsistencies
- Self-service users need to use their best judgment, which produces technical debt to be resolved

See other possible cons in the Governance challenges section below.

Method 2: In-depth governance planning first

Method 2 prioritizes control and stability. It lies at the opposite end of the spectrum from method 1. Method 2 involves doing extensive governance planning before rolling out Fabric. This situation is most likely to occur when the implementation of Fabric is led by IT. It's also likely to occur when the organization operates in a highly regulated industry, or when an existing data governance board imposes significant prerequisites and up-front requirements.

Pros:

- More fully prepared to meet regulatory requirements
- More fully prepared to support the user community

Cons:

- Favors enterprise content development more than self-service
- Slower to allow the user population to begin to get value and improve decision-making
- Encourages poor habits and workarounds when there's a significant delay in allowing the use of data for decision-making

Method 3: Iterative governance with rollouts

Method 3 seeks a balance between agility and governance. It's an ideal scenario that does *just enough* governance planning upfront. Frequent and continual governance improvements iteratively occur over time alongside Fabric development projects that deliver value.

Pros:

- Puts equal priority on governance and user productivity
- Emphasizes a *learning as you go* mentality
- Encourages iterative releases to groups of users in stages

Cons:

- Requires a high level of communication to be successful with agile governance practices
- Requires additional discipline to keep documentation and training current
- Introducing new governance guidelines and policies too often causes a certain level of user disruption

For more information about up-front planning, see the Preparing to migrate to Power BI article.

Governance challenges

If your organization has implemented Fabric without a governance approach or strategic direction (as described above by method 1), there could be numerous challenges requiring attention. Depending on the approach that you've taken and your current state, some of the following challenges could be applicable to your organization.

Strategy challenges

- Lack of a cohesive data governance strategy that aligns with the business strategy
- Lack of executive support for governing data as a strategic asset

- Insufficient adoption planning for advancing adoption and the maturity level of BI and analytics

People challenges

- Lack of aligned priorities between centralized teams and business units
- Lack of identified champions with sufficient expertise and enthusiasm throughout the business units to advance organizational adoption objectives
- Lack of awareness of self-service best practices
- Resistance to following newly introduced governance guidelines and policies
- Duplicate effort spent across business units
- Lack of clear accountability, roles, and responsibilities

Process challenges

- Lack of clearly defined processes resulting in chaos and inconsistencies
- Lack of standardization or repeatability
- Insufficient ability to communicate and share lessons learned
- Lack of documentation and over-reliance on tribal knowledge
- Inability to comply with security and privacy requirements

Data quality and data management challenges

- Sprawl of data and reports
- Inaccurate, incomplete, or outdated data
- Lack of trust in the data, especially for content produced by self-service content creators
- Inconsistent reports produced without sufficient data validation
- Valuable data not used or difficult to access
- Fragmented, siloed, and duplicated data
- Lack of data catalog, inventory, glossary, or lineage
- Unclear data ownership and stewardship

Skills and data literacy challenges

- Varying levels of ability to interpret, create, and communicate with data effectively
- Varying levels of technical skillsets and skill gaps
- Lack of ability to confidently manage data diversity and volume
- Underestimating the level of complexity for BI solution development and management throughout its entire lifecycle

- Short tenure with continual staff transfers and turnover
- Coping with the speed of change for cloud services

> **Tip**
>
> Identifying your current challenges—as well as your strengths—is essential to do proper governance planning. There's no single straightforward solution to the challenges listed above. Each organization needs to find the right balance and approach that solves the challenges that are most important to them. The challenges presented above will help you identify how they might affect your organization, so you can start thinking about what the right solution is for your circumstances.

Governance planning

Some organizations have implemented Fabric without a governance approach or clear strategic direction (as described above by method 1). In this case, the effort to begin governance planning can be daunting.

If a formal governance body doesn't currently exist in your organization, then the focus of your governance planning and implementation efforts will be broader. If, however, there's an existing data governance board in the organization, then your focus is primarily to integrate with existing practices and customize them to accommodate the objectives for self-service and enterprise data and BI scenarios.

> **Important**
>
> Governance is a big undertaking, and it's never completely *done*. Relentlessly prioritizing and iterating on improvements will make the scope more manageable. If you track your progress and accomplishments each week and each month, you'll be amazed at the impact over time. The **maturity levels** at the end of each article in this series can help you to assess where you are currently.

Some potential governance planning activities and outputs that you might find valuable are described next.

Strategy

Key activities:

- Conduct a series of workshops to gather information and assess the current state of data culture, adoption, and data and BI practices. For guidance about how to gather information and define the current state of BI adoption, including governance, see BI strategic planning.
- Use the current state assessment and information gathered to define the desired future state, including governance objectives. For guidance about how to use this current state definition to decide on your desired future state, see BI tactical planning.
- Validate the focus and scope of the governance program.
- Identify existing bottom-up initiatives in progress.
- Identify immediate pain points, issues, and risks.
- Educate senior leadership about governance, and ensure executive sponsorship is sufficient to sustain and grow the program.
- Clarify where Power BI fits in to the overall BI and analytics strategy for the organization.
- Assess internal factors such as organizational readiness, maturity levels, and key challenges.
- Assess external factors such as risk, exposure, regulatory, and legal requirements—including regional differences.

Key output:

- Business case with cost/benefit analysis
- Approved governance objectives, focus, and priorities that are in alignment with high-level business objectives
- Plan for short-term goals and priorities (quick wins)
- Plan for long-term and deferred goals and priorities
- Success criteria and measurable key performance indicators (KPIs)
- Known risks documented with a mitigation plan
- Plan for meeting industry, governmental, contractual, and regulatory requirements that impact BI and analytics in the organization
- Funding plan

People

Key activities:

- Establish a governance board and identify key stakeholders.
- Determine focus, scope, and a set of responsibilities for the governance board.
- Establish a COE.
- Determine focus, scope, and a set of responsibilities for COE.
- Define roles and responsibilities.

- Confirm who has decision-making, approval, and veto authority.

Key output:

- Charter for the governance board
- Charter and priorities for the COE
- Staffing plan
- Roles and responsibilities
- Accountability and decision-making matrix
- Communication plan
- Issue management plan

Policies and processes

Key activities:

- Analyze immediate pain points, issues, risks, and areas to improve the user experience.
- Prioritize data policies to be addressed by order of importance.
- Identify existing processes in place that work well and can be formalized.
- Determine how new data policies will be socialized.
- Decide to what extent data policies might differ or be customized for different groups.

Key output:

- Process for how data policies and documentation will be defined, approved, communicated, and maintained
- Plan for requesting valid exceptions and departures from documented policies

Project management

The implementation of the governance program should be planned and managed as a series of projects.

Key activities:

- Establish a timeline with priorities and milestones.
- Identify related initiatives and dependencies.
- Identify and coordinate with existing bottom-up initiatives.
- Create an iterative project plan that's aligned with high-level prioritization.
- Obtain budget approval and funding.
- Establish a tangible way to track progress.

Key output:

- Project plan with iterations, dependencies, and sequencing
- Cadence for retrospectives with a focus on continual improvements

> **ⓘ Important**
>
> The scope of activities listed above that will be useful to take on will vary considerably between organizations. If your organization doesn't have existing processes and workflows for creating these types of outputs, refer to the guidance found in the **adoption roadmap conclusion** for some helpful resources, as well as the **implementation planning BI strategy articles**.

Governance policies

Decision criteria

All governance decisions should be in alignment with the established goals for organizational adoption. Once the strategy is clear, more tactical governance decisions will need to be made which affect the day-to-day activities of the self-service user community. These types of tactical decisions correlate directly to the data policies that get created.

How we go about making governance decisions depends on:

- **Who owns and manages the data and BI content?** The Content ownership and management article introduced three types of strategies: business-led self-service, managed self-service, and enterprise. Who owns and manages the content has a significant impact on governance requirements.
- **What is the scope for delivery of the data and BI content?** The Content delivery scope article introduced four scopes for delivery of content: personal, team, departmental, and enterprise. The scope of delivery has a considerable impact on governance requirements.
- **What is the data subject area?** The data itself, including its sensitivity level, is an important factor. Some data domains inherently require tighter controls. For instance, personally identifiable information (PII), or data subject to regulations, should be subject to stricter governance requirements than less sensitive data.
- **Is the data, and/or the BI solution, considered critical?** If you can't make an informed decision easily without this data, you're dealing with critical data elements. Certain reports and apps could be deemed critical because they meet a

set of predefined criteria. For instance, the content is delivered to executives. Predefined criteria for what's considered *critical* helps everyone have clear expectations. Critical data is usually subject to stricter governance requirements.

> **Tip**
>
> Different combinations of the above four criteria will result in different governance requirements for Fabric content.

Key Fabric governance decisions

As you explore your goals and objectives and pursue more tactical data governance decisions as described above, it will be important to determine what the highest priorities are. Deciding where to focus your efforts can be challenging.

The following list includes items that you might choose to prioritize when introducing governance for Fabric.

- Recommendations and requirements for content ownership and management
- Recommendations and requirements for content delivery scope
- Recommendations and requirements for content distribution and sharing with colleagues, as well as for external users, such as customers, partners, or vendors
- How users are permitted to work with regulated data and highly sensitive data
- Allowed use of unverified data sources that are unknown to IT
- When manually maintained data sources, such as Excel or flat files, are permitted
- Who is permitted to create a workspace
- How to manage workspaces effectively
- How personal workspaces are effectively used
- Which workspaces are assigned to Fabric capacity
- Who is allowed to be a Fabric administrator
- Security, privacy, and data protection requirements, and allowed actions for content assigned to each sensitivity label
- Allowed or encouraged use of personal gateways
- Allowed or encouraged use of self-service purchasing of user licenses
- Requirements for who can certify content, as well as requirements that must be met
- Application lifecycle management for managing content through its entire lifecycle, including development, test, and production stages
- Additional requirements applicable to critical content, such as data quality verifications and documentation

- Requirements to use standardized master data and common data definitions to improve consistency across data assets
- Recommendations and requirements for use of external tools by advanced content creators

If you don't make governance decisions and communicate them well, users will use their own judgment for how things should work—and that often results in inconsistent approaches to common tasks.

Although not every governance decision needs to be made upfront, it's important that you identify the areas of greatest risk in your organization. Then, incrementally implement governance policies and processes that will deliver the most impact.

Data policies

A data policy is a document that defines what users can and can't do. You might call it something different, but the goal remains the same: when decisions—such as those discussed in the previous section—are made, they're documented for use and reference by the community of users.

A data policy should be as short as possible. That way, it's easy for people to understand what is being asked of them.

A data policy should include:

- Policy name, purpose, description, and details
- Specific responsibilities
- Scope of the policy (organization-wide versus departmental-specific)
- Audience for the policy
- Policy owner, approver, and contact
- How to request an exception
- How the policy will be audited and enforced
- Regulatory or legal requirements met by the policy
- Reference to terminology definitions
- Reference to any related guidelines or policies
- Effective date, last revision date, and change log

> **Note**
>
> Locate, or link to, data policies from your centralized portal.

Here are three common data policy examples you might choose to prioritize.

Expand table

Policy	Description
Data ownership policy	Specifies when an owner is required for a data asset, and what the data owner's responsibilities include, such as: supporting colleagues who view the content, maintaining appropriate confidentiality and security, and ensuring compliance.
Data certification (endorsement) policy	Specifies the process that is followed to certify content. Requirements might include activities such as: data accuracy validation, data source and lineage review, technical review of the data model, security review, and documentation review.
Data classification and protection policy	Specifies activities that are allowed and not allowed per classification (sensitivity level). It should specify activities such as: allowed sharing with external users, with or without a non-disclosure agreement (NDA), encryption requirements, and ability to download the data. Sometimes, it's also called a *data handling policy* or a *data usage policy*. For more information, see the Information protection for Power BI article.

⊗ Caution

Having a lot of documentation can lead to a false sense that everything is under control, which can lead to complacency. The level of engagement that the COE has with the user community is one way to improve the chances that governance guidelines and policies are consistently followed. Auditing and monitoring activities are also important.

Scope of policies

Governance decisions will rarely be one-size-fits-all across the entire organization. When practical, it's wise to start with standardized policies, and then implement exceptions as needed. Having a clearly defined strategy for how policies will be handled for centralized and decentralized teams will make it much easier to determine how to handle exceptions.

Pros of organization-wide policies:

- Much easier to manage and maintain
- Greater consistency
- Encompasses more use cases
- Fewer policies overall

Cons of organization-wide policies:

- Inflexible
- Less autonomy and empowerment

Pros of departmental-scope policies:

- Expectations are clearer when tailored to a specific group
- Customizable and flexible

Cons of departmental-scope policies:

- More work to manage
- More policies that are siloed
- Potential for conflicting information
- Difficult to scale more broadly throughout the organization

> **Tip**
>
> Finding the right balance of standardization and customization for supporting self-service data and BI across the organization can be challenging. However, by starting with organizational policies and mindfully watching for exceptions, you can make meaningful progress quickly.

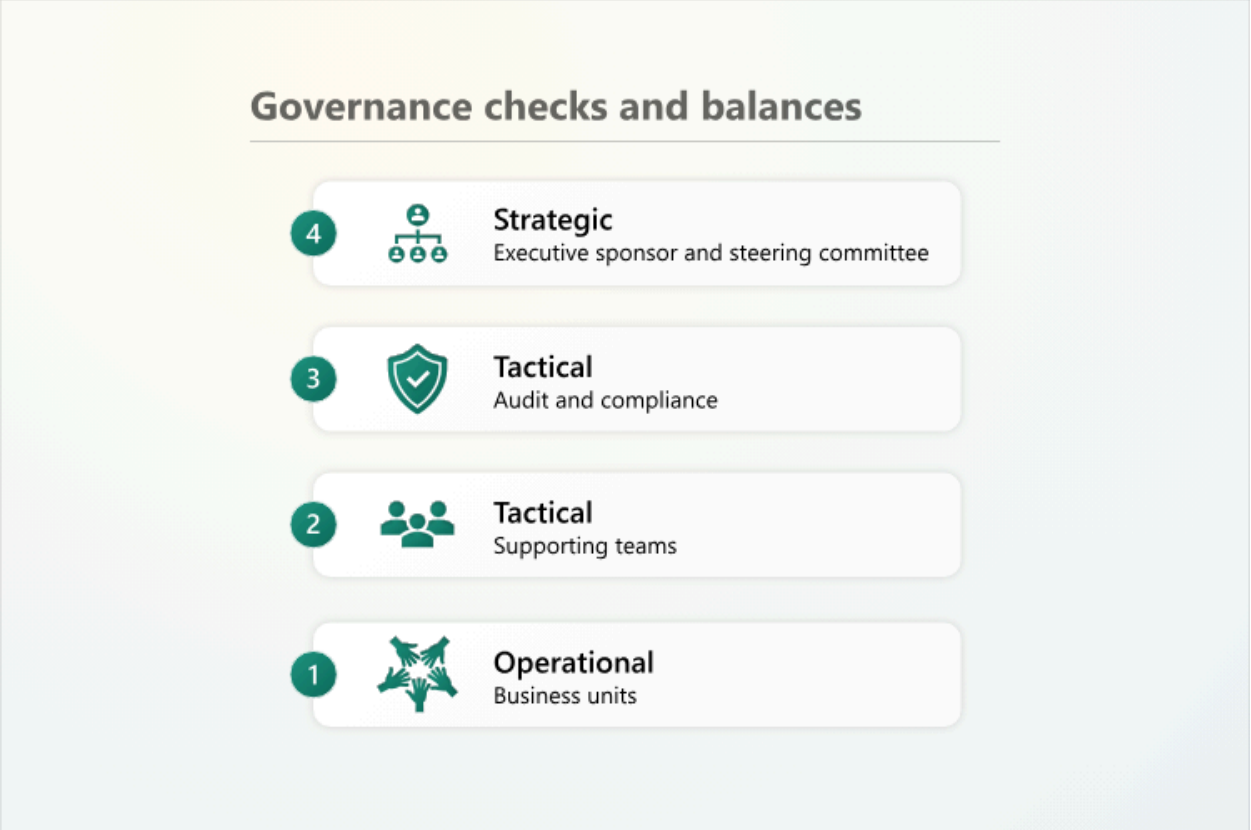
Staffing and accountability

The organizational structure for data governance varies substantially between organizations. In larger organizations there might be a data governance office with dedicated staff. Some organizations have a data governance board, council, or steering committee with assigned members coming from different business units. Depending on the extent of the data governance body within the organization, there could be an executive team separate from a functional team of people.

> **Important**
>
> Regardless of how the governance body is structured, it's important that there's a person or group with sufficient influence over data governance decisions. This person should have authority to enforce those decisions across organizational boundaries.

Checks and balances

Governance accountability is about checks and balances.



Starting with the first level, the levels of checks and balances in the above diagram include:

Expand table

Level	Description
1	**Operational - Business units**: Level 1 is the foundation of a well-governed system, which includes users within the business units performing their work. Self-service data and BI creators have a lot of responsibilities related to authoring, publishing, sharing, security, and data quality. Self-service data and BI consumers also have responsibilities for the proper use of data.
2	**Tactical - Supporting teams**: Level 2 includes several groups that support the efforts of the users in the business units. Supporting teams include the COE, enterprise data and BI, the data governance office, as well as other ancillary teams. Ancillary teams can include IT, security, HR, and legal. A change control board is included here as well.
3	**Tactical - Audit and compliance**: Level 3 includes internal audit, risk management, and compliance teams. These teams provide guidance to levels 1 and 2. They also provide enforcement when necessary.
4	**Strategic - Executive sponsor and steering committee**: The highest level includes the executive-level oversight of strategy and priorities. This level handles any escalated issues that couldn't be solved at lower levels. Therefore, it's important to have a leadership team with sufficient authority to be able to make decisions when necessary.

> **Important**
>
> Everyone has a responsibility to adhere to policies for ensuring that organizational data is secure, protected, and well-managed as an organizational asset. Sometimes this is cited as *everyone is a data steward*. To make this a reality, start with the users in the business units (level 1 described above) as the foundation.

Roles and responsibilities

Once you have a sense for your governance strategy, roles and responsibilities should be defined to establish clear expectations.

Governance team structure, roles (including terminology), and responsibilities vary widely among organizations. Very generalized roles are described in the table below. In some cases, the same person could serve multiple roles. For instance, the Chief Data Officer (CDO) could also be the executive sponsor.

Expand table

Role	Description
Chief Data Officer or Chief Analytics Officer	Defines the strategy for use of data as an enterprise asset. Oversees enterprise-wide governance guidelines and policies.
Data governance board	Steering committee with members from each business unit who, as domain owners, are empowered to make enterprise governance decisions. They make decisions on behalf of the business unit *and* in the best interest of the organization. Provides approvals, decisions, priorities, and direction to the enterprise data governance team and working committees.
Data governance team	Creates governance policies, standards, and processes. Provides enterprise-wide oversight and optimization of data integrity, trustworthiness, privacy, and usability. Collaborates with the COE to provide governance education, support, and mentoring to data owners and content creators.
Data governance working committees	Temporary or permanent teams that focus on individual governance topics, such as security or data quality.
Change management board	Coordinates the requirements, processes, approvals, and scheduling for release management processes with the objective of reducing risk and minimizing the impact of changes to critical applications.
Project	Manages individual governance projects and the ongoing data governance

Role	Description
management office	program.
Fabric executive sponsor	Promotes adoption and the successful use of Fabric. Actively ensures that Fabric decisions are consistently aligned with business objectives, guiding principles, and policies across organizational boundaries. For more information, see the Executive sponsorship article.
Center of Excellence	Mentors the community of creators and consumers to promote the effective use of Fabric for decision-making. Provides cross-departmental coordination of Fabric activities to improve practices, increase consistency, and reduce inefficiencies. For more information, see the Center of Excellence article.
Fabric champions	A subset of content creators found within the business units who help advance the adoption of Fabric. They contribute to data culture growth by advocating the use of best practices and actively assisting colleagues. For more information, see the Community of practice article.
Fabric administrators	Day-to-day-system oversight responsibilities to support the internal processes, tools, and people. Handles monitoring, auditing, and management. For more information, see the System oversight article.
Information technology	Provides occasional assistance to Fabric administrators for services related to Fabric, such as Microsoft Entra ID, Microsoft 365, Teams, SharePoint, or OneDrive.
Risk management	Reviews and assesses data sharing and security risks. Defines ethical data policies and standards. Communicates regulatory and legal requirements.
Internal audit	Auditing of compliance with regulatory and internal requirements.
Data steward	Collaborates with governance committee and/or COE to ensure that organizational data has acceptable data quality levels.
All BI creators and consumers	Adheres to policies for ensuring that data is secure, protected, and well-managed as an organizational asset.

Tip

Name a backup for each person in key roles, for example, members of the data governance board. In their absence, the backup person can attend meetings and make time-sensitive decisions when necessary.

Considerations and key actions

Checklist - Considerations and key actions you can take to establish or strengthen your governance initiatives.

- ✓ **Align goals and guiding principles**: Confirm that the high-level goals and guiding principles of the data culture goals are clearly documented and communicated. Ensure that alignment exists for any new governance guidelines or policies.
- ✓ **Understand what's currently happening**: Ensure that you have a deep understanding of how Fabric is currently used for self-service and enterprise data and BI scenarios. Document opportunities for improvement. Also, document strengths and good practices that would be helpful to scale out more broadly.
- ✓ **Prioritize new governance guidelines and policies**: For prioritizing which new guidelines or policies to create, select an important pain point, high priority need, or known risk for a data domain. It should have significant benefit and can be achieved with a feasible level of effort. When you implement your first governance guidelines, choose something users are likely to support because the change is low impact, or because they are sufficiently motivated to make a change.
- ✓ **Create a schedule to review policies**: Determine the cadence for how often data policies are reevaluated. Reassess and adjust when needs change.
- ✓ **Decide how to handle exceptions**: Determine how conflicts, issues, and requests for exceptions to documented policies will be handled.
- ✓ **Understand existing data assets**: Confirm that you understand what critical data assets exist. Create an inventory of ownership and lineage, if necessary. Keep in mind that you can't govern what you don't know about.
- ✓ **Verify executive sponsorship**: Confirm that you have support and sufficient attention from your executive sponsor, as well as from business unit leaders.
- ✓ **Prepare an action plan**: Include the following key items:
  - **Initial priorities**: Select one data domain or business unit at a time.
  - **Timeline**: Work in iterations long enough to accomplish meaningful progress, yet short enough to periodically adjust.
  - **Quick wins**: Focus on tangible, tactical, and incremental progress.
  - **Success metrics**: Create measurable metrics to evaluate progress.

Questions to ask

Use questions like those found below to assess governance.

- At a high level, what's the current governance strategy? To what extent is the purpose and importance of this governance strategy clear to both end users and the central data and BI teams?
- In general, is the current governance strategy effective?
- What are the key regulatory and compliance criteria that the organization (or specific business units) must adhere to? Where's this criteria documented? Is this information readily available to people who work with data and share data items as a part of their role?
- How well does the current governance strategy align to the user's way of working?
- Is a specific role or team responsible for governance in the organization?
- Who has the authority to create and change governance policies?
- Do governance teams use Microsoft Purview or another tool to support governance activities?
- What are the prioritized governance risks, such as risks to security, information protection, and data loss prevention?
- What's the potential business impact of the identified governance risks?
- How frequently is the governance strategy re-evaluated? What metrics are used to evaluate it, and what mechanisms exist for business users to provide feedback?
- What types of user behaviors create risk when users work with data? How are those risks mitigated?
- What sensitivity labels are in place, if any? Are data and BI decision makers aware of sensitivity labels and the benefits to the business?
- What data loss prevention policies are in place, if any?
- How is "Export to Excel" handled? What steps are taken to prevent data loss prevention? What's the prevalence of "Export to Excel"? What do people do with data once they have it in Excel?
- Are there practices or solutions that are out of regulatory compliance that must be urgently addressed? Are these examples justified with an explanation of the potential business impact, should they not be addressed?

Tip

"Export to Excel" is typically a controversial topic. Often, business users focus on the requirement to have "Export to Excel" possible in BI solutions. Enabling "Export to

Excel" can be counter-productive because a business objective isn't to get data into Excel. Instead, define why end users need the data in Excel. Ask what they do with the data once it's in Excel, which business questions they try to answer, what decisions they make, and what actions they take with the data.

Focusing on business decisions and actions helps steer focus away from tools and features and toward helping people achieve their business objectives.

Maturity levels

The following maturity levels will help you assess the current state of your governance initiatives.

Expand table

Level	State of governance
100: Initial	• Due to a lack of governance planning, the good data management and informal governance practices that are occurring are overly reliant on judgment and experience level of individuals. • There's a significant reliance on undocumented tribal knowledge.
200: Repeatable	• Some areas of the organization have made a purposeful effort to standardize, improve, and document their data management and governance practices. • An initial governance approach exists. Incremental progress is being made.
300: Defined	• A complete governance strategy with focus, objectives, and priorities is enacted and broadly communicated. • Specific governance guidelines and policies are implemented for the top few priorities (pain points or opportunities). They're actively and consistently followed by users. • Roles and responsibilities are clearly defined and documented.
400: Capable	• All Fabric governance priorities align with organizational goals and business objectives. Goals are reassessed regularly. • Processes exist to customize policies for decentralized business units, or to

Level	State of governance
	handle valid exceptions to standard governance policies. • It's clear where Fabric fits into the overall data and BI strategy for the organization. • Fabric activity log and API data is actively analyzed to monitor and audit Fabric activities. Proactive action is taken based on the data.
500: Efficient	• Regular reviews of KPIs or OKRs evaluate measurable governance goals. Iterative, continual progress is a priority. • Agility and implementing continual improvements from lessons learned (including scaling out methods that work) are top priorities for the COE. • Fabric activity log and API data is actively used to inform and improve adoption and governance efforts.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about mentoring and user enablement.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Mentoring and user enablement

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

A critical objective for adoption efforts is to enable users to accomplish as much as they can within the requisite guardrails established by governance guidelines and policies. For this reason, the act of mentoring users is one of the most important responsibilities of the Center of Excellence (COE), and it has a direct influence on how user adoption occurs. For more information about user adoption, see Microsoft Fabric adoption maturity levels.

Skills mentoring

Mentoring and helping users in the Fabric community become more effective can take on various forms, such as:

- Office hours
- Co-development projects
- Best practices reviews
- Extended support

Office hours

Office hours are a form of ongoing community engagements managed by the COE. As the name implies, office hours are times of regularly scheduled availability where members of the community can engage with experts from the COE to receive assistance with minimal process overhead. Office hours are usually group-based, so Fabric champions and other members of the community can also help solve an issue if a topic is in their area of expertise.

Office hours are a very popular and productive activity in many organizations. Some organizations call them *drop-in hours* or even a fun name such as *Power Hour* or *Fabric Fridays*. The primary goal is usually to get questions answered, solve problems, and

remove blockers. Office hours can also be used as a platform for the user community to share ideas, suggestions, and even complaints.

The COE publishes the times for regular office hours when one or more COE members are available. Ideally, office hours are held on a regular and frequent basis. For instance, it could be every Tuesday and Thursday. Consider offering different time slots or rotating times if you have a global workforce.

> **Tip**
>
> One option is to set specific office hours each week. However, users might not show up, so that can end up being inefficient. Alternatively, consider leveraging Microsoft Bookings to schedule office hours. It shows the blocks of time when each COE expert is available, with Outlook integration ensuring availability is up to date.

Office hours are an excellent user enablement approach because:

- Content creators and the COE actively collaborate to answer questions and solve problems together.
- Real work is accomplished while learning and problem solving.
- Others might observe, learn, and participate.
- Individual groups can head to a breakout room to solve a specific problem.

Office hours benefit the COE as well because:

- They're a great way for the COE to identify champions or users with specific skills that the COE didn't previously know about.
- The COE can learn what users throughout the organization are struggling with. It helps inform whether additional resources, documentation, or training might be required.

> **Tip**
>
> It's common for some tough issues to come up during office hours that cannot be solved quickly, such as getting a complex DAX calculation to work, or addressing performance challenges in a complex solution. Set clear expectations for what's in scope for office hours, and if there's any commitment for follow up.

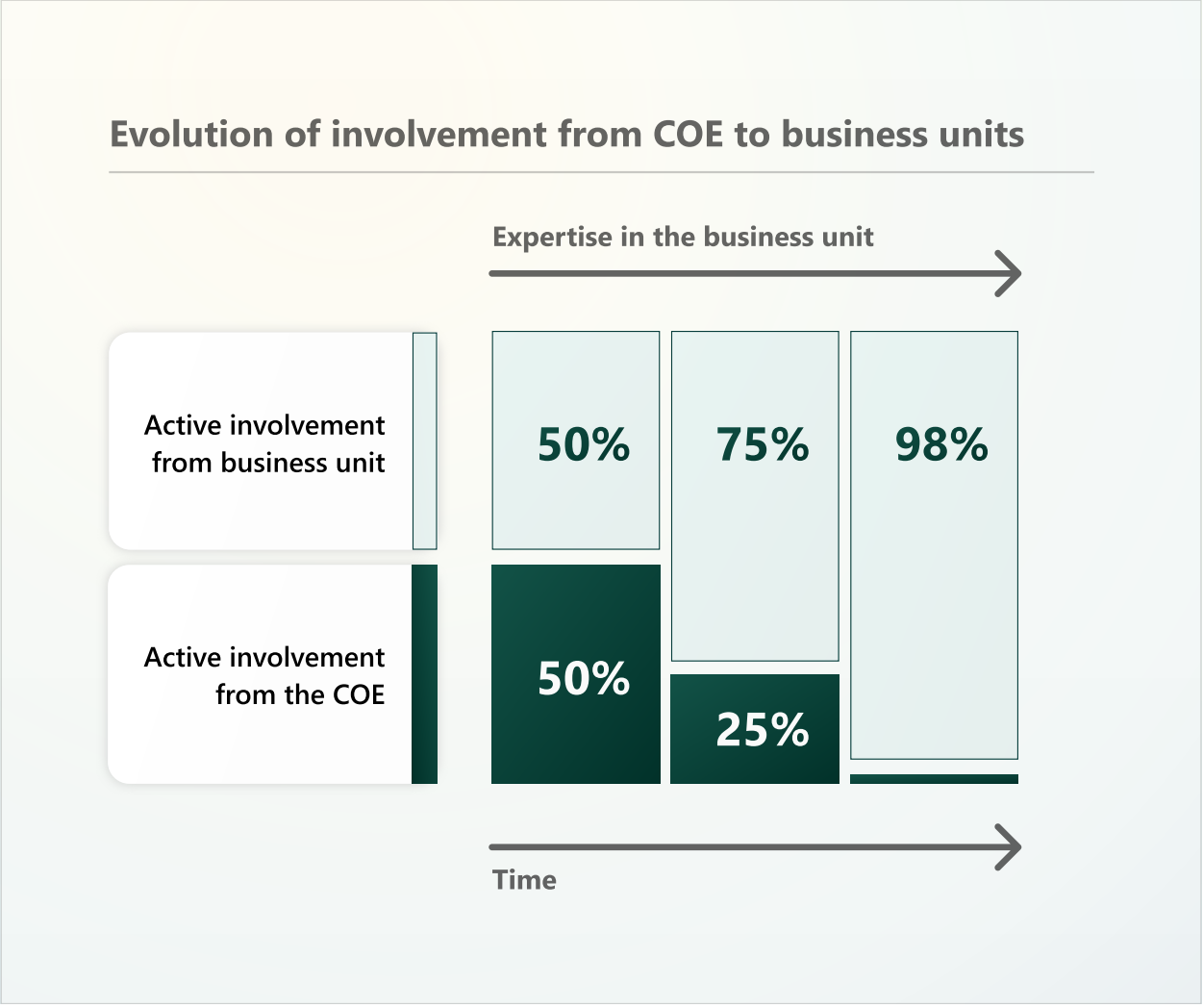
Co-development projects

One way the COE can provide mentoring services is during a *co-development project*. A co-development project is a form of assistance offered by the COE where a user or business unit takes advantage of the technical expertise of the COE to solve business problems with data. Co-development involves stakeholders from the business unit and the COE working in partnership to build a high-quality self-service analytics or business intelligence (BI) solution that the business stakeholders couldn't deliver independently.

The goal of co-development is to help the business unit develop expertise over time while also delivering value. For example, the sales team has a pressing need to develop a new set of commission reports, but the sales team doesn't yet have the knowledge to complete it on their own.

A co-development project forms a partnership between the business unit and the COE. In this arrangement, the business unit is fully invested, deeply involved, and assumes ownership of the project.

Time involvement from the COE reduces over time until the business unit gains expertise and becomes self-reliant.



The active involvement shown in the above diagram changes over time, as follows:

- **Business unit**: 50% initially, up to 75%, finally at 98%-100%.
- **COE**: 50% initially, down to 25%, finally at 0%-2%.

Ideally, the period for the gradual reduction in involvement is identified up-front in the project. This way, both the business unit and the COE can sufficiently plan the timeline and staffing.

Co-development projects can deliver significant short- and long-term benefits. In the short term, the involvement from the COE can often result in a better-designed and better-performing solution that follows best practices and aligns with organizational standards. In the long term, co-development helps increase the knowledge and capabilities of the business stakeholder, making them more self-sufficient, and more confident to deliver quality self-service data and BI solutions in the future.

> **Important**
>
> Essentially, a co-development project helps less experienced users learn the right way to do things. It reduces the risk that refactoring might be needed later, and it increases the ability for a solution to scale and grow over time.

Best practices reviews

The COE could also offer *best practices reviews*. A best practices review can be extremely helpful for content creators who would like to validate their work. They might also be known as *advisory services*, *internal consulting time*, or *technical reviews*. Unlike a co-development project (described previously), a best practices review occurs after the solution has been developed.

During a review, an expert from the COE evaluates self-service Fabric content developed by a member of the community and identifies areas of risk or opportunities for improvement.

Here are some examples of when a best practices review could be beneficial.

- The sales team has a Power BI app that they intend to distribute to thousands of users throughout the organization. Since the app represents high priority content distributed to a large audience, they'd like to have it certified. The standard process to certify content includes a best practices review.
- The finance team would like to assign a workspace to a capacity. A review of the workspace content is required to ensure sound development practices are followed. This type of review is common when the capacity is shared among

multiple business units. (A review might not be required when the capacity is assigned to only one business unit.)
- The operations team is creating a new Fabric solution they expect to be widely used. They would like to request a best practices review before it goes into user acceptance testing (UAT), or before a request is submitted to the change management board.

A best practices review is most often focused on the semantic model design, though the review can encompass all types of data items (such as a lakehouse, data warehouse, data pipeline, dataflow, or semantic model). The review can also encompass reporting items (such as reports, dashboards, or metrics).

Before content is deployed, a best practices review can be used to verify other design decisions, like:

- Code in notebooks follows organizational standards and best practices.
- The appropriate data preparation approach (dataflows, pipelines, notebooks, and others) are used where needed.
- Data sources used are appropriate and query folding is invoked whenever possible where Power Query and dataflows are used.
- Data preparation steps are clean, orderly, and efficient.
- Connectivity mode and storage mode choices (for example, Direct Lake, import, live connection, DirectQuery, and composite model frameworks) are appropriate.
- Location for data sources, like flat files, and original Power BI Desktop files are suitable (preferably stored in a backed-up location with versioning and appropriate security, such as Teams files or a SharePoint shared library).
- Semantic models are well-designed, clean, and understandable, and use a star schema design.
- Model relationships are configured correctly.
- DAX calculations use efficient coding practices (particularly if the data model is large).
- The semantic model size is within a reasonable limit and data reduction techniques are applied.
- Row-level security (RLS) appropriately enforces data permissions.
- Data is accurate and has been validated against the authoritative source(s).
- Approved common definitions and terminology are used.
- Good data visualization practices are followed, including designing for accessibility.

Once the content has been deployed, the best practices review isn't necessarily complete yet. Completing the remainder of the review could also include items such as:

- The target workspace is suitable for the content.
- Workspace security roles are appropriate for the content.
- Other permissions (such as app audience permissions, Build permission, or use of the individual item sharing feature) are correctly and appropriately configured.
- Contacts are identified, and correctly correlate to the owners of the content.
- Sensitivity labels are correctly assigned.
- Fabric item endorsement (certified or promoted) is appropriate.
- Data refresh is configured correctly, failure notifications include the proper users, and uses the appropriate data gateway in standard mode (if applicable).
- All appropriate semantic model best practices rules are followed and, preferably, are automated via a community tool called Best Practices Analyzer for maximum efficiency and productivity.

Extended support

From time to time, the COE might get involved with complex issues escalated from the help desk. For more information, see the User support article.

> **Note**
>
> Offering mentoring services might be a culture shift for your organization. Your reaction might be that users don't usually ask for help with a tool like Excel, so why would they with Power BI? The answer lies in the fact that Power BI and Fabric are extraordinarily powerful tools. They provide data preparation and data modeling capabilities in addition to data visualization. Having the ability to aid and enable users can significantly improve their skills and increase the quality of their solutions—it reduces risks too.

Centralized portal

A single centralized portal, or hub, is where the user community can find:

- Access to the community Q&A forum.
- Announcements of interest to the community, such as new features and release plan updates.
- Schedules and registration links for office hours, lunch and learns, training sessions, and user group meetings.
- Announcements of key changes to content and change log (if appropriate).
- How to request help or support.
- Training materials.

- Documentation, onboarding materials, and frequently asked questions (FAQ).
- Governance guidance and approaches recommended by the COE.
- Report templates.
- Examples of best practices solutions.
- Recordings of knowledge sharing sessions.
- Entry points for accessing managed processes, such as license acquisition, access requests, and gateway configuration.

> **Tip**
>
> In general, only 10%-20% of your community will go out of their way to actively seek out training and educational information. These types of users might naturally evolve to become your **champions**. Everyone else is usually just trying to get the job done as quickly as possible, because their time, focus, and energy are needed elsewhere. Therefore, it's crucial to make information easy for your community users to find.

The goal is to consistently direct users in the community to the centralized portal to find information. The corresponding obligation for the COE is to ensure that the information users need is available in the centralized portal. Keeping the portal updated requires discipline when everyone is busy.

In larger organizations, it can be difficult to implement one single centralized portal. When it's not practical to consolidate into a single portal, a centralized hub can serve as an aggregator, which contains links to the other locations.

> **Important**
>
> Although saving time finding information is important, the goal of a centralized portal is more than that. It's about making information readily available to help your user community do the right thing. They should be able to find information during their normal course of work, with as little friction as possible. Until it's easier to complete a task within the guardrails established by the COE and data governance team, some users will continue to complete their tasks by circumventing policies that are put in place. The recommended path must become the path of least resistance. Having a centralized portal can help achieve this goal.

It takes time for community users to think of the centralized portal as their natural first stop for finding information. It takes consistent redirection to the portal to change habits. Sending someone a link to an original document location in the portal builds

better habits than, for instance, including the answer in an email response. It's the same challenge described in the User support article.

Training

A key factor for successfully enabling self-service users in a Fabric community is training. It's important that the right training resources are readily available and easily discoverable. While some users are so enthusiastic about analytics that they'll find information and figure things out on their own, it isn't true for most of the user community.

Making sure your self-service users (particularly content creators and owners) have access to the training resources they need to be successful doesn't mean that you need to develop your own training content. Developing training content is often counterproductive due to the rapidly evolving nature of the product. Fortunately, an abundance of training resources is available in the worldwide community. A curated set of links goes a long way to help users organize and focus their training efforts, especially for tool training, which focuses on the technology. All external links should be validated by the COE for accuracy and credibility. It's a key opportunity for the COE to add value because COE stakeholders are in an ideal position to understand the learning needs of the community, and to identify and locate trusted sources of quality learning materials.

You'll find the greatest return on investment with creating custom training materials for *organizational-specific* processes, while relying on content produced by others for everything else. It's also useful to have a short training class that focuses primarily on topics like how to find documentation, getting help, and interacting with the community.

> **Tip**
>
> One of the goals of training is to help users learn new skills while helping them avoid bad habits. It can be a balancing act. For instance, you don't want to overwhelm new users by adding in a lot of complexity and friction to a beginner-level class for report creators. However, it's a great investment to make newer content creators aware of things that could otherwise take them a while to figure out. An ideal example is teaching the ability to use a **live connection** to report from an existing semantic model. By teaching this concept at the earliest logical time, you can save a less experienced creator thinking they always need one semantic model for every report (and encourage the good habit of reusing existing semantic models across reports).

Some larger organizations experience continual employee transfers and turnover. Such frequent change results in an increased need for a repeatable set of training resources.

Training resources and approaches

There are many training approaches because people learn in different ways. If you can monitor and measure usage of your training materials, you'll learn over time what works best.

Some training might be delivered more formally, such as classroom training with hands-on labs. Other types of training are less formal, such as:

- Lunch and learn presentations
- Short how-to videos targeted to a specific goal
- Curated set of online resources
- Internal user group presentations
- One-hour, one-week, or one-month challenges
- Hackathon-style events

The advantages of encouraging knowledge sharing among colleagues are described in the Community of practice article.

> **Tip**
>
> Whenever practical, learning should be correlated with building something meaningful and realistic. However, simple demo data does have value during a training course. It allows a learner to focus on how to use the technology rather than the data itself. After completion of introductory session(s), consider offering a *bring your own data* type of session. These types of sessions encourage the learner to apply their new technical skills to an actual business problem. Try to include multiple facilitators from the COE during this type of follow-up session so questions can be answered quickly.

The types of users you might target for training include:

- Content owners, subject matter experts (SMEs), and workspace administrators
- Data creators (for example, users who create semantic models for report creators to use, or who create dataflows, lakehouses, or warehouses for other semantic model creators to use)
- Report creators
- Content consumers and viewers
- Satellite COE members and the champions network

- Fabric administrators

> **Important**
>
> Each type of user represents a different audience that has different training needs. The COE will need to identify how best to meet the needs of each audience. For instance, one audience might find a standard introductory Power BI Desktop class overwhelming, whereas another will want more challenging information with depth and detail for end-to-end solutions that include multiple Fabric workloads. If you have a diverse population of Fabric content creators, consider creating personas and tailoring the experience to an extent that's practical.

The completion of training can be a leading indicator for success with user adoption. Some organizations add an element of fun by granting badges, like *blue belt* or *black belt*, as users progress through the training programs.

Give some consideration to how you want to handle users at various stages of user adoption. Training needs are very different for:

- Onboarding new users (sometimes referred to as *training day zero*).
- Users with minimal experience.
- More experienced users.

How the COE invests its time in creating and curating training materials will change over time as adoption and maturity grows. You might also find over time that some community champions want to run their own tailored set of training classes within their functional business unit.

Sources for trusted Fabric training content

A curated set of online resources is valuable to help community members focus and direct their efforts on what's important. Some publicly available training resources you might find helpful include:

- Microsoft Learn training for Power BI
- Microsoft Learn training for Fabric
- Power BI courses and "in a day" training materials
- LinkedIn Learning for Power BI
- LinkedIn Learning for Fabric

Consider using Microsoft Viva Learning, which is integrated into Microsoft Teams. It includes content from sources such as Microsoft Learn and LinkedIn Learning. Custom

content produced by your organization can be included as well.

In addition to Microsoft content and custom content produced by your organization, you might choose to provide your user community with a curated set of recommended links to trusted online sources. There's a wide array of videos, blogs, and articles produced by the worldwide community. The community comprises Fabric and Power BI experts, Microsoft Most Valued Professions (MVPs), and enthusiasts. Providing a curated learning path that contains specific, reputable, current, and high-quality resources will provide the most value to your user community.

If you do make the investment to create custom in-house training, consider creating short, targeted content that focuses on solving one specific problem. It makes the training easier to find and consume. It's also easier to maintain and update over time.

> **Tip**
>
> The *Help and Support* menu in the Fabric portal is customizable. When your centralized location for training documentation is operational, update the **tenant setting in the Admin portal** with the link. The link can then be accessed from menu when users select the *Get Help* option. Also, be sure to teach users about the *Help* ribbon tab in Power BI Desktop. It includes links to guided learning, training videos, documentation, and more.

Documentation

Concise, well-written documentation can be a significant help for users trying to get things done. Your needs for documentation, and how it's delivered, will depend on how Fabric is managed in your organization. For more information, see the Content ownership and management article.

Certain aspects of Fabric tend to be managed by a centralized team, such as the COE. The following types of documentation are helpful in these situations:

- How to request a Power BI license (and whether there are requirements for manager approval)
- How to request a new capacity
- How to request a new workspace
- How to request a workspace be added to an existing capacity
- How to request access to a gateway data source
- How to request software installation

> **Tip**
>
> For certain activities that are repeated over and over, consider automating them using Power Apps and Power Automate. In this case, your documentation will also include how to access and use the Power Platform functionality.

Different aspects of your documentation can be managed by self-service users, decentralized teams, or by a centralized team. The following types of documentation might differ based on who owns and manages the content:

- How to request a new report
- How to request a report enhancement
- How to request access to data
- How to request new data be prepared and made available for use
- How to request an enhancement to existing data or visualizations

> **Tip**
>
> When planning for a centralized portal, as described earlier in this article, plan how to handle situations when guidance or governance policies need to be customized for one or more business units.

There are also going to be some governance decisions that have been made and should be documented, such as:

- How to request content be certified
- What are the approved file storage locations
- What are the data retention and purge requirements
- What are the requirements for handling sensitive data and personally identifiable information (PII)

Documentation should be located in your centralized portal, which is a searchable location where, preferably, users already work. Either Teams or SharePoint work very well. Creating documentation in either wiki pages or in documents can work equally well, provided that the content is organized well and is easy to find. Shorter documents that focus on one topic are usually easier to consume than long, comprehensive documents.

> **Important**

One of the most helpful pieces of documentation you can publish for the community is a description of the **tenant settings**, and the group memberships required for each tenant setting. Users read about features and functionality online, and sometimes find that it doesn't work for them. When they are able to quickly look up your organization's tenant settings, it can save them from becoming frustrated and attempting workarounds. Effective documentation can reduce the number of help desk tickets that are submitted. It can also reduce the number of people who need to be assigned the Fabric administrator role (who might have this role solely for the purpose of viewing settings).

Over time, you might choose to allow certain types of documentation to be maintained by the community if you have willing volunteers. In this case, you might want to introduce an approval process for changes.

When you see questions repeatedly arise in the Q&A forum (as described in the User support article), during office hours, or during lunch and learns, it's a great indicator that creating new documentation might be appropriate. When the documentation exists, it allows colleagues to reference it when needed. Documentation contributes to user enablement and a self-sustaining community.

Tip

When creating custom documentation or training materials, reference existing Microsoft sites using links whenever possible. Most community bloggers don't keep blog posts or videos up to date.

Power BI template files

A Power BI template is a .pbit file. It can be provided as a starting point for content creators. It's the same as a .pbix file, which can contain queries, a data model, and a report, but with one exception: the template file doesn't contain any data. Therefore, it's a smaller file that can be shared with content creators and owners, and it doesn't present a risk of inappropriately sharing data.

Providing Power BI template files for your community is a great way to:

- Promote consistency.
- Reduce learning curve.
- Show good examples and best practices.
- Increase efficiency.

Power BI template files can improve efficiency and help people learn during the normal course of their work. A few ways that template files are helpful include:

- Reports can use examples of good visualization practices
- Reports can incorporate organizational branding and design standards
- Semantic models can include the structure for commonly used tables, like a date table
- Helpful DAX calculations can be included, like a year-over-year (YoY) calculation
- Common parameters can be included, like a data source connection string
- An example of report and/or semantic model documentation can be included

> ⓘ **Note**
>
> Providing templates not only saves your content creators time, it also helps them move quickly beyond a blank page in an empty solution.

Power BI project files

A Power BI project is a .pbip file. Like a template file (previously described), a project file doesn't contain any data. It's a file format that advanced content creators can use for advanced data model and report management scenarios. For example, you can use project files to save time in development by sharing common model patterns, like date tables, DAX measure expressions, or calculation groups.

You can use Power BI project files with Power BI Desktop developer mode for:

- Advanced editing and authoring (for example, in a code editor such as Visual Studio Code).
- Purposeful separation of semantic model and report items (unlike the .pbix or .pbit files).
- Enabling multiple content creators and developers to work on the same project concurrently.
- Integrating with source control (such as by using Fabric Git integration).
- Using continuous integration and continuous delivery (CI/CD) techniques to automate integration, testing and deployment of changes, or versions of content.

> ⓘ **Note**
>
> Power BI includes capabilities such as .pbit template files and .pbip project files that make it simple to share starter resources with authors. Other Fabric workloads provide different approaches to content development and sharing. Having a set of

starter resources is important regardless of the items being shared. For example, your portal might include a set of SQL scripts or notebooks that present tested approaches to solve common problems.

Considerations and key actions

Checklist - Considerations and key actions you can take to establish, or improve, mentoring and user enablement.

- ✓ **Consider what mentoring services the COE can support**: Decide what types of mentoring services the COE is capable of offering. Types can include office hours, co-development projects, and best practices reviews.
- ✓ **Communicate regularly about mentoring services**: Decide how you will communicate and advertise mentoring services, such as office hours, to the user community.
- ✓ **Establish a regular schedule for office hours**: Ideally, hold office hours at least once per week (depending on demand from users as well as staffing and scheduling constraints).
- ✓ **Decide what the expectations will be for office hours**: Determine what the scope of allowed topics or types of issues users can bring to office hours. Also, determine how the queue of office hours requests will work, whether any information should be submitted ahead of time, and whether any follow up afterwards can be expected.
- ✓ **Create a centralized portal**: Ensure that you have a well-supported centralized hub where users can easily find training materials, documentation, and resources. The centralized portal should also provide links to other community resources such as the Q&A forum and how to find help.
- ✓ **Create documentation and resources**: In the centralized portal, create, compile, and publish useful documentation. Identify and promote the top 3-5 resources that will be most useful to the user community.
- ✓ **Update documentation and resources regularly**: Ensure that content is reviewed and updated on a regular basis. The objective is to ensure that the information available in the portal is current and reliable.
- ✓ **Compile a curated list of reputable training resources**: Identify training resources that target the training needs and interests of your user community. Post the list in the centralized portal and create a schedule to review and validate the list.

- ✓ **Consider whether custom in-house training will be useful**: Identify whether custom training courses, developed in-house, will be useful and worth the time investment. Invest in creating content that's specific to the organization.
- ✓ **Provide templates and projects**: Determine how you'll use templates including Power BI template files and Power BI project files. Include the resources in your centralized portal, and in training materials.
- ✓ **Create goals and metrics**: Determine how you'll measure effectiveness of the mentoring program. Create KPIs (key performance indicators) or OKRs (objectives and key results) to validate that the COE's mentoring efforts strengthen the community and its ability to provide self-service BI.

Questions to ask

Use questions like those found below to assess mentoring and user enablement.

- Is there an effective process in place for users to request training?
- Is there a process in place to evaluate user skill levels (such as beginner, intermediate, or advanced)? Can users study for and achieve Microsoft certifications by using company resources?
- What's the onboarding process to introduce new people in the user community to data and BI solutions, tools, and processes?
- Have all users followed the appropriate Microsoft Learn learning paths for their roles during onboarding?
- What kinds of challenges do users experience due to lack of training or mentorship?
- What impact does lack of enablement have on the business?
- When users exhibit behavior that creates governance risks, are they punished or do they undergo education and mentorship?
- What training materials are in place to educate people about governance processes and policies?
- Where's the central documentation maintained? Who maintains it?
- Do central resources exist, like organizational design guidelines, themes, or template files?

Maturity levels

The following maturity levels will help you assess the current state of your mentoring and user enablement.

Expand table

Level	State of mentoring and user enablement
100: Initial	• Some documentation and resources exist. However, they're siloed and inconsistent. • Few users are aware of, or take advantage of, available resources.
200: Repeatable	• A centralized portal exists with a library of helpful documentation and resources. • A curated list of training links and resources are available in the centralized portal. • Office hours are available so the user community can get assistance from the COE.
300: Defined	• The centralized portal is the primary hub for community members to locate training, documentation, and resources. The resources are commonly referenced by champions and community members when supporting and learning from each other. • The COE's skills mentoring program is in place to assist users in the community in various ways.
400: Capable	• Office hours have regular and active participation from all business units in the organization. • Best practices reviews from the COE are regularly requested by business units. • Co-development projects are repeatedly executed with success by the COE and members of business units.
500: Efficient	• Training, documentation, and resources are continually updated and improved by the COE to ensure the community has current and reliable information. • Measurable and tangible business value is gained from the mentoring program by using KPIs or OKRs.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about the community of practice.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

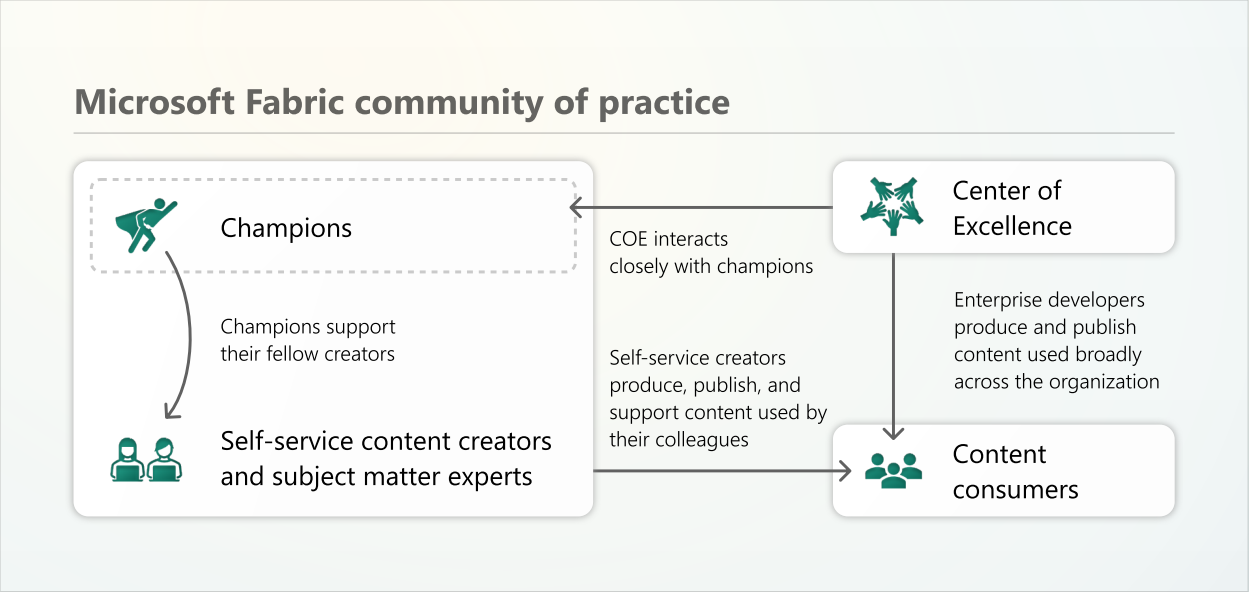
Microsoft Fabric adoption roadmap: Community of practice

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

A *community of practice* is a group of people with a common interest that interacts with, and helps, each other on a voluntary basis. Using a tool such as Microsoft Fabric to produce effective analytics is a common interest that can bring people together across an organization.

The following diagram provides an overview of an internal community.



The above diagram shows the following:

- The **community of practice** includes everyone with an interest in Fabric.
- The **Center of Excellence (COE)** forms the nucleus of the community. The COE oversees the entire community and interacts most closely with its champions.
- **Self-service content creators and subject matter experts (SMEs)** produce, publish, and support content that's used by their colleagues, who are consumers.
- **Content consumers** view content produced by both self-service creators and enterprise business intelligence (BI) developers.
- **Champions** are a subset of the self-service content creators. Champions are in an excellent position to support their fellow content creators to generate effective

analytics solutions.

Champions are the smallest group among creators and SMEs. Self-service content creators and SMEs represent a larger number of people. Content consumers represent the largest number of people in most organizations.

> **Note**
>
> All references to the Fabric community in this adoption series of articles refer to internal users, unless explicitly stated otherwise. There's an active and vibrant worldwide community of bloggers and presenters who produce a wealth of knowledge about Fabric. However, internal users are the focus of this article.

For information about related topics including resources, documentation, and training provided for the Fabric community, see the Mentoring and user enablement article.

Champions network

One important part of a community of practice is its *champions*. A champion is a self-service content creator who works in a business unit that engages with the COE. A champion is recognized by their peers as the *go-to* expert. A champion continually builds and shares their knowledge even if it's not an official part of their job role. Champions influence and help their colleagues in many ways including solution development, learning, skills improvement, troubleshooting, and keeping up to date.

Champions emerge as leaders of the community of practice who:

- Have a deep interest in analytics being used effectively and adopted successfully throughout the organization.
- Possess strong technical skills as well as domain knowledge for their functional business unit.
- Have an inherent interest in getting involved and helping others.
- Are early adopters who are enthusiastic about experimenting and learning.
- Can effectively translate business needs into solutions.
- Communicate well with colleagues.

> **Tip**
>
> To add an element of fun, some organizations refer to their champions network as *ambassadors*, *Jedis*, *ninjas*, or *rangers*. Microsoft has an internal community called *BI Champs*.

Often, people aren't directly asked to become champions. Commonly, champions are identified by the COE and recognized for the activities they're already doing, such as frequently answering questions on an internal discussion channel or participating in lunch and learn sessions.

Different approaches will be more effective for different organizations, and each organization will find what works best for them as their maturity level increases.

> **Important**
>
> Someone very well might be acting in the role of a champion without even knowing it, and without any formal recognition. The COE should always be on the lookout for champions. COE members should actively monitor the discussion channel to see who is particularly helpful. The COE should deliberately encourage and support potential champions, and when appropriate, invite them into a champions network to make the recognition formal.

Knowledge sharing

The overriding objective of a community of practice is to facilitate knowledge sharing among colleagues and across organizational boundaries. There are many ways knowledge sharing occurs. It could be during the normal course of work. Or, it could be during a more structured activity, such as:

Expand table

Activity	Description
Discussion channel	A Q&A forum where anyone in the community can post and view messages. Often used for help and announcements. For more information, see the User support article.
Lunch and learn sessions	Regularly scheduled sessions where someone presents a short session about something they've learned or a solution they've created. The goal is to get a variety of presenters involved, because it's a powerful message to hear firsthand what colleagues have achieved.
Office hours with the COE	Regularly scheduled times when COE experts are available so the community can engage with them. Community users can receive assistance with minimal process overhead. For more information, see the Mentoring and user enablement article.
Internal blog posts or wiki posts	Short blog posts, usually covering technical how-to topics.

Activity	Description
Internal analytics user group	A subset of the community that chooses to meet as a group on a regularly scheduled basis. User group members often take turns presenting to each other to share knowledge and improve their presentation skills.
Book club	A subset of the community select a book to read on a schedule. They discuss what they've learned and share their thoughts with each other.
Internal analytics conferences or events	An annual or semi-annual internal conference that delivers a series of sessions focused on the needs of self-service content creators, subject matter experts, and stakeholders.

> **Tip**
>
> Inviting an external presenter can reduce the effort level and bring a fresh viewpoint for learning and knowledge sharing.

Incentives

A lot of effort goes into forming and sustaining a successful community. It's advantageous to everyone to empower and reward users who work for the benefit of the community.

Rewarding community members

Incentives that the entire community (including champions) find particularly rewarding can include:

- **Contests with a small gift card or time off**: For example, you might hold a performance tuning event with the winner being the person who successfully reduced the size of their data model the most.
- **Ranking based on help points**: The more frequently someone participates in Q&A, they achieve a change in status on a leaderboard. This type of gamification promotes healthy competition and excitement. By getting involved in more conversations, the participant learns and grows personally in addition to helping their colleagues.
- **Leadership communication**: Reach out to a manager when someone goes above and beyond so that their leader, who might not be active in the community, sees the value that their staff member provides.

Rewarding champions

Different types of incentives will appeal to different types of people. Some community members will be highly motivated by praise and feedback. Some will be inspired by gamification and a bit of fun. Others will highly value the opportunity to improve their level of knowledge.

Incentives that champions find particularly rewarding can include:

- **More direct access to the COE**: The ability to have *connections* in the COE is valuable. It's depicted in the diagram shown earlier in this article.
- **Champion of the month**: Publicly thank one of your champions for something outstanding they did recently. It could be a fun tradition at the beginning of a monthly lunch and learn.
- **A private experts discussion area**: A private area for the champions to share ideas and learn from each other is usually highly valued.
- **Specialized or deep dive information and training**: Access to additional information to help champions grow their skillsets (as well as help their colleagues) will be appreciated. It could include attending advanced training classes or conferences.

Communication plan

Communication with the community occurs through various types of communication channels. Common communication channels include:

- Internal discussion channel or forum.
- Announcements channel.
- Organizational newsletter.

The most critical communication objectives include ensuring your community members know that:

- The COE exists.
- How to get help and support.
- Where to find resources and documentation.
- Where to find governance guidelines.
- How to share suggestions and ideas.

> **Tip**

Consider requiring a simple quiz before a user is granted a Power BI or Fabric license. This *quiz* is a misnomer because it doesn't focus on any technical skills. Rather, it's a short series of questions to verify that the user knows where to find help and resources. It sets them up for success. It's also a great opportunity to have users acknowledge any governance policies or data privacy and protection agreements you need them to be aware of. For more information, see the **System oversight** article.

Types of communication

There are generally four types of communication to plan for:

- **New employee communications** can be directed to new employees (and contractors). It's an excellent opportunity to provide onboarding materials for how to get started. It can include articles on topics like how to get Power BI Desktop installed, how to request a license, and where to find introductory training materials. It can also include general data governance guidelines that all users should be aware of.
- **Onboarding communications** can be directed to employees who are just acquiring a license or are getting involved with the community of practice. It presents an excellent opportunity to provide the same materials as given to new employee communications (as mentioned above).
- **Ongoing communications** can include regular announcements and updates directed to all users, or subsets of users, like:
  - Announcements about changes that are planned to key organizational content. For example, changes are to be published for a critical shared semantic model that's used heavily throughout the organization. It can also include the announcement of new features. For more information about planning for change, see the Tenant-level monitoring article.
  - Feature announcements, which are more likely to receive attention from the reader if the message includes meaningful context about why it's important. (Although an RSS feed can be a helpful technique, with the frequent pace of change, it can become noisy and might be ignored.)
- **Situational communications** can be directed to specific users or groups based on a specific occurrence discovered while monitoring the platform. For example, perhaps you notice a significant amount of sharing from the personal workspace a particular user, so you choose to send them some information about the benefits of workspaces and Power BI apps.

Tip

One-way communication to the user community is important. Don't forget to also include bidirectional communication options to ensure the user community has an opportunity to provide feedback.

Community resources

Resources for the internal community, such as documentation, templates, and training, are critical for adoption success. For more information about resources, see the Mentoring and user enablement article.

Considerations and key actions

Checklist - Considerations and key actions you can take for the community of practice follow.

Initiate, grow, and sustain your champions network:

- ✓ **Clarify goals**: Clarify what your specific goals are for cultivating a champions network. Make sure these goals align with your overall data and BI strategy, and that your executive sponsor is on board.
- ✓ **Create a plan for the champions network**: Although some aspects of a champions network will always be informally led, determine to what extent the COE will purposefully cultivate and support champion efforts throughout individual business units. Consider how many champions is ideal for each functional business area. Usually, 1-2 champions per area works well, but it can vary based on the size of the team, the needs of the self-service community, and how the COE is structured.
- ✓ **Decide on commitment level for champions**: Decide what level of commitment and expected time investment will be required of champions. Be aware that the time investment will vary from person to person, and team to team due to different responsibilities. Plan to clearly communicate expectations to people who are interested in getting involved. Obtain manager approval when appropriate.
- ✓ **Decide how to identify champions**: Determine how you will respond to requests to become a champion, and how the COE will seek out champions. Decide if you will openly encourage interested employees to self-identify as a champion and ask to learn more (less common). Or, whether the COE will observe efforts and extend a private invitation (more common).

- ✓ **Determine how members of the champions network will be managed**: One excellent option for managing who the champions are is with a security group. Consider:
  - How you will communicate with the champions network (for example, in a Teams channel, a Yammer group, and/or an email distribution list).
  - How the champions network will communicate and collaborate with each other directly (across organizational boundaries).
  - Whether a private and exclusive discussion forum for champions and COE members is appropriate.
- ✓ **Plan resources for champions**: Ensure members of the champions network have the resources they need, including:
  - Direct access to COE members.
  - Influence on data policies being implemented (for example, requirements for a semantic model certification policy).
  - Influence on the creation of best practices and guidance (for example, recommendations for accessing a specific source system).
- ✓ **Involve champions**: Actively involve certain champions as satellite members of the COE. For more information about ways to structure the COE, see the Center of Excellence article.
- ✓ **Create a feedback loop for champions**: Ensure that members of the champions network can easily provide information or submit suggestions to the COE.
- ✓ **Routinely provide recognition and incentives for champions**: Not only is praise an effective motivator, but the act of sharing examples of successful efforts can motivate and inspire others.

Improve knowledge sharing:

- ✓ **Identify knowledge sharing activities**: Determine what kind of activities for knowledge sharing fit well into the organizational data culture. Ensure that all planned knowledge sharing activities are supportable and sustainable.
- ✓ **Confirm roles and responsibilities**: Verify who will take responsibility for coordinating all knowledge sharing activities.

Introduce incentives:

- ✓ **Identify incentives for champions**: Consider what type of incentives you could offer to members of your champions network.
- ✓ **Identify incentives for community members**: Consider what type of incentives you could offer to your broader internal community.

Improve communications:

- ✓ **Establish communication methods**: Evaluate which methods of communication fit well in your data culture. Set up different ways to communicate, including history retention and search.
- ✓ **Identify responsibility**: Determine who will be responsible for different types of communication, how, and when.

Questions to ask

Use questions like those found below to assess the community of practice.

- Is there a centralized portal for a community of practice to engage in knowledge sharing?
- Do technical questions and requests for support always go through central teams like the COE or support? Alternatively, to what extent is the community of practice engaging in knowledge sharing?
- Do any incentives exist for people to engage in knowledge sharing or improve their skills with data and BI tools?
- Is there a system of recognition to acknowledge significant self-service efforts in teams?
- Are champions recognized among the user community? If so, what explicit recognition do they get for their expertise? How are they identified?
- If no champions are recognized, are there any potential candidates?
- What role do central teams envision that champions play in community of practice?
- How often do central data and BI teams engage with the user community? What medium do these interactions take? Are they bidirectional discussions or unidirectional communications?
- How are changes and announcements communicated within the community of practice?
- Among the user community, who is the most enthusiastic about analytics and BI tools? Who is the least enthusiastic, or the most negative, and why?

Maturity levels

The following maturity levels will help you assess the current state of your community of practice.

Expand table

Level	State of the community of practice
100: Initial	• Some self-service content creators are doing great work throughout the organization. However, their efforts aren't recognized. • Efforts to purposefully share knowledge across the organizational boundaries are rare and unstructured. • Communication is inconsistent, without a purposeful plan.
200: Repeatable	• The first set of champions are identified. • The goals for a champions network are identified. • Knowledge sharing practices are gaining traction.
300: Defined	• Knowledge sharing in multiple forms is a normal occurrence. Information sharing happens frequently and purposefully. • Goals for transparent communication with the user community are defined.
400: Capable	• Champions are identified for all business units. They actively support colleagues in their self-service efforts. • Incentives to recognize and reward knowledge sharing efforts are a common occurrence. • Regular and frequent communication occurs based on a predefined communication plan.
500: Efficient	• Bidirectional feedback loops exist between the champions network and the COE. • Key performance indicators measure community engagement and satisfaction. • Automation is in place when it adds direct value to the user experience (for example, automatic access to a group that provides community resources).

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about user support.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: User support

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

This article addresses user support. It focuses primarily on the resolution of issues.

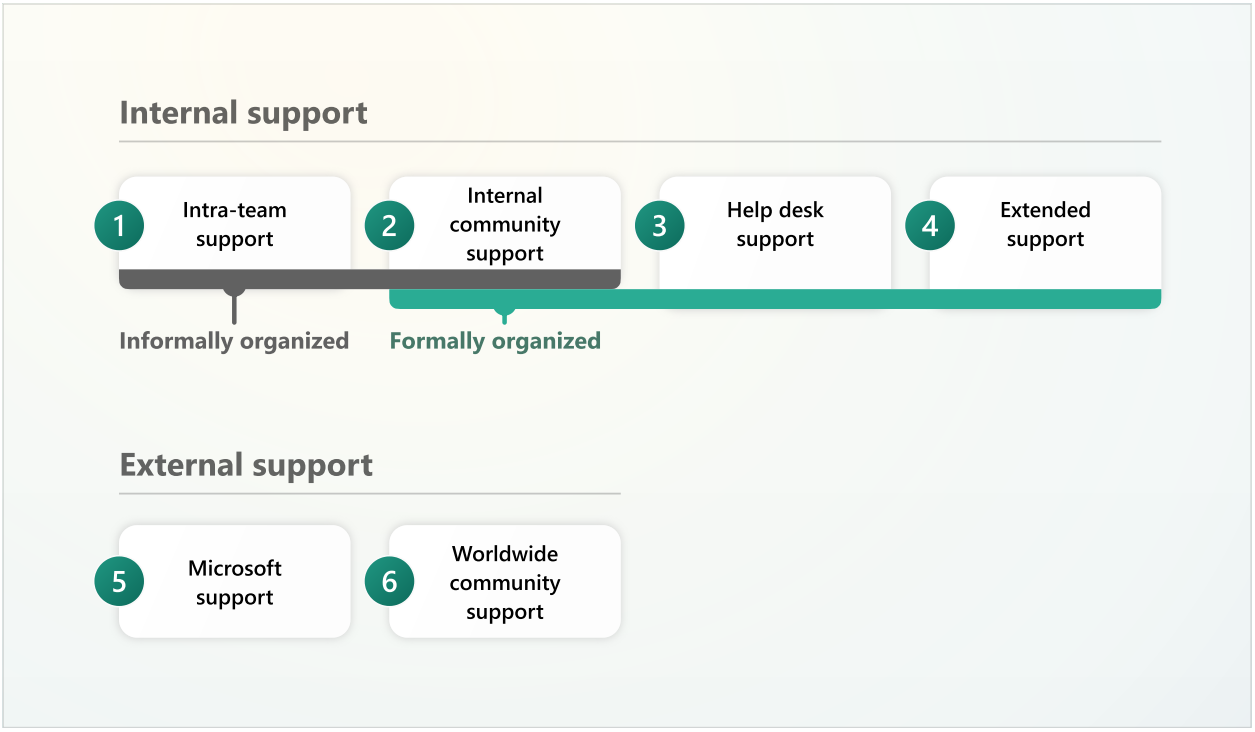
The first sections of this article focus on user support aspects you have control over internally within your organization. The final topics focus on external resources that are available.

For a description of related topics, including skills mentoring, training, documentation, and co-development assistance provided to the internal Fabric user community, see the Mentoring and user enablement article. The effectiveness of those activities can significantly reduce the volume of formal user support requests and increase user experience overall.

Types of user support

If a user has an issue, do they know what their options are to resolve it?

The following diagram shows some common types of user support that organizations employ successfully:



The six types of user support shown in the above diagram include:

Expand table

Type	Description
1	**Intra-team support (internal)** is very informal. Support occurs when team members learn from each other during the natural course of their job.
2	**Internal community support (internal)** can be organized informally, formally, or both. It occurs when colleagues interact with each other via internal community channels.
3	**Help desk support (internal)** handles formal support issues and requests.
4	**Extended support (internal)** involves handling complex issues escalated by the help desk.
5	**Microsoft support (external)** includes support for licensed users and Fabric administrators. It also includes comprehensive documentation.
6	**Community support (external)** includes the worldwide community of experts, Microsoft Most Valued Professionals (MVPs), and enthusiasts who participate in forums and publish content.

In some organizations, intra-team and internal community support are most relevant for self-service data and business intelligence (BI)—content is owned and managed by creators and owners in decentralized business units. Conversely, the help desk and extended support are reserved for technical issues and enterprise data and BI (content is owned and managed by a centralized BI team or Center of Excellence). In some organizations, all four types of support could be relevant for any type of content.

> **Tip**
>
> For more information about business-led self-service, managed self-service, and enterprise data and BI concepts, see the **Content ownership and management** article.

Each of the six types of user support introduced above are described in further detail in this article.

Intra-team support

Intra-team support refers to when team members learn from and help each other during their daily work. Self-service content creators who emerge as your champions tend to take on this type of informal support role voluntarily because they have an intrinsic desire to help. Although it's an informal support mode, it shouldn't be undervalued. Some estimates indicate that a large percentage of learning at work is peer learning, which is particularly helpful for analysts who are creating domain-specific analytics solutions.

> **Note**
>
> Intra-team support does not work well for individuals who are the only data analyst within a department. It's also not effective for those who don't have very many connections yet in their organization. When there aren't any close colleagues to depend on, other types of support, as described in this article, become more important.

Internal community support

Assistance from your fellow community members often takes the form of messages in a discussion channel, or a forum set up specifically for the community of practice. For example, someone posts a message that they're having problems getting a DAX calculation to work or are looking for the right Python module to import. They then receive a response from someone in the organization with suggestions or links.

> **Tip**
>
> The goal of an internal Fabric community is to be self-sustaining, which can lead to reduced formal support demands and costs. It can also facilitate managed self-

service content creation occurring on a broader scale versus a purely centralized approach. However, there will always be a need to monitor, manage, and nurture the internal community. Here are two specific tips:

- Be sure to cultivate multiple experts in the more difficult topics like **T-SQL**, **Python**, **Data Analysis eXpressions (DAX)** and the **Power Query M formula language**. When a community member becomes a recognized expert, they could become overburdened with too many requests for help.
- A greater number of community members might readily answer certain types of questions (for example, report visualizations), whereas a smaller number of members will answer others (for example, complex T-SQL or DAX). It's important for the COE to allow the community a chance to respond yet also be willing to promptly handle unanswered questions. If users repeatedly ask questions and don't receive an answer, it will significantly hinder growth of the community. In this case, a user is likely to leave and never return if they don't receive any responses to their questions.

An internal community discussion channel is commonly set up as a Teams channel or a Yammer group. The technology chosen should reflect where users already work, so that the activities occur within their natural workflow.

One benefit of an internal discussion channel is that responses can come from people that the original requester has never met before. In larger organizations, a community of practice brings people together based on a common interest. It can offer diverse perspectives for getting help and learning in general.

Use of an internal community discussion channel allows the Center of Excellence (COE) to monitor the kind of questions people are asking. It's one way the COE can understand the issues users are experiencing (commonly related to content creation, but it could also be related to consuming content).

Monitoring the discussion channel can also reveal additional analytics experts and potential champions who were previously unknown to the COE.

ⓘ Important

It's a best practice to continually identify emerging champions, and to engage with them to make sure they're equipped to support their colleagues. As described in the **Community of practice** article, the COE should actively monitor the discussion channel to see who is being helpful. The COE should deliberately encourage and

support community members. When appropriate, invite them into the champions network.

Another key benefit of a discussion channel is that it's searchable, which allows other people to discover the information. It is, however, a change of habit for people to ask questions in an open forum rather than private messages or email. Be sensitive to the fact that some individuals aren't comfortable asking questions in such a public way. It openly acknowledges what they don't know, which might be embarrassing. This reluctance might reduce over time by promoting a friendly, encouraging, and helpful discussion channel.

Tip

You might be tempted to create a bot to handle some of the most common, straightforward questions from the community. A bot can work for uncomplicated questions such as "How do I request a license?" or "How do I request a workspace?" Before taking this approach, consider if there are enough routine and predictable questions that would make the user experience better rather than worse. Often, a well-created FAQ (frequently asked questions) works better, and it's faster to develop and easier to maintain.

Help desk support

The help desk is usually operated as a shared service, staffed by the IT department. Users who will likely rely on a more formal support channel include those who are:

- Less experienced users.
- Newer to the organization.
- Reluctant to post a message to the internal discussion community.
- Lacking connections and colleagues within the organization.

There are also certain technical issues that can't be fully resolved without IT involvement, like software installation and upgrade requests when machines are IT-managed.

Busy help desk personnel are usually dedicated to supporting multiple technologies. For this reason, the easiest types of issues to support are those which have a clear resolution and can be documented in a knowledgebase. For instance, software installation prerequisites or requirements to get a license.

Some organizations ask the help desk to handle only very simple break-fix issues. Other organizations have the help desk get involved with anything that is repeatable, like new

workspace requests, managing gateway data sources, or requesting a new capacity.

> **Important**
>
> Your Fabric governance decisions will directly impact the volume of help desk requests. For example, if you choose to limit **workspace creation permissions in the tenant settings**, it will result in users submitting help desk tickets. While it's a legitimate decision to make, you must be prepared to satisfy the request very quickly. Respond to this type of request within 1-4 hours, if possible. If you delay too long, users will use what they already have or find a way to work around your requirements. That might not be the ideal scenario. Promptness is critical for certain help desk requests. Consider that automation by using **Power Apps** and **Power Automate** can help make some processes more efficient. For more information, see **Tenant-level workspace planning**.

Over time, troubleshooting and problem resolution skills become more effective as help desk personnel expand their knowledgebase and experience with supporting Fabric. The best help desk personnel are those who have a good grasp of what users need to accomplish.

> **Tip**
>
> Purely technical issues, for example **data refresh** failure or the need to **add a new user to a gateway data source**, usually involve straightforward responses associated with a service-level agreement (SLA). For instance, there could be an SLA to respond to blocking issues within one hour and resolve them within eight hours. It's generally more difficult to define SLAs for troubleshooting issues, like data discrepancies.

Extended support

Since the COE has deep insight into how Fabric is used throughout the organization, they're a great option for extended support should a complex issue arise. Involving the COE in the support process should be by an escalation path.

Managing requests as purely an escalation path from the help desk gets difficult to enforce since COE members are often well-known to business users. To encourage the habit of going through the proper channels, COE members should redirect users to submit a help desk ticket. It will also improve the data quality for analyzing help desk requests.

Microsoft support

In addition to the internal user support approaches discussed in this article, there are valuable external support options directly available to users and Fabric administrators that shouldn't be overlooked.

Microsoft documentation

Check the Fabric support website for high-priority issues that broadly affect all customers. Global Microsoft 365 administrators have access to additional support issue details within the Microsoft 365 portal.

Refer to the comprehensive Fabric documentation. It's an authoritative resource that can aid troubleshooting and search for information. You can prioritize results from the documentation site. For example, enter a site-targeted search request into your web search engine, like `power bi gateway site:learn.microsoft.com`.

Power BI Pro and Premium Per User end-user support

Licensed users are eligible to log a support ticket with Microsoft.

> **Tip**
>
> Make it clear to your internal user community whether you prefer technical issues to be reported to the internal help desk. If your help desk is equipped to handle the workload, having a centralized internal area collect user issues can provide a superior user experience versus every user trying to resolve issues on their own. Having visibility and analyzing support issues is also helpful for the **COE**.

Administrator support

There are several support options available for Fabric administrators.

For customers who have a Microsoft Unified Support contract, consider granting help desk and COE members access to the Microsoft Services Hub. One advantage of the Microsoft Services Hub is that your help desk and COE members can be set up to submit and view support requests.

Worldwide community support

In addition to the internal user support approaches described in this article, and Microsoft support options described previously, you can leverage the worldwide Fabric community.

The worldwide community is useful when a question can be easily understood by someone without domain knowledge, and when it doesn't involve confidential data or sensitive internal processes.

Publicly available community forums

There are several public community forums where users can post issues and receive responses from any user in the world. Getting answers from anyone, anywhere, can be very powerful and exceedingly helpful. However, as is the case with any public forum, it's important to validate the advice and information posted on the forum. The advice posted on the internet might not be suitable for your situation.

Publicly available discussion areas

It's very common to see people posting Fabric technical questions on social media platforms. You might find discussions, post announcements, and users helping each other.

Community documentation

The Fabric global community is vibrant. Every day, there are a great number of Fabric blog posts, articles, webinars, and videos published. When relying on community information for troubleshooting, watch out for:

- How recent the information is. Try to verify when it was published or last updated.
- Whether the situation and context of the solution found online truly fits your circumstance.
- The credibility of the information being presented. Rely on reputable blogs and sites.

Considerations and key actions

Checklist - Considerations and key actions you can take for user support follow.

Improve your intra-team support:

- ✓ **Provide recognition and encouragement**: Provide incentives to your champions as described in the Community of practice article.
- ✓ **Reward efforts**: Recognize and praise meaningful grassroots efforts when you see them happening.
- ✓ **Create formal roles**: If informal intra-team efforts aren't adequate, consider formalizing the roles you want to enact in this area. Include the expected contributions and responsibilities in the HR job description, when appropriate.

Improve your internal community support:

- ✓ **Continually encourage questions**: Encourage users to ask questions in the designated community discussion channel. As the habit builds over time, it will become normalized to use that as the first option. Over time, it will evolve to become more self-supporting.
- ✓ **Actively monitor the discussion area**: Ensure that the appropriate COE members actively monitor this discussion channel. They can step in if a question remains unanswered, improve upon answers, or make corrections when appropriate. They can also post links to additional information to raise awareness of existing resources. Although the goal of the community is to become self-supporting, it still requires dedicated resources to monitor and nurture it.
- ✓ **Communicate options available**: Make sure your user population knows the internal community support area exists. It could include the prominent display of links. You can include a link in regular communications to your user community. You can also customize the help menu links in the Fabric portal to direct users to your internal resources.
- ✓ **Set up automation**: Ensure that all licensed users automatically have access to the community discussion channel. It's possible to automate license setup by using group-based licensing.

Improve your internal help desk support:

- ✓ **Determine help desk responsibilities**: Decide what the initial scope of Fabric support topics that the help desk will handle.
- ✓ **Assess the readiness level**: Determine whether your help desk is prepared to handle Fabric support. Identify whether there are readiness gaps to be addressed.
- ✓ **Arrange for additional training**: Conduct knowledge transfer sessions or training sessions to prepare the help desk staff.
- ✓ **Update the help desk knowledgebase**: Include known questions and answers in a searchable knowledgebase. Ensure someone is responsible for regular updates to the knowledgebase to reflect new and enhanced features over time.

- ✓ **Set up a ticket tracking system**: Ensure a good system is in place to track requests submitted to the help desk.
- ✓ **Decide whether anyone will be on-call for any issues related to Fabric**: If appropriate, ensure the expectations for 24/7 support are clear.
- ✓ **Determine what SLAs will exist**: When a specific service level agreement (SLA) exists, ensure that expectations for response and resolution are clearly documented and communicated.
- ✓ **Be prepared to act quickly**: Be prepared to address specific common issues extremely quickly. Slow support response will result in users finding workarounds.

Improve your internal COE extended support:

- ✓ **Determine how escalated support will work**: Decide what the escalation path will be for requests the help desk cannot directly handle. Ensure that the COE (or equivalent personnel) is prepared to step in when needed. Clearly define where help desk responsibilities end, and where COE extended support responsibilities begin.
- ✓ **Encourage collaboration between COE and system administrators**: Ensure that COE members and Fabric administrators have a direct escalation path to reach global administrators for Microsoft 365 and Azure. It's critical to have a communication channel when a widespread issue arises that's beyond the scope of Fabric.
- ✓ **Create a feedback loop from the COE back to the help desk**: When the COE learns of new information, the IT knowledgebase should be updated. The goal is for the primary help desk personnel to continually become better equipped at handling more issues in the future.
- ✓ **Create a feedback loop from the help desk to the COE**: When support personnel observe redundancies or inefficiencies, they can communicate that information to the COE, who might choose to improve the knowledgebase or get involved (particularly if it relates to governance or security).

Questions to ask

Use questions like those found below to assess user support.

- Who is responsible for supporting enterprise data and BI solutions? What about self-service solutions?
- How are the business impact and urgency of issues identified to effectively detect and prioritize critical issues?
- Is there a clear process for business users to report issues with data and BI solutions? How does this differ between enterprise and self-service solutions? What are the escalation paths?
- What types of issues do content creators and consumers typically experience? For example, do they experience data quality issues, performance issues, access issues, and others?
- Are any issues closed without them being resolved? Are there "known issues" in data items or reports today?
- Is a process in place for data asset owners to escalate issues with self-service BI solutions to central teams like the COE?
- How frequent are issues in the data and existing solutions? What proportion of these issues are found before they impact business end users?
- How long does it typically take to resolve issues? Is this timing sufficient for business users?
- What are examples of recent issues and the concrete impact on the business?
- Do enterprise teams and content creators know how to report Fabric issues to Microsoft? Can enterprise teams effectively leverage community resources to unblock critical issues?

> **⊗ Caution**
>
> When assessing user support and describing risks or issues, be careful to use neutral language that doesn't place blame on individuals or teams. Ensure everyone's perspective is fairly represented in an assessment. Focus on objective facts to accurately understand and describe the context.

Maturity levels

The following maturity levels will help you assess the current state of your Power BI user support.

Expand table

Level	State of user support
100: Initial	• Individual business units find effective ways of supporting each other. However, the tactics and practices are siloed and not consistently applied. • An internal discussion channel is available. However, it's not monitored closely. Therefore, the user experience is inconsistent.
200: Repeatable	• The COE actively encourages intra-team support and growth of the champions network. • The internal discussion channel gains traction. It's become known as the default place for questions and discussions. • The help desk handles a small number of the most common technical support issues.
300: Defined	• The internal discussion channel is popular and largely self-sustaining. The COE actively monitors and manages the discussion channel to ensure that all questions are answered quickly and correctly. • A help desk tracking system is in place to monitor support frequency, response topics, and priorities. • The COE provides appropriate extended support when required.
400: Capable	• The help desk is fully trained and prepared to handle a broader number of known and expected technical support issues. • SLAs are in place to define help desk support expectations, including extended support. The expectations are documented and communicated so they're clear to everyone involved.
500: Efficient	• Bidirectional feedback loops exist between the help desk and the COE. • Key performance indicators measure satisfaction and support methods. • Automation is in place to allow the help desk to react faster and reduce errors (for example, use of APIs and scripts).

Related content

In the next article in the Microsoft Fabric adoption roadmap series, learn about system oversight and administration activities.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: System oversight

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

System oversight—also known as Fabric administration—is the ongoing, day-to-day, administrative activities. It's specifically concerned with:

- **Governance**: Enact governance guidelines and policies to support self-service and enterprise data and business intelligence (BI) scenarios.
- **User empowerment**: Facilitate and support the internal processes and systems that empower the internal user community to the extent possible, while adhering to the organization's regulations and requirements.
- **Adoption**: Allow for broader organizational adoption of Fabric with effective governance and data management practices.

> **Important**
>
> Your organizational data culture objectives provide direction for your governance decisions, which in turn dictate how Fabric administration activities take place and by whom.

System oversight is a broad and deep topic. The goal of this article is to introduce some of the most important considerations and actions to help you become successful with your organizational adoption objectives.

Fabric administrators

The Fabric administrator role is a defined role in Microsoft 365, which delegates a subset of management activities. Global Microsoft 365 administrators are implicitly Fabric administrators. Power Platform administrators are also implicitly Fabric administrators.

A key governance decision is who to assign as a Fabric administrator. It's a centralized role that affects your entire tenant. Ideally, there are two to four people in the

organization who are capable of managing Fabric. Your administrators should operate in close coordination with the Center of Excellence (COE).

High privilege role

The Fabric administrator role is a high privilege role because:

- **User experience**: Settings that are managed by a Fabric administrator have a significant effect on user capabilities and user experience. For more information, see Govern tenant settings.
- **Full security access**: Fabric administrators can update access permissions for workspaces in the tenant. The result is that an administrator can allow permission to view or download data and reports as they see fit. For more information, see Govern tenant settings.
- **Personal workspace access**: Fabric administrators can access contents and govern the personal workspace of any user.
- **Metadata**: Fabric administrators can view all tenant metadata, including all user activities that occur in the Fabric portal (described in the Auditing and monitoring section below).

> **Important**
>
> Having too many Fabric administrators is a risk. It increases the probability of unapproved, unintended, or inconsistent management of the tenant.

Roles and responsibilities

The types of activities that an administrator will do on a day-to-day basis will differ between organizations. What's important, and given priority in your data culture, will heavily influence what an administrator does to support business-led self-service, managed self-service, and enterprise data and BI scenarios. For more information, see the Content ownership and management article.

> **Tip**
>
> The best type of person to serve as a Fabric administrator is one who has enough knowledge about the tools and workloads to understand what self-service users need to accomplish. With this understanding, the administrator can balance user empowerment and governance.

In addition to the Fabric administrator, there are other roles which use the term *administrator*. The following table describes the roles that are commonly and regularly used.

Expand table

Role	Scope	Description
Fabric administrator	Tenant	Manages tenant settings and other settings in the Fabric portal. All general references to *administrator* in this article refer to this type of administrator.
Capacity administrator	One capacity	Manages workspaces and workloads, and monitors the health of a Fabric capacity.
Data gateway administrator	One gateway	Manages gateway data source configuration, credentials, and users assignments. Might also handle gateway software updates (or collaborate with infrastructure team on updates).
Workspace administrator	One workspace	Manages workspace settings and access.

The Fabric ecosystem of workloads is broad and deep. There are many ways that Fabric integrates with other systems and platforms. From time to time, it'll be necessary to work with other administrators and IT professionals. For more information, see Collaborate with other administrators.

The remainder of this article provides an overview of the most common activities that a Fabric administrator does. It focuses on activities that are important to carry out effectively when taking a strategic approach to organizational adoption.

Service management

Overseeing the tenant is a crucial aspect to ensure that all users have a good experience with Power BI. A few of the key governance responsibilities of a Fabric administrator include:

- **Tenant settings**: Control which Power BI features and capabilities are enabled, and for which users in your organization.
- **Domains**: Group together two or more workspaces that have similar characteristics.
- **Workspaces**: Review and manage workspaces in the tenant.
- **Embed codes**: Govern which reports have been published publicly on the internet.
- **Organizational visuals**: Register and manage organizational visuals.

- **Azure connections**: Integrate with Azure services to provide additional functionality.

For more information, see Tenant administration.

User machines and devices

The adoption of Fabric depends directly on content creators and consumers having the tools and applications they need. Here are some important questions to consider.

- How will users request access to new tools? Will access to licenses, data, and training be available to help users use tools effectively?
- How will content consumers view content that's been published by others?
- How will content creators develop, manage, and publish content? What's your criteria for deciding which tools and applications are appropriate for which use cases?
- How will you install and set up tools? Does that include related prerequisites and data connectivity components?
- How will you manage ongoing updates for tools and applications?

For more information, see User tools and devices.

Architecture

In the context of Fabric, architecture relates to data architecture, capacity management, and data gateway architecture and management.

Data architecture

Data architecture refers to the principles, practices, and methodologies that govern and define what data is collected, and how it's ingested, stored, managed, integrated, modeled, and used.

There are many data architecture decisions to make. Frequently the COE engages in data architecture design and planning. It's common for administrators to get involved as well, especially when they manage databases or Azure infrastructure.

> **Important**
>
> Data architecture decisions have a significant impact on Fabric adoption, user satisfaction, and individual project success rates.

A few data architecture considerations that affect adoption include:

- Where does Fabric fit into the organization's entire data architecture? Are there other existing components such as an enterprise data warehouse (EDW) or a data lake that will be important to factor into plans?
- Is Fabric used end-to-end for data preparation, data modeling, and data presentation or is Fabric used for only some of those capabilities?
- Are managed self-service patterns followed to find the best balance between data reusability and report creator flexibility?
- Where will users consume the content? Generally, the three main ways to deliver content are: the Fabric portal, Power BI Report Server, and embedded in custom applications. Additionally, Microsoft Teams is a convenient alternative for users who spend a lot of time in Teams.
- Who is responsible for managing and maintaining the data architecture? Is it a centralized team, or a decentralized team? How is the COE represented in this team? Are certain skillsets required?
- What data sources are the most important? What types of data will we be acquiring?
- What semantic model connectivity mode and storage mode choices (for example, Direct Lake, import, live connection, DirectQuery, or composite model frameworks) are the best fit for the use cases?
- To what extent is data reusability encouraged using lakehouses, warehouses, and shared semantic models?
- To what extent is the reusability of data preparation logic and advanced data preparation encouraged by using data pipelines, notebooks, and dataflows?

It's important for administrators to become fully aware of Fabric's technical capabilities—as well as the needs and goals of their stakeholders—before they make architectural decisions.

> **Tip**
>
> Get into the good habit of completing a technical proof of concept (POC) to test out assumptions and ideas. Some organizations also call them *micro-projects* when the goal is to deliver a small unit of work. The goal of a POC is to address unknowns and reduce risk as early as possible. A POC doesn't have to be throwaway work, but it should be narrow in scope. Best practices reviews, as described in the **Mentoring and user enablement** article, are another useful way to help content creators with important architectural decisions.

Capacity management

Capacity includes features and capabilities to deliver analytics solutions at scale. There are two types of Fabric organizational licenses: Premium per User (PPU) and capacity. There are several types of capacity licenses. The type of capacity license determines which Fabric workloads are supported.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

The use of capacity can play a significant role in your strategy for creating, managing, publishing, and distributing content. A few of the top reasons to invest in capacity include:

- Unlimited Power BI content distribution to large numbers of read-only users. Content consumption by users with a free Power BI license is available in Premium capacity only, not PPU. Content consumption by free users is also available with an F64 Fabric capacity license or higher.
- Access to Fabric experiences for producing end-to-end analytics.
- Deployment pipelines to manage the publication of content to development, test, and production workspaces. They're highly recommended for critical content to improve release stability.
- XMLA endpoint, which is an industry standard protocol for managing and publishing a semantic model, or querying the semantic model from any XMLA-compliant tool.
- Increased model size limits, including large semantic model support.
- More frequent data refreshes.
- Storage of data in a specific geographic area that's different from the home region.

The above list isn't all-inclusive. For a complete list, see Power BI Premium features.

Manage Fabric capacity

Overseeing the health of Fabric capacity is an essential ongoing activity for administrators. Each capacity SKU includes a set of resources. Capacity units (CUs) are used to measure compute resources for each SKU.

> **⊗ Caution**
>
> Lack of management, and consistently exceeding the limits of your capacity resources can often result in performance challenges and user experience challenges. Both challenges, if not managed correctly, can contribute to negative impact on adoption efforts.

Suggestions for managing Fabric capacity:

- Define who is responsible for managing the capacity. Confirm the roles and responsibilities so that it's clear what action will be taken, why, when, and by whom.
- Create a specific set of criteria for content that will be published to capacity. It's especially relevant when a single capacity is used by multiple business units because the potential exists to disrupt other users if the capacity isn't well-managed. Consider requiring a best practices review (such as reasonable semantic model size and efficient calculations) before publishing new content to a production capacity.
- Regularly use the Fabric capacity metrics app to understand resource utilization and patterns for the capacity. Most importantly, look for consistent patterns of overutilization, which will contribute to user disruptions. An analysis of usage patterns should also make you aware if the capacity is underutilized, indicating more value could be gained from the investment.
- Set the tenant setting so Fabric notifies you if the capacity becomes overloaded, or if an outage or incident occurs.

Autoscale

Autoscale is intended to handle occasional or unexpected bursts in capacity usage levels. Autoscale can respond to these bursts by automatically increasing CPU resources to support the increased workload.

Automated scaling up reduces the risk of performance and user experience challenges in exchange for a financial impact. If the capacity isn't well-managed, autoscale might trigger more often than expected. In this case, the metrics app can help you to determine underlying issues and do capacity planning.

Decentralized capacity management

Capacity administrators are responsible for assigning workspaces to a specific capacity.

Be aware that workspace administrators can also assign a workspace to PPU if the workspace administrator possesses a PPU license. However, it would require that all other workspace users must also have a PPU license to collaborate on, or view, Power BI content in the workspace. Other Fabric workloads can't be included in a workspace assigned to PPU.

It's possible to set up multiple capacities to facilitate decentralized management by different business units. Decentralizing management of certain aspects of Fabric is a great way to balance agility and control.

Here's an example that describes one way you could manage your capacity.

- Purchase a P3 capacity node in Microsoft 365. It includes 32 virtual cores (v-cores).
- Use 16 v-cores to create the first capacity. It will be used by the Sales team.
- Use 8 v-cores to create the second capacity. It will be used by the Operations team.
- Use the remaining 8 v-cores to create the third capacity. It will support general use.

The previous example has several advantages.

- Separate capacity administrators can be set up for each capacity. Therefore, it facilitates decentralized management situations.
- If a capacity isn't well-managed, the effect is confined to that capacity only. The other capacities aren't impacted.
- Billing and chargebacks to other business units are straightforward.
- Different workspaces can be easily assigned to the separate capacities.

However, the previous example has disadvantages, too.

- The limits per capacity are lower. The maximum memory size allowed for semantic models isn't the entire P3 capacity node size that was purchased. Rather, it's the assigned capacity size where the semantic model is hosted.
- It's more likely one of the smaller capacities will need to be scaled up at some point in time.
- There are more capacities to manage in the tenant.

> **Note**
>
> Resources for Power BI Premium per Capacity are referred to as v-cores. However, a Fabric capacity refers to them as capacity units (CUs). The scale for CUs and v-cores

is different for each SKU. For more information, see the Fabric licensing documentation.

Data gateway architecture and management

A data gateway facilitates the secure and efficient transfer of data between organizational data sources and the Fabric service. A data gateway is needed for data connectivity to on-premises or cloud services when a data source is:

- Located within the enterprise data center.
- Configured behind a firewall.
- Within a virtual network.
- Within a virtual machine.

There are three types of gateways.

- **On-premises data gateway (standard mode)** is a gateway service that supports connections to registered data sources for many users to use. The gateway software installations and updates are installed on a machine that's managed by the customer.
- **On-premises data gateway (personal mode)** is a gateway service that supports data refresh only. This gateway mode is typically installed on the PC of a content creator. It supports use by one user only. It doesn't support live connection or DirectQuery connections.
- **Virtual network data gateway** is a Microsoft managed service that supports connectivity for many users. Specifically, it supports connectivity for semantic models and dataflows stored in workspaces assigned to Premium capacity or Premium Per User.

Tip

The decision of **who can install gateway software** is a governance decision. For most organizations, use of the data gateway in standard mode, or a virtual network data gateway, should be strongly encouraged. They're far more scalable, manageable, and auditable than data gateways in personal mode.

Decentralized gateway management

The On-premises data gateway (standard mode) and Virtual network data gateway support specific data source types that can be registered, together with connection

details and how credentials are stored. Users can be granted permission to use the gateway data source so that they can schedule a refresh or run DirectQuery queries.

Certain aspects of gateway management can be done effectively on a decentralized basis to balance agility and control. For example, the Operations group might have a gateway dedicated to its team of self-service content creators and data owners.

Decentralized gateway management works best when it's a joint effort as follows.

Managed by the decentralized data owners:

- Departmental data source connectivity information and privacy levels.
- Departmental data source stored credentials (including responsibility for updating routine password changes).
- Departmental data source users who are permitted to use each data source.

Managed by centralized data owners (includes data sources that are used broadly across the organization; management is centralized to avoid duplicated data sources):

- Centralized data source connectivity information and privacy levels.
- Centralized data source stored credentials (including responsibility for updating routine password changes).
- Centralized data source users who are permitted to use each data source.

Managed by IT:

- Gateway software updates (gateway updates are usually released monthly).
- Installation of drivers and custom connectors (the same ones that are installed on user machines).
- Gateway cluster management (number of machines in the gateway cluster for high availability, disaster recovery, and to eliminate a single point of failure, which can cause significant user disruptions).
- Server management (for example, operating system, RAM, CPU, or networking connectivity).
- Management and backup of gateway encryption keys.
- Monitoring of gateway logs to assess when scale-up or scale-out is necessary.
- Alerting of downtime or persistent low resources on the gateway machine.

> **Tip**
>
> Allowing a decentralized team to manage certain aspects of the gateway means they can move faster. The tradeoff of decentralized gateway management does mean running more gateway servers so that each can be dedicated to a specific area of the organization. If gateway management is handled entirely by IT, it's

imperative to have a good process in place to quickly handle requests to add data sources and apply user updates.

User licenses

Every user needs a commercial license, which is integrated with a Microsoft Entra identity. The user license could be Free, Pro, or Premium Per User (PPU).

A user license is obtained via a subscription, which authorizes a certain number of licenses with a start and end date.

> **Note**
>
> Although each user requires a license, a Pro or PPU license is only required to share Power BI content. Users with a free license can create and share Fabric content other than Power BI items.

There are two approaches to procuring subscriptions.

- **Centralized**: Microsoft 365 billing administrator purchases a subscription for Pro or PPU. It's the most common way to manage subscriptions and assign licenses.
- **Decentralized**: Individual departments purchase a subscription via self-service purchasing.

Self-service purchasing

An important governance decision relates to what extent self-service purchasing will be allowed or encouraged.

Self-service purchasing is useful for:

- Larger organizations with decentralized business units that have purchasing authority and want to handle payment directly with a credit card.
- Organizations that intend to make it as easy as possible to purchase subscriptions on a monthly commitment.

Consider disabling self-service purchasing when:

- Centralized procurement processes are in place to meet regulatory, security, and governance requirements.
- Discounted pricing is obtained through an Enterprise Agreement (EA).

- Existing processes are in place to handle intercompany chargebacks.
- Existing processes are in place to handle group-based licensing assignments.
- Prerequisites are required for obtaining a license, such as approval, justification, training, or a governance policy requirement.
- There's a valid need, such as a regulatory requirement, to control access closely.

User license trials

Another important governance decision is whether user license trials are allowed. By default, trials are enabled. That means when content is shared with a colleague, if the recipient doesn't have a Pro or PPU license, they'll be prompted to start a trial to view the content (if the content doesn't reside within a workspace backed by capacity). The trial experience is intended to be a convenience that allows users to continue with their normal workflow.

Generally, disabling trials isn't recommended. It can encourage users to seek workarounds, perhaps by exporting data or working outside of supported tools and processes.

Consider disabling trials only when:

- There are serious cost concerns that would make it unlikely to grant full licenses at the end of the trial period.
- Prerequisites are required for obtaining a license (such as approval, justification, or a training requirement). It's not sufficient to meet this requirement during the trial period.
- There's a valid need, such as a regulatory requirement, to control access to the Fabric service closely.

> **Tip**
>
> Don't introduce too many barriers to obtaining a Fabric license. Users who need to get work done will find a way, and that way might involve workarounds that aren't ideal. For instance, without a license to use Fabric, people might rely far too much on sharing files on a file system or via email when significantly better approaches are available.

Cost management

Managing and optimizing the cost of cloud services, like Fabric, is an important activity. Here are several activities you can consider.

- Analyze who is using—and, more to the point, not using—their allocated Fabric licenses and make necessary adjustments. Fabric usage is analyzed using the activity log.
- Analyze the cost effectiveness of capacity or Premium Per User. In addition to the additional features, perform a cost/benefit analysis to determine whether capacity licensing is more cost-effective when there are a large number of consumers.
- Carefully monitor and manage Fabric capacity. Understanding usage patterns over time will allow you to predict when to purchase more capacity. For example, you might choose to scale up a single capacity from a P1 to P2, or scale out from one P1 capacity to two P1 capacities.
- If there are occasional spikes in the level of usage, use of autoscale with Fabric is recommended to ensure the user experience isn't interrupted. Autoscale will scale up capacity resources for 24 hours, then scale them back down to normal levels (if sustained activity isn't present). Manage autoscale cost by constraining the maximum number of v-cores, and/or with spending limits set in Azure. Due to the pricing model, autoscale is best suited to handle occasional unplanned increases in usage.
- For Azure data sources, co-locate them in the same region as your Fabric tenant whenever possible. It will avoid incurring Azure egress charges. Data egress charges are minimal, but at scale can add up to be considerable unplanned costs.

Security, information protection, and data loss prevention

Security, information protection, and data loss prevention (DLP) are joint responsibilities among all content creators, consumers, and administrators. That's no small task because there's sensitive information everywhere: personal data, customer data, or customer-authored data, protected health information, intellectual property, proprietary organizational information, just to name a few. Governmental, industry, and contractual regulations could have a significant impact on the governance guidelines and policies that you create related to security.

The Power BI security whitepaper is an excellent resource for understanding the breadth of considerations, including aspects that Microsoft manages. This section will introduce several topics that customers are responsible for managing.

User responsibilities

Some organizations ask Fabric users to accept a self-service user acknowledgment. It's a document that explains the user's responsibilities and expectations for safeguarding

organizational data.

One way to automate its implementation is with a Microsoft Entra terms of use policy. The user is required to view and agree to the policy before they're permitted to visit the Fabric portal for the first time. You can also require it to be acknowledged on a recurring basis, like an annual renewal.

Data security

In a cloud shared responsibility model, securing the data is always the responsibility of the customer. With a self-service data platform, self-service content creators have responsibility for properly securing the content that they shared with colleagues.

The COE should provide documentation and training where relevant to assist content creators with best practices (particularly situations for dealing with ultra-sensitive data).

Administrators can help by following best practices themselves. Administrators can also raise concerns when they see issues that could be discovered when managing workspaces, auditing user activities, or managing gateway credentials and users. There are also several tenant settings that are usually restricted except for a few users (for instance, the ability to publish to web or the ability to publish apps to the entire organization).

External guest users

External users—such as partners, customers, vendors, and consultants—are a common occurrence for some organizations, and rare for others. How you handle external users is a governance decision.

External user access is controlled by tenant settings and certain Microsoft Entra ID settings. For details of external user considerations, review the Distribute Power BI content to external guest users using Microsoft Entra B2B whitepaper.

Information protection and data loss prevention

Fabric supports capabilities for information protection and data loss prevention (DLP) in the following ways.

- **Information protection**: Microsoft Purview Information Protection (formerly known as Microsoft Information Protection) includes capabilities for discovering, classifying, and protecting data. A key principle is that data can be better protected

once it's been classified. The key building block for classifying data is sensitivity labels. For more information, see Information protection for Power BI planning.

- **Data loss prevention for Power BI**: Microsoft Purview Data Loss Prevention (formerly known as Office 365 Data Loss Prevention) supports DLP policies for Power BI. By using sensitivity labels or sensitive information types, DLP policies for Power BI help an organization locate sensitive semantic models. For more information, see Data loss prevention for Power BI planning.
- **Microsoft Defender for Cloud Apps**: Microsoft Defender for Cloud Apps (formerly known as Microsoft Cloud App Security) supports policies that help protect data, including real-time controls when users interact with the Power BI service. For more information, see Defender for Cloud Apps for Power BI planning.

Data residency

For organizations with requirements to store data within a geographic region, Fabric capacity can be set for a specific region that's different from the home region of the Fabric tenant.

Encryption keys

Microsoft handles encryption of *data at rest* in Microsoft data centers with transparent server-side encryption and auto-rotation of certificates. For customers with regulatory requirements to manage the Premium encryption key themselves, Premium capacity can be configured to use Azure Key Vault. Using customer-managed keys—also known as *bring-your-own-key* or *BYOK*—is a precaution to ensure that, in the event of a human error by a service operator, customer data can't be exposed.

Be aware that Premium Per User (PPU) only supports BYOK when it's enabled for the entire Fabric tenant.

Auditing and monitoring

It's critical that you make use of auditing data to analyze adoption efforts, understand usage patterns, educate users, support users, mitigate risk, improve compliance, manage license costs, and monitor performance. For more information about why auditing your data is valuable, see Auditing and monitoring overview.

There are different ways to approach auditing and monitoring depending on your role and your objectives. The following articles describe various considerations and planning activities.

- **Report-level auditing**: Techniques that report creators can use to understand which users are using the reports that they create, publish, and share.
- **Data-level auditing**: Methods that data creators can use to track the performance and usage patterns of data assets that they create, publish, and share.
- **Tenant-level auditing**: Key decisions and actions administrators can take to create an end-to-end auditing solution.
- **Tenant-level monitoring**: Tactical actions administrators can take to monitor the Power BI service, including updates and announcements.

REST APIs

The Power BI REST APIs and the Fabric REST APIs provide a wealth of information about your Fabric tenant. Retrieving data by using the REST APIs should play an important role in managing and governing a Fabric implementation. For more information about planning for the use of REST APIs for auditing, see Tenant-level auditing.

You can retrieve auditing data to build an auditing solution, manage content programmatically, or increase the efficiency of routine actions. The following table presents some actions you can perform with the REST APIs.

Expand table

Action	Documentation resource(s)
Audit user activities	REST API to get activity events
Audit workspaces, items, and permissions	Collection of asynchronous metadata scanning REST APIs to obtain a tenant inventory
Audit content shared to entire organization	REST API to check use of widely shared links
Audit tenant settings	REST API to check tenant settings
Publish content	REST API to deploy items from a deployment pipeline or clone a report to another workspace
Manage content	REST API to refresh a semantic model or take over ownership of a semantic model
Manage gateway data sources	REST API to update credentials for a gateway data source
Export content	REST API to export a report
Create workspaces	REST API to create a new workspace

Action	Documentation resource(s)
Manage workspace permissions	REST API to assign user permissions to a workspace
Update workspace name or description	REST API to update workspace attributes
Restore a workspace	REST API to restore a deleted workspace
Programmatically retrieve a query result from a semantic model	REST API to run a DAX query against a semantic model
Assign workspaces to capacity	REST API to assign workspaces to capacity
Programmatically change a data model	Tabular Object Model (TOM) API
Embed Power BI content in custom applications	Power BI embedded analytics client APIs

> **Tip**
>
> There are many other Power BI REST APIs. For a complete list, see **Using the Power BI REST APIs**.

Planning for change

Every month, Microsoft releases new Fabric features and capabilities. To be effective, it's crucial that everyone involved with system oversight stays current. For more information, see Tenant-level monitoring.

> **Important**
>
> Don't underestimate the importance of staying current. If you get a few months behind on announcements, it can become difficult to properly manage Fabric and support your users.

Considerations and key actions

Checklist - Considerations and key actions you can take for system oversight follow.

Improve system oversight:

- ✓ **Verify who is permitted to be a Fabric administrator**: If possible, reduce the number of people granted the Fabric administrator role if it's more than a few people.
- ✓ **Use PIM for occasional administrators**: If you have people who *occasionally* need Fabric administrator rights, consider implementing Privileged Identity Management (PIM) in Microsoft Entra ID. It's designed to assign just-in-time role permissions that expire after a few hours.
- ✓ **Train administrators**: Check the status of cross-training and documentation in place for handling Fabric administration responsibilities. Ensure that a backup person is trained so that needs can be met timely, in a consistent way.

Improve management of the Fabric service:

- ✓ **Review tenant settings**: Conduct a review of all tenant settings to ensure they're aligned with data culture objectives and governance guidelines and policies. Verify which groups are assigned for each setting.
- ✓ **Document the tenant settings**: Create documentation of your tenant settings for the internal Fabric community and post it in the centralized portal. Include which groups a user would need to request to be able to use a feature. Use the Get Tenant Settings REST API to make the process more efficient, and to create snapshots of the settings on a regular basis.
- ✓ **Customize the *Get Help* links**: When user resources are established, as described in the Mentoring and user enablement article, update the tenant setting to customize the links under the *Get Help* menu option. It will direct users to your documentation, community, and help.

Improve management of user machines and devices:

- ✓ **Create a consistent onboarding process**: Review your process for how onboarding of new content creators is handled. Determine if new requests for software, such as Power BI Desktop, and user licenses (Free, Pro, or PPU) can be handled together. It can simplify onboarding since new content creators won't always know what to ask for.
- ✓ **Handle user machine updates**: Ensure an automated process is in place to install and update software, drivers, and settings to ensure all users have the same version.

Data architecture planning:

- ✓ **Assess what your end-to-end data architecture looks like**: Make sure you're clear on:

    - How Fabric is currently used by the different business units in your organization versus how you want Fabric to be used. Determine if there's a gap.
    - If there are any risks that should be addressed.
    - If there are any high-maintenance situations to be addressed.
    - What data sources are important for Fabric users, and how they're documented and discovered.
- ✓ **Review existing data gateways**: Find out what gateways are being used throughout your organization. Verify that gateway administrators and users are set correctly. Verify who is supporting each gateway, and that there's a reliable process in place to keep the gateway servers up to date.
- ✓ **Verify use of personal gateways**: Check the number of personal gateways that are in use, and by whom. If there's significant usage, take steps to move towards use of the standard mode gateway.

Improve management of user licenses:

- ✓ **Review the process to request a user license**: Clarify what the process is, including any prerequisites, for users to obtain a license. Determine whether there are improvements to be made to the process.
- ✓ **Determine how to handle self-service license purchasing**: Clarify whether self-service licensing purchasing is enabled. Update the settings if they don't match your intentions for how licenses can be purchased.
- ✓ **Confirm how user trials are handled**: Verify user license trials are enabled or disabled. Be aware that all user trials are Premium Per User. They apply to Free licensed users signing up for a trial, and Pro users signing up for a Premium Per User trial.

Improve cost management:

- ✓ **Determine your cost management objectives**: Consider how to balance cost, features, usage patterns, and effective utilization of resources. Schedule a routine process to evaluate costs, at least annually.
- ✓ **Obtain activity log data**: Ensure you have access to the activity log data to assist with cost analysis. It can be used to understand who is—or isn't—using the license assigned to them.

Improve security and data protection:

- ✓ **Clarify exactly what the expectations are for data protection**: Ensure the expectations for data protection, such as how to use sensitivity labels, are documented and communicated to users.
- ✓ **Determine how to handle external users**: Understand and document the organizational policies around sharing Fabric content with external users. Ensure

that settings in Fabric support your policies for external users.
- ✓ **Set up monitoring**: Investigate the use of Microsoft Defender for Cloud Apps to monitor user behavior and activities in Fabric.

Improve auditing and monitoring:

- ✓ **Plan for auditing needs**: Collect and document the key business requirements for an auditing solution. Consider your priorities for auditing and monitoring. Make key decisions related to the type of auditing solution, permissions, technologies to be used, and data needs. Consult with IT to clarify what auditing processes currently exist, and what preferences of requirements exist for building a new solution.
- ✓ **Consider roles and responsibilities**: Identify which teams will be involved in building an auditing solution, as well as the ongoing analysis of the auditing data.
- ✓ **Extract and store user activity data**: If you aren't currently extracting and storing the raw data, begin retrieving user activity data.
- ✓ **Extract and store snapshots of tenant inventory data**: Begin retrieving metadata to build a tenant inventory, which describes all workspaces and items.
- ✓ **Extract and store snapshots of users and groups data**: Begin retrieving metadata about users, groups, and service principals.
- ✓ **Create a curated data model**: Perform data cleansing and transformations of the raw data to create a curated data model that'll support analytical reporting for your auditing solution.
- ✓ **Analyze auditing data and act on the results**: Create analytic reports to analyze the curated auditing data. Clarify what actions are expected to be taken, by whom, and when.
- ✓ **Include additional auditing data**: Over time, determine whether other auditing data would be helpful to complement the activity log data, such as security data.

> **Tip**
>
> For more information, see **Tenant-level auditing**.

Use the REST APIs:

- ✓ **Plan for your use of the REST APIs**: Consider what data would be most useful to retrieve from the Power BI REST APIs and the Fabric REST APIs.
- ✓ **Conduct a proof of concept**: Do a small proof of concept to validate data needs, technology choices, and permissions.

Questions to ask

Use questions like those found below to assess system oversight.

- Are there atypical administration settings enabled or disabled? For example, is the entire organization allowed to publish to web (we strongly advise restricting this feature).
- Do administration settings and policies align with, or inhibit, business the way user work?
- Is there a process in place to critically appraise new settings and decide how to set them? Alternatively, are only the most restrictive settings set as a precaution?
- Are Microsoft Entra security groups used to manage who can do what?
- Do central teams have visibility of effective auditing and monitoring tools?
- Do monitoring solutions depict information about the data assets, user activities, or both?
- Are auditing and monitoring tools actionable? Are there clear thresholds and actions set, or do monitoring reports simply describe what's in the data estate?
- Is Azure Log Analytics used (or planned to be used) for detailed monitoring of Fabric capacities? Are the potential benefits and cost of Azure Log Analytics clear to decision makers?
- Are sensitivity labels and data loss prevention policies used? Are the potential benefits and cost of these clear to decision makers?
- Do administrators know the current number of licenses and licensing cost? What proportion of the total BI spend goes to Fabric capacity, and to Pro and PPU licenses? If the organization is only using Pro licenses for Power BI content, could the number of users and usage patterns warrant a cost-effective switch to Power BI Premium or Fabric capacity?

Maturity levels

The following maturity levels will help you assess the current state of your Power BI system oversight.

Expand table

Level	State of system oversight
100: Initial	• Tenant settings are configured independently by one or more administrators based on their best judgment. • Architecture needs, such as gateways and capacities, are satisfied on an as-needed basis. However, there isn't a strategic plan. • Fabric activity logs are unused, or selectively used for tactical purposes.
200: Repeatable	• The tenant settings purposefully align with established governance guidelines and policies. All tenant settings are reviewed regularly. • A small number of specific administrators are selected. All administrators have a good understanding of what users are trying to accomplish in Fabric, so they're in a good position to support users. • A well-defined process exists for users to request licenses and software. Request forms are easy for users to find. Self-service purchasing settings are specified. • Sensitivity labels are configured in Microsoft 365. However, use of labels remains inconsistent. The advantages of data protection aren't well understood by users.
300: Defined	• The tenant settings are fully documented in the centralized portal for users to reference, including how to request access to the correct groups. • Cross-training and documentation exist for administrators to ensure continuity, stability, and consistency. • Sensitivity labels are assigned to content consistently. The advantages of using sensitivity labels for data protection are understood by users. • An automated process is in place to export Fabric activity log and API data to a secure location for reporting and auditing.
400: Capable	• Administrators work closely with the COE and governance teams to provide oversight of Fabric. A balance of user empowerment and governance is successfully achieved. • Decentralized management of data architecture (such as gateways or capacity management) is effectively handled to balance agility and control. • Automated policies are set up and actively monitored in Microsoft Defender for Cloud Apps for data loss prevention. • Activity log and API data is actively analyzed to monitor and audit Fabric activities. Proactive action is taken based on the data.

Level	State of system oversight
500: Efficient	• The Fabric administrators work closely with the COE actively stay current. Blog posts and release plans from the Fabric product team are reviewed frequently to plan for upcoming changes. • Regular cost management analysis is done to ensure user needs are met in a cost-effective way. • The Fabric REST API is used to retrieve tenant setting values on a regular basis. • Activity log and API data is actively used to inform and improve adoption and governance efforts.

Related content

For more information about system oversight and Fabric administration, see the following resources.

- Administer Microsoft Fabric
- Administer Power BI - Part 1
- Administer Power BI - Part 2
- Administrator in a Day Training – Day 1
- Administrator in a Day Training – Day 2
- Power BI security whitepaper
- External guest users whitepaper
- Power BI implementation planning

In the next article in the Microsoft Fabric adoption roadmap series, learn about effective change management.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap: Change management

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

When working toward improved data and business intelligence (BI) adoption, you should plan for effective *change management*. In the context of data and BI, change management includes procedures that address the impact of change for people in an organization. These procedures safeguard against disruption and productivity loss due to changes in solutions or processes.

> **Note**
>
> Effective change management is particularly important when you migrate to Power BI.

Effective change management improves adoption and productivity because it:

- Helps content creators and consumers use analytics more effectively and sooner.
- Limits redundancy in data, analytical tools, and solutions.
- Reduces the likelihood of risk-creating behaviors that affect shared resources (like Fabric capacity) or organizational compliance (like data security and privacy).
- Mitigates resistance to change that obstructs planning and inhibits user adoption.
- Mitigates the impact of change and improving user wellbeing by reducing the potential for disruption, stress, and conflict.

Effective change management is critical for successful adoption at all levels. To successfully manage change, consider the key actions and activities described in the following sections.

> **Important**
>
> Change management is a fundamental obstacle to success in many organizations. Effective change management requires that you understand that it's about people —not tools or processes.

Successful change management involves empathy and communication. Ensure that change isn't forced or resistance to change is ignored, because it can widen organizational divides and further inhibit effectiveness.

Tip

Whenever possible, we recommend that you describe and promote change as *improvement*—it's much less threatening. For many people, *change* implies a cost in terms of effort, focus, and time. Alternatively, *improvement* means a benefit because it's about making something better.

Types of change to manage

When implementing data and BI solutions, you should manage different types of change. Also, depending on the scale and scope of your implementation, you should address different aspects of change.

Consider the following types of change to manage when you plan for Fabric adoption.

Process-level changes

Process-level changes are changes that affect a broader user community or the entire organization. These changes typically have a larger impact, and so they require more effort to manage. Specifically, this change management effort includes specific plans and activities.

Here are some examples of process-level changes.

- Change from centralized to decentralized approaches to ownership (change in content ownership and management).
- Change from enterprise to departmental, or from team to personal content delivery (change in content delivery scope).
- Change of central team structure (for example, forming a Center of Excellence).
- Changes in governance policies.
- Migration from other analytics products to Fabric, and the changes this migration involves, like:
  - The separation of semantic models and reports, and a model-based approach to analytics.
  - Transitioning from exports or static reports to interactive analytical reports, which can involve filtering and cross-filtering.

    - Moving from distributing reports as PowerPoint files or flat files to accessing reports directly from the Fabric portal.
    - Shifting from information in tables, paginated reports, and spreadsheets to interactive visualizations and charts.
    - Changing from an on-premises or platform as a service (PaaS) platform to a software as a service (SaaS) tool.

> **Note**
>
> Typically, giving up export-based processes or Excel reporting is a significant challenge. That's because these methods are usually deeply engrained in the organization and are tied to the autonomy and data skills of your users.

Solution-level changes

Solution-level changes are changes that affect a single solution or set of solutions. These changes limit their impact to the user community of those solutions and their dependent processes. Although solution-level changes typically have a lower impact, they also tend to occur more frequently.

> **Note**
>
> In the context of this article, a *solution* is built to address specific business needs for users. A solution can take many forms, such as a data pipeline, a lakehouse, a semantic model, or a report. The considerations for change management described in this article are relevant for all types of solutions, and not only reporting projects.

Here are some examples of solution-level changes.

- Changes in calculation logic for KPIs or measures.
- Changes in how master data or hierarchies for business attributes are mapped, grouped, or described.
- Changes in data freshness, detail, format, or complexity.
- Introduction of advanced analytics concepts, like predictive analytics or prescriptive analytics, or general statistics (if the user community isn't familiar with these concepts, already).
- Changes in the presentation of data, like:
    - Styling, colors, and other formatting choices for visuals.
    - The type of visualization.

    - How data is grouped or summarized (such as changing from different measures of central tendency, like average, median, or geometric mean).
- Changes in how content consumers interact with data (like connecting to a shared semantic model instead of exporting information for personal usage scenarios).

How you prepare change management plans and activities will depend on the types of change. To successfully and sustainably manage change, we recommend that you implement incremental changes.

Address change incrementally

Change management can be a significant undertaking. Taking an incremental approach can help you facilitate change in a way that's sustainable. To adopt an incremental approach, you identify the highest priority changes and break them into manageable parts, implementing each part with iterative phases and action plans.

The following steps outline how you can incrementally address change.

1. **Define what's changing**: Describe the change by outlining the before and after states. Clarify the specific parts of the process or situation that you'll change, remove, or introduce. Justify why this change is necessary, and when it should occur.
2. **Describe the impact of the change**: For each of these changes, estimate the business impact. Identify which processes, teams, or individuals the change affects, and how disruptive it will be for them. Also consider any downstream effects the change has on other dependent solutions or processes. Downstream effects might result in other changes. Additionally, consider how long the situation remained the same before it was changed. Changes to longer-standing processes tend to have a higher impact, as preferences and dependencies arise over time.
3. **Identify priorities**: Focus on the changes with the highest potential impact. For each change, outline a more detailed description of the changes and how it will affect people.
4. **Plan how to incrementally implement the change**: Identify whether any high-impact changes can be broken into stages or parts. For each part, describe how it might be incrementally implemented in phases to limit its impact. Determine whether there are any constraints or dependencies (such as when changes can be made, or by whom).
5. **Create an action plan for each phase**: Plan the actions you will take to implement and support each phase of the change. Also, plan for how you can mitigate disruption in high-impact phases. Be sure to include a rollback plan in your action plan, whenever possible.

> **Tip**
>
> Iteratively plan how you'll implement each phase of these incremental changes as part of your quarterly **tactical planning**.

When you plan to mitigate the impact of changes on Power BI adoption, consider the activities described in the following sections.

Effectively communicate change

Ensure that you clearly and concisely describe planned changes for the user community. Important communication should originate from the executive sponsor, or another leader with relevant authority. Be sure to communicate the following details.

- **What's changing**: What the situation is now and what it will be after the change.
- **Why it's changing**: The benefit and value of the change for the audience.
- **When it's changing**: An estimation of when the change will take effect.
- **Further context**: Where people can go for more information.
- **Contact information**: Who people should contact provide feedback, ask questions, or raise concerns.

Consider maintaining a history of communications in your centralized portal. That way, it's easy to find communications, timings, and details of changes after they've occurred.

> **Important**
>
> You should communicate change with sufficient advanced notice so that people are prepared. The higher the potential impact of the change, the earlier you should communicate it. If unexpected circumstances prevent advance notice, be sure to explain why in your communication.

Plan training and support

Changes to tools, processes, and solutions typically require training to use them effectively. Additionally, extra support might be required to address questions or respond to support requests.

Here are some actions you can take to plan for training and support.

- Centralize training and support by using a centralized portal. The portal can help organize discussions, collect feedback, and distribute training materials or

documentation by topic.
- Consider incentives to encourage self-sustaining support within a community.
- Schedule recurring office hours to answer questions and provide mentorship.
- Create and demonstrate end-to-end scenarios for people to practice a new process.
- For high-impact changes, prepare training and support plans that realistically assess the effort and actions needed to prevent the change from causing disruption.

> ⓘ **Note**
>
> These training and support actions will differ depending on the scale and scope of the change. For high-impact, large-scale changes (like transitioning from enterprise to managed self-service approaches to data and BI), you'll likely need to plan iterative, multi-phase plans that span multiple planning periods. In this case, carefully consider the effort and resources needed to deliver success.

Involve executive leadership

Executive support is critical to effective change management. When an executive supports a change, it demonstrates its strategic importance or benefit to the rest of the organization. This top-down endorsement and reinforcement is particularly important for high-impact, large-scale changes, which have a higher potential for disruption. For these scenarios, ensure that you actively engage and involve your executive sponsor to endorse and reinforce the change.

> ⊗ **Caution**
>
> Resistance to change from the executive leadership is often a warning sign that stronger **business alignment** is needed between the business and BI strategies. In this scenario, consider specific alignment sessions and change management actions with executive leadership.

Involve stakeholders

To effectively manage change, you can also take a bottom-up approach by engaging the stakeholders, who are the people the change affects. When you create an action plan to address the changes, identify and engage key stakeholders in focused, limited sessions. In this way you can understand the impact of the change on the people whose work will

be affected by the change. Take note of their concerns and their ideas for how you might lessen the impact of this change. Ensure that you identify any potentially unexpected effects of the change on other people and processes.

Handle resistance to change

It's important to address resistance to change, as it can have substantial negative impacts on adoption and productivity. When you address resistance to change, consider the following actions and activities.

- **Involve your executive sponsor**: The authority, credibility, and influence of the executive sponsor is essential to support change management and resolve disputes.
- **Identify blocking issues**: When change disrupts the way people work, this change can prevent people from effectively completing tasks in their regular activities. For such blocking issues, identify potential workarounds when you take into account the changes.
- **Focus on data and facts instead of opinions**: Resistance to change is sometimes due to opinions and preferences, because people are familiar with the situation prior to the change. Understand why people have these opinions and preferences. Perhaps it's due to convenience, because people don't want to invest time and effort in learning new tools or processes.
- **Focus on business questions and processes instead of requirements**: Changes often introduce new processes to address problems and complete tasks. New processes can lead to a resistance to change because people focus on what they miss instead of fully understanding what's new and why.

Additionally, you can have a significant impact on change resistance by engaging *promoters* and *detractors*.

Identify and engage promoters

Promoters are vocal, credible individuals in a user community who advocate in favor of a tool, solution, or initiative. Promoters can have a positive impact on adoption because they can influence peers to understand and accept change.

To effectively manage change, you should identify and engage promoters early in the process. You should involve them and inform them about the change to better utilize and amplify their advocacy.

> **Tip**

The promoters you identify might also be great candidates for your **champions network**.

Identify and engage detractors

Detractors are the opposite of promoters. They are vocal, credible individuals in a user community who advocate against a tool, solution, or initiative. Detractors can have a significant negative influence on adoption because they can convince peers that the change isn't beneficial. Additionally, detractors can advocate for alternative or solutions marked for retirement, making it more difficult to decommission old tools, solutions, or processes.

To effectively manage change, you should identify and engage detractors early in the process. That way, you can mitigate the potential negative impact they have. Furthermore, if you address their concerns, you might convert these detractors into promoters, helping your adoption efforts.

Tip

A common source of detractors is content owners for solutions that are going to be modified or replaced. The change can sometimes threaten these content owners, who are incentivized to resist the change in the hope that their solution will remain in use. In this case, identify these content owners early and involve them in the change. Giving these individuals a sense of ownership of the implementation will help them embrace, and even advocate in favor, of the change.

Questions to ask

Use questions like those found below to assess change management.

- Is there a role or team responsible for change management in the organization? If so, how are they involved in data and BI initiatives?
- Is change seen as an obstacle to achieving strategic success among people in the organization? Is the importance of change management acknowledged in the organization?

- Are there any significant promoters for data and BI solutions and processes in the user community? Conversely, are there any significant detractors?
- What communication and training efforts are performed to launch new data tools and solutions? How long do they last?
- How is change in the user community handled (for example, with new hires or promoted individuals)? What onboarding activities introduce these new individuals to existing solutions, processes, and policies?
- Do people who create Excel reports feel threatened or frustrated by initiatives to automate reporting with BI tools?
- To what extent do people associate their identities with the tools they use and the solutions they have created and own?
- How are changes to existing solutions planned and managed? Are changes planned, with a visible roadmap, or are they reactive? Do people get sufficient notification about upcoming changes?
- How frequently do changes disrupt existing processes and tools?
- How long does it take to decommission legacy systems or solutions when new ones become available? How long does it take to implement changes to existing solutions?
- To what extent do people agree with the statement *I am overwhelmed with the amount of information I am required to process*? To what extent do people agree with the sentiment *things are changing too much, too quickly*?

Maturity levels

An assessment of change management evaluates how effectively the organization can enact and respond to change.

The following maturity levels will help you assess your current state of change management, as it relates to data and BI initiatives.

Expand table

Level	State of change management
100: Initial	• Change is usually reactive, and it's also poorly communicated. • The purpose or benefits of change aren't well understood, and resistance to

Level	State of change management
	change causes conflict and disruption. • No clear teams or roles are responsible for managing change for data initiatives.
200: Repeatable	• Executive leadership and decision makers recognize the need for change management in data and BI projects and initiatives. • Some efforts are taken to plan or communicate change, but they're inconsistent and often reactive. Resistance to change is still common. Change often disrupts existing processes and tools.
300: Defined	• Formal change management plans or roles are in place. These plans include communication tactics and training, but they're not consistently or reliably followed. Change occasionally disrupts existing processes and tools. • Successful change management is championed by key individuals that bridge organizational boundaries.
400: Capable	• Empathy and effective communication are integral to change management strategies. • Change management efforts are owned by particular roles or teams, and effective communication results in a clear understanding of the purpose and benefits of change. Change rarely interrupts existing processes and tools.
500: Efficient	• Change is an integral part of the organization. People in the organization understand the inevitability of change, and see it as a source for momentum instead of disruption. Change almost never unnecessarily interrupts existing processes or tools. • Systematic processes address change as a challenge of people and not processes.

Related content

In the next article in the Microsoft Fabric adoption roadmap series, in conclusion, learn about adoption-related resources that you might find valuable.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft Fabric adoption roadmap conclusion

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Microsoft Fabric adoption roadmap* series of articles. For an overview of the series, see Microsoft Fabric adoption roadmap.

This article concludes the series on Microsoft Fabric adoption. The strategic and tactical considerations and action items presented in this series will assist you in your analytics adoption efforts, and with creating a productive data culture in your organization.

This series covered the following aspects of Fabric adoption.

- Adoption introduction
- Adoption maturity levels
- Data culture
- Executive sponsorship
- Business alignment
- Content ownership and management
- Content delivery scope
- Center of Excellence
- Governance
- Mentoring and enablement
- Community of practice
- User support
- System oversight
- Change management

The rest of this article includes suggested next actions to take. It also includes other adoption-related resources that you might find valuable.

Next actions to take

It can be overwhelming to decide where to start. The following series of steps provides a process to help you approach your next actions.

1. **Learn**: First, read this series of articles end-to-end. Become familiar with the strategic and tactical considerations and action items that directly lead to successful analytics adoption. They'll help you to build a data culture in your organization. Discuss the concepts with your colleagues.
2. **Assess current state**: For each area of the adoption roadmap, assess your current state. Document your findings. Your goal is to have full clarity on where you're now so that you can make informed decisions about what to do next.
3. **Clarify your strategic goals**: Ensure that you're clear on what your organization's goals are for adopting Fabric. Confirm that your adoption and data culture goals align with your organization's broader strategic goals for the use of data, analytics, and business intelligence (BI) in general. Focus on what your immediate strategy is for the next 3-12 months. For more information about defining your goals, see the strategic planning article.
4. **Prioritize**: Clarify what's most important to achieve in the next 12-18 months. For instance, you might identify specific user enablement or risk reduction areas that are a higher priority than other areas. Determine which advancements in maturity levels you should prioritize first. For more information about defining your priorities, see the strategic planning article.
5. **Identify future state**: For each area of the roadmap, identify the gaps between what you want to happen (your future state) and what's happening (your current state). Focus on the next 12-18 months for identifying your desired future state.
6. **Customize maturity levels**: Using the information you have on your strategy and future state, customize the maturity levels for each area of the roadmap. Update or delete the description for each maturity level so that they're realistic, based on your goals and strategy. Your current state, priorities, staffing, and funding will influence the time and effort it will take to advance to higher maturity levels.
7. **Define measurable objectives**: Create KPIs (key performance indicators) or OKRs (objectives and key results) to define specific goals for the next quarter. Ensure that the objectives have clear owners, are measurable, time-bound, and achievable. Confirm that each objective aligns with your strategic BI goals and priorities.
8. **Create tactical plans**: Add specific action items to your project plan. Action items will identify who will do what, and when. Include short, medium, and longer-term (backlog) items in your project plan to make it easy to track and reprioritize.
9. **Track action items**: Use your preferred project planning software to track continual, incremental progress of your action items. Summarize progress and status every quarter for your executive sponsor.
10. **Adjust**: As new information becomes available—and as priorities change—reevaluate and adjust your focus. Reexamine your strategic goals, objectives, and action items once a quarter so you're certain that you're focusing on the right actions.

11. **Celebrate**: Pause regularly to appreciate your progress. Celebrate your wins. Reward and recognize people who take the initiative and help achieve your goals. Encourage healthy partnerships between IT and the different areas of the business.
12. **Repeat**: Continue learning, experimenting, and adjusting as you progress with your implementation. Use feedback loops to continually learn from everyone in the organization. Ensure that continual, gradual, improvement is a priority.

A few important key points are implied within the previous suggestions.

- **Focus on the near term**: Although it's important to have an eye on the big picture, we recommend that you focus primarily on the next quarter, next semester, and next year. It's easier to assess, plan, and act when you focus on the near term.
- **Progress will be incremental**: Changes that happen every day, every week, and every month add up over time. It's easy to become discouraged and sense a lack of progress when you're working on a large adoption initiative that takes time. If you keep track of your incremental progress, you'll be surprised at how much you can accomplish over the course of a year.
- **Changes will continually happen**: Be prepared to reconsider decisions that you make, perhaps every quarter. It's easier to cope with continual change when you expect the plan to change.
- **Everything correlates together**: As you progress through each of the steps listed above, it's important that everything's correlated from the high-level strategic organizational objectives, all the way down to more detailed action items. That way, you'll know that you're working on the right things.

Power BI implementation planning

Successfully implementing analytics throughout the organization requires deliberate thought and planning. The Power BI implementation planning series of articles, which is a work in progress, is intended to complement the Microsoft Fabric adoption roadmap. It includes key considerations, actions, decision-making criteria, recommendations, and it describes implementation patterns for important common usage scenarios.

Power BI adoption framework

The Power BI adoption framework describes additional aspects of *how* to adopt Power BI in more detail. The original intent of the framework was to support Microsoft partners with a lightweight set of resources for use when helping their customers deploy and adopt Power BI.

The framework can augment this Microsoft Fabric adoption roadmap series. The roadmap series focuses on the *why* and *what* of adopting Fabric, more so than the *how*.

> **Note**
>
> When completed, the Power BI implementation planning series (described in the previous section) will replace the Power BI adoption framework.

Microsoft's BI transformation

Consider reading about Microsoft's journey and experience with driving a data culture. This article describes the importance of two terms: *discipline at the core* and *flexibility at the edge*. It also shares Microsoft's views and experience about the importance of establishing a COE.

Power Platform adoption

The Power Platform team has an excellent set of adoption-related content. Its primary focus is on Power Apps, Power Automate, and Power Virtual Agents. Many of the ideas presented in this content can be applied to Power BI also.

The Power CAT Adoption Maturity Model, published by the Power CAT team, describes repeatable patterns for successful Power Platform adoption.

The Power Platform Center of Excellence Starter Kit is a collection of components and tools to help you develop a strategy for adopting and supporting Microsoft Power Platform.

The Power Platform adoption best practices includes a helpful set of documentation and best practices to help you align business and technical strategies.

The Power Platform adoption framework is a community-driven project with excellent resources on adoption of Power Platform services at scale.

Microsoft 365 and Azure adoption

You might also find useful adoption-related guidance published by other Microsoft technology teams.

- The Maturity Model for Microsoft 365 provides information and resources to use capabilities more fully and efficiently.

- Microsoft Learn has a learning path for using the Microsoft service adoption framework to drive adoption in your enterprise.
- The Microsoft Cloud Adoption Framework for Azure is a collection of documentation, implementation guidance, best practices, and tools to accelerate your cloud adoption journey.

A wide variety of other adoption guides for individual technologies can be found online. A few examples include:

- Microsoft Teams adoption guide.
- Microsoft Security and Compliance adoption guide.
- SharePoint Adoption Resources.

Industry guidance

The Data Management Book of Knowledge (DMBOK2) is a book available for purchase from DAMA International. It contains a wealth of information about maturing your data management practices.

> **Note**
>
> The additional resources provided in this article aren't required to take advantage of the guidance provided in this Fabric adoption series. They're reputable resources should you wish to continue your journey.

Partner community

Experienced partners are available to help your organization succeed with adoption initiatives. To engage a partner, visit the Power BI partner portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning

Article • 12/30/2024

In this video, watch Matthew introduce you to the Power BI implementation planning series of articles.
https://learn-video.azurefd.net/vod/player?id=959505f6-668e-4b27-b09f-ebab8d9e6a44&locale=en-us&embedUrl=%2Fpower-bi%2Fguidance%2Fpowerbi-implementation-planning-introduction

Successfully implementing Power BI throughout the organization requires deliberate thought and planning. The Power BI implementation planning series provides you with key considerations, actions, decision-making criteria, and tactical recommendations. The articles in this series cover key subject areas when implementing Power BI, and they describe patterns for common usage scenarios.

Subject areas

When you implement Power BI, there are many subject areas to consider. The following subject areas form part of the Power BI implementation planning series:

- BI strategy
- Tenant setup
- User tools and devices
- Subscriptions, licenses, and trials
- Tenant administration
- Workspaces
- Content lifecycle management
- Content distribution and sharing
- Security
- Information protection and data loss prevention
- Data gateways
- Integration with other services
- Auditing and monitoring
- Adoption tracking

Usage scenarios

The series includes usage scenarios that illustrate different ways that creators and consumers can deploy and use Power BI:

- Content collaboration and delivery
- Self-service BI
- Content management and deployment
- Embedding and hybrid

Purpose

This series complements the Fabric adoption roadmap, which describes considerations for successful Microsoft Fabric and Power BI adoption and a healthy data culture. Power BI implementation planning guidance that correlates with the adoption roadmap goals will be added to this series.

Recommendations

To set yourself up for success, we recommend that you work through the following steps:

1. Read the complete Fabric adoption roadmap, familiarizing yourself with each roadmap subject area. Assess your current state of Fabric adoption, and gain clarity on the data culture objectives for your organization.
2. Explore Power BI implementation planning articles that are relevant to you. Start with the Power BI usage scenarios, which convey how you can use Power BI in diverse ways. Be sure to understand which usage scenarios apply to your organization, and by whom. Also, consider how these usage scenarios might influence the implementation strategies you decide on.
3. Read the articles for each of the subject areas that are listed above. You might choose to initially do a broad review of the contents from top to bottom. Or you might choose to start with subject areas that are your highest priority. Carefully review the key decisions and actions that are included for each topic (at the end of each section). We recommend that you use them as a starting point for creating and customizing your plan.
4. When necessary, refer to Power BI documentation and Fabric documentation for details on specific topics.

Target audience

The intended audience of this series of articles might be interested in the following outcomes:

- Identifying areas to improve or strengthen their Power BI implementation.

- Increasing their ability to efficiently manage and securely deliver Power BI content.
- Planning the implementation of Power BI within their organization.
- Increasing their organization's return on investment (ROI) in Power BI.

This series is certain to be helpful for organizations that are in their early stages of a Power BI implementation or are planning an expanded implementation. It might also be helpful for those who work in an organization with one or more of the following characteristics:

- Power BI has pockets of viral adoption and success in the organization, but it's not consistently well-managed or purposefully governed.
- Power BI is deployed with some meaningful scale, but there are many unrealized opportunities for improvement.

> **Tip**
>
> Some knowledge of Power BI and general business intelligence concepts is assumed. To get the most from this content, we recommend that you become familiar with the **Fabric adoption roadmap** first.

Acknowledgments

The Power BI implementation planning articles are written by Melissa Coates, Kurt Buhler, and Peter Myers. Matthew Roche, from the Fabric Customer Advisory Team, provides strategic guidance and feedback to the subject matter experts.

Related content

In the next article in this series, learn about usage scenarios that describe how you can use Power BI in diverse ways.

Other helpful resources include:

- Fabric adoption roadmap
- Power BI migration overview

Experienced Power BI partners are available to help your organization succeed with the migration process. To find a Power BI partner, visit the Microsoft Power BI partners portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: BI strategy

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces the business intelligence (BI) strategy series of articles. The BI strategy series is targeted at multiple audiences:

- **Executive leadership**: Individuals who are responsible for defining organizational objectives and strategies, like the Microsoft Fabric or Power BI executive sponsor or a Chief Executive Officer (CEO), Chief Information Officer (CIO), or Chief Data Officer (CDO).
- **BI and analytics directors or managers**: Decision makers who are responsible for overseeing the BI program and BI strategic planning.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for tactical planning, measuring, and monitoring progress toward the BI key results. These teams also plan key initiatives and solutions.
- **Subject matter experts (SMEs) and content owners and creators**: The teams and individuals that champion analytics in a team or department and conduct BI solution planning. These teams and individuals are responsible for supporting the strategy and data needs of their business area.

Defining your BI strategy is essential to get the most business value from data and analytics. Having a clearly defined BI strategy is important to ensure efforts are aligned with organizational priorities. In some circumstances, it's particularly important.

We recommend that you pay special attention to these articles if your organization is:

- **Migrating to, or implementing, Fabric or Power BI for the first time**: A clear BI strategy is crucial to the successful implementation of any new platform or tool.
- **Experiencing significant growth of Fabric or Power BI usage**: A BI strategy brings clarity and structure to organic adoption, helping to enable users while mitigating risk.
- **Seeking to become data-driven or achieve digital transformation**: A BI strategy is critical to modernizing your organization and helping you to achieve a competitive

advantage.

- **Experiencing significant business or technological change**: Planning your BI strategy ensures that your organization can use change as momentum and not as an obstacle.
- **Reevaluating your business strategy**: Your business strategy should influence your BI strategy, which in turn can lead to changes in your business strategy. All strategies should be in alignment in order to achieve your organizational objectives.

In short, this series of articles is about defining a BI strategy. It describes what a BI strategy is, why it's important, and how you can plan your BI strategy. The articles in this series are intended to complement the Fabric adoption roadmap.

Become data-driven with a BI strategy

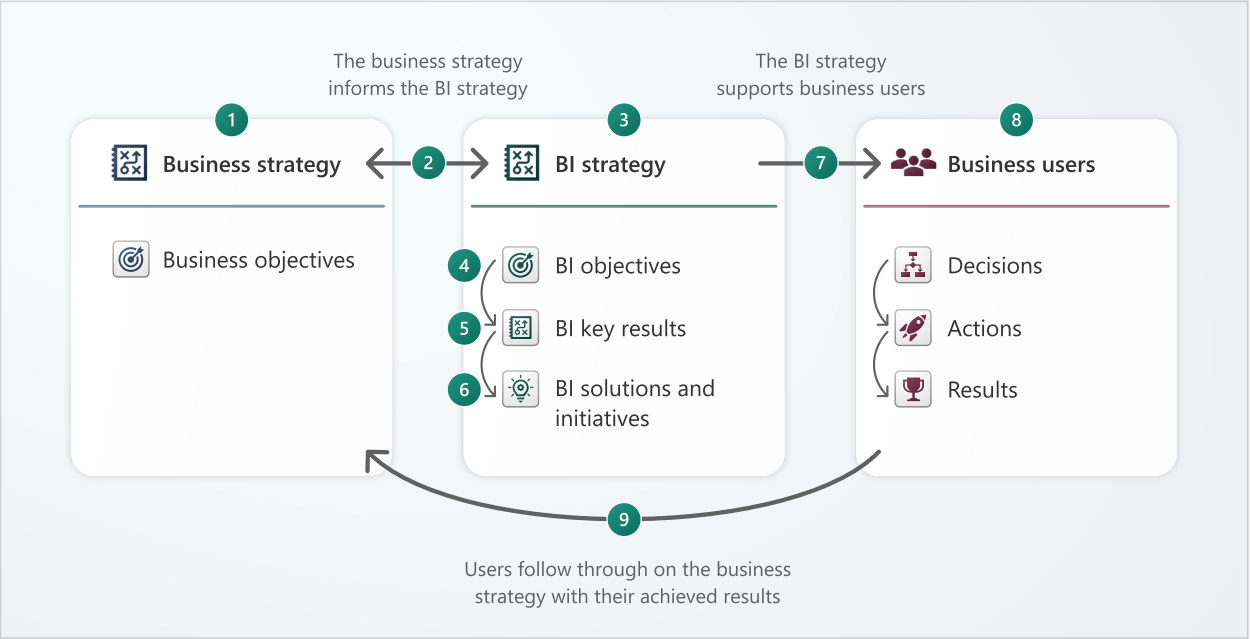
Successful adoption and implementation of analytics solutions helps an organization meet their business objectives. To achieve successful adoption and implementation, you need a *BI strategy*. A BI strategy might sometimes be described as an *analytics strategy* or *becoming data driven*.

A BI strategy is a plan to implement, use, and manage data and analytics to better enable your users to meet their business objectives. An *effective* BI strategy ensures that data and analytics support your business strategy.

Relationship between BI strategy and business strategy

Your business strategy should directly inform your BI strategy. As your business objectives evolve, your BI processes and tools might also need to evolve, especially as new data needs arise. New opportunities and insights learned from BI solutions can also lead to changes to your business strategy. Understanding and supporting the relationship between your business and BI strategies is essential in order to make valuable BI solutions, and to ensure that people use them effectively.

The following diagram depicts how a BI strategy supports the business strategy by enabling business users.



The diagram depicts the following concepts.

Expand table

Item	Description
1	The business strategy describes how the organization will achieve its business objectives.
2	The business strategy directly informs the BI strategy. The primary purpose of the BI strategy is to support—and potentially inform—the business strategy.
3	The BI strategy is a plan to implement, use, and manage data and analytics.
4	BI objectives define how BI will support the business objectives. BI objectives describe the desired future state of the BI environment.
5	To make progress toward your BI objectives, you identify and describe BI key results that you want to achieve in a specific time period. These key results describe paths to your desired future state.
6	To achieve your BI key results, you plan and implement BI solutions and initiatives. A solution might be developed by a central IT or BI team, or by a member of the community of practice as a self-service solution.
7	The purpose of BI solutions and initiatives is to enable business users to achieve their key results.
8	Business users use BI solutions and initiatives to make informed decisions that lead to effective actions.
9	Business users follow through on the business strategy with their achieved results. They achieve these results by taking the right actions at the right time, which is made possible in part by an effective BI strategy.

> **Note**
>
> In the objectives and key results (OKRs) framework, *objectives* are clear, high-level descriptions of what you want to achieve. In contrast, *key results* are specific, achievable outcomes to measure progress toward one of your objectives.
>
> Further, *initiatives* or *solutions* are processes or tools built to help you achieve one or more key results. Solutions address specific data needs for users. A solution can take many forms, such as a data pipeline, a data lakehouse, or a Power BI semantic model or report.
>
> For more about OKRs, see Get to know OKRs (Microsoft Viva Goals).

Consider the following, high-level example for a hypothetical organization.

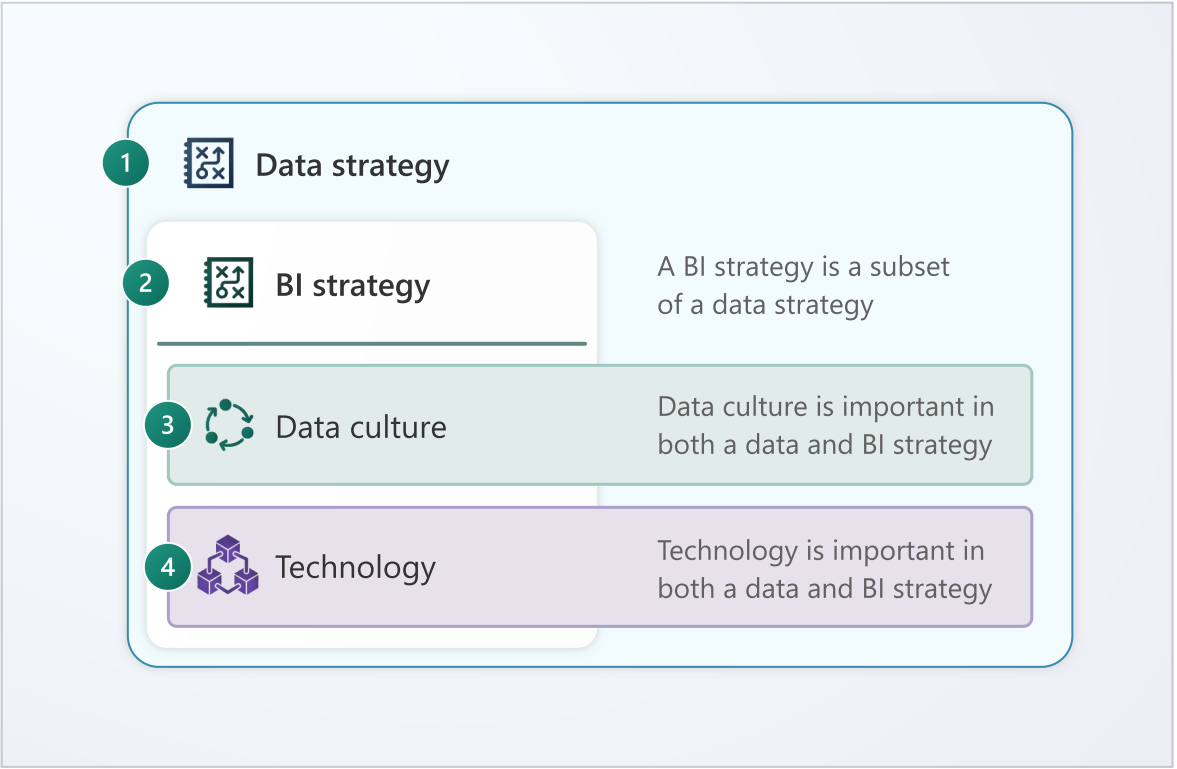
Expand table

Area	Example
Business strategy	The organizational objective is to improve customer satisfaction and reduce customer churn. One business strategy to achieve this objective is to reduce the number of late customer deliveries.
BI strategy	• **BI objective**: To support the business strategy, the BI objective is to improve the effectiveness of orders and deliveries reporting. • **BI key results**: To achieve the BI objective, the organization defines specific BI key results for the quarter. One such key result is to reduce the time to produce reports about on-time delivery by 80%, so that reports are available daily, instead of weekly. Another key result is to provide combined inventory and orders data for the largest distribution center. Demand planners can use inventory data to improve delivery planning. • **BI solutions and initiatives**: To achieve these BI key results, the organization plans BI solutions and initiatives, like implementing automated data pipelines, and a consolidated data lakehouse that stores business-ready orders and inventory data to support reporting and analytics. They enact a training program to enable users to make the most of the newly available data.
Business users	Enabled by these BI solutions and initiatives, business users can more effectively identify and mitigate potential late deliveries. These solutions result in fewer late deliveries and improved customer satisfaction, allowing the organization to achieve progress toward its business objectives.

Relationship between BI strategy and data strategy

Your BI strategy describes how successful Fabric adoption and Power BI implementation will deliver business value to your organization. However, a BI strategy transcends tools and technologies. While your BI strategy might start small, it can grow to encompass all of your analytical data, tools, and processes when you experience success. Furthermore, the concepts in a BI strategy are also important in a broader *data strategy*. While a BI strategy is about the use of data and tools for analytical purposes, a data strategy is concerned with the wider management and use of data within the organization. Thus, your BI strategy is a subset of your data strategy, as they share many related concepts.

The following diagram depicts how a BI strategy is a subset of a data strategy, and how they share concepts related to data culture and technology.



The diagram depicts the following concepts.

Expand table

Item	Description
1	A data strategy describes the focus areas and objectives for the wider use and management of data in an organization. A data strategy encompasses more than only BI.
2	The BI strategy is a subset of a data strategy.
3	Data culture is important in both a BI strategy and a data strategy. Different data culture areas describe a vision for behaviors, values, and processes that enable people to work effectively with data. An example of a data culture area is data literacy.

Item	Description
4	Technology is important in both a BI strategy and a data strategy. Different technical areas support the business data needs and use cases. An example of a technical area is data visualization.

A BI strategy can encompass many data culture and technical areas. However, when planning your BI strategy, you should be cautious not to attempt to address too many of these areas at first. A successful BI strategy starts small. It focuses on a few focus areas and broadens scope over time, ensuring consistent progress. Later, as you experience sustainable success with your BI strategy, it can incrementally evolve to encompass more areas.

> **ⓘ Important**
>
> This series of BI strategy articles focuses on the Power BI workload in Fabric. However, planning a BI strategy is a technology-agnostic exercise. As such, the concepts described in the articles can apply irrespective of your chosen BI tools and technologies.

Defining a BI strategy

There are many ways to define a BI strategy. Typically, when you define a BI strategy, you begin by identifying the focus areas for which you describe your BI objectives. Based upon these objectives, you define time-bound, prioritized actions in key results. To achieve these key results, you build solutions and enact specific key initiatives. You then incrementally scale your BI strategy to encompass more focus areas and additional objectives as you experience success.

The following diagram depicts how you can define your BI strategy at three different planning levels, as depicted in the following diagram.

The diagram depicts the following three planning levels.

Expand table

Item	Description
1	**Strategic planning**: You begin by defining your strategic BI focus areas and objectives, and how they support your business strategy. These BI objectives are high-level descriptions of what you want to achieve and why.
2	**Tactical planning**: You then identify your specific BI key results. These key results are specific, measurable, short-term actions that describe how you'll make progress toward your long-term, strategic BI objectives.
3	**Solution planning**: The BI solutions and initiatives that you create should be a direct result of tactical planning. These solutions enable you to achieve your BI key results and make progress toward your BI objectives.

Important

Defining a BI strategy requires prioritization, planning, and active involvement from many teams and individuals across your organization.

BI strategy example

The following high-level, hypothetical example explains how you can transition from business objectives to BI objectives. It then explains how to transition from BI objectives

to key results, and then to BI solutions and initiatives.

Business objectives and strategy

In this example, the organization has set an objective to increase sales effectiveness. One strategy the business uses to achieve this objective is to sell more high-margin products to its top customers.

BI focus areas and objectives

To achieve the business strategy, the organization wants the salespeople to adopt data-driven decision making. To this end, the BI team works with the sales teams to understand their data needs, and to define long-term, strategic BI focus areas and objectives.

In this example, the BI focus areas and objectives are:

- **Data literacy**: Improve the ability of the salespeople to make decisions based on data and report visualizations.
- **Content ownership**: Clarify who owns the data and reporting items for different use cases.
- **Mentoring and user enablement**: More effectively enable the salespeople with the skills and tools to answer questions with data.
- **Governance**: More effectively balance governance risk and enablement of the sales teams.
- **Data engineering**: Create a unified view of sales and profitability data for analytics.

> **Note**
>
> In this example, many other factors might be important. However, the organization has identified these particular focus areas and objectives to support the business strategy.

Key results

To achieve their BI objectives, the BI team conducts tactical planning to identify and describe their short-term key results. The BI team creates an introductory data literacy program for the salespeople. Also, the BI team drafts a user enablement plan and an accountability plan for salespeople who want to perform self-service analytics. These

plans allow the salespeople to request access to data after they've completed specific training materials and signed a self-service user acknowledgment.

In this example, the BI key results in the first quarter are:

- **Data literacy**: Ensure that 90 percent of the salespeople complete the data literacy program.
- **Content ownership**: Identify one champion in each sales team, and train the champions to connect to centralized semantic models and create their own reports.
- **Mentoring and user enablement**: Create a centralized portal in the first quarter to share training resources, template files, and host weekly office hours Q&A sessions.
- **Governance**: Reduce export activities by salespeople from centralized reports by 20%.
- **Data engineering**: Choose an architecture for consolidating sales and profitability data.
- **Data security**: Define and implement the data security rules for the sales and profitability data that salespeople will use.
- **Information protection and data loss prevention (DLP)**: Define how content creators should endorse content by promoting or certifying data items. Conduct an investigation into whether the organization needs sensitivity labels and DLP policies.

Key initiatives and solutions

To achieve its key results, the organization aims to enact the following key initiatives, or design and deploy the following BI solutions.

- Central BI teams will design and start a proof of concept for a medallion lakehouse architecture to store the sales and profitability data.
- Central BI teams will publish an enterprise semantic model as a Power BI semantic model that includes all data required for central reporting and key self-service reporting scenarios.
- Central BI teams will draft a prototype tenant-wide monitoring solution for user activities based on data from the Power BI activity log.
- Security rules applied to the Power BI semantic model enforce that salespeople can only access data for their assigned customers.
- Central BI teams will create central reports that show aggregate sales and profitability across regions and product groups. These central reports will support more sophisticated analysis by using interactive visualizations.

> **Note**
>
> This example describes a simple scenario for the purpose of explaining the three planning levels of a BI strategy. In reality, your strategic BI objectives, key results, and key initiatives and solutions are likely to be more complex.

Iteratively plan a BI strategy

Your BI strategy should evolve as you scale and as your organization experiences change. For these reasons, planning your BI strategy is a continuous, iterative process.

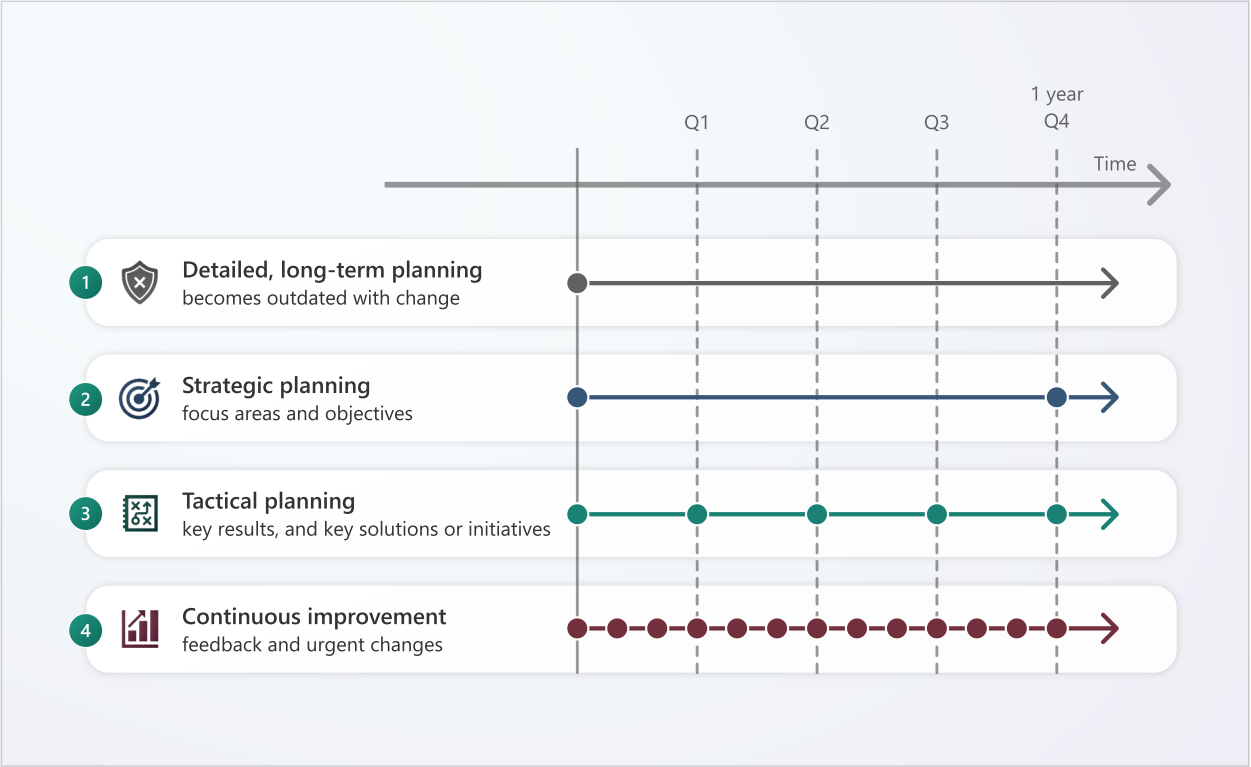
Iteratively planning your BI strategy is beneficial for two reasons.

- **Incremental progress**: Define your BI strategy by focusing on your focus areas and breaking them into manageable parts. You can implement these parts in phases and complete them incrementally over multiple continuous improvement cycles. With each cycle, you can evaluate progress and lesson learned to sustainably grow your strategy. In contrast, an all-in approach can become overwhelming, running out of steam before it produces value.
- **Overcome change**: Keep pace with changes to technology and your business strategy. Iterative planning and implementation phases can help your strategy remain relevant to business data needs. In contrast, detailed, multi-year strategic plans can quickly become outdated.

It's unrealistic to expect all-encompassing, long-term planning to survive beyond 12-18 months. For instance, attempting to create an exhaustive three to five-year plan can result in over-investment, a failure to keep up with change, and an inability to support changes in your business strategy. Instead, you should define and operationalize your strategies by using iterative approaches, with achievable outcomes in a maximum period of 18 months.

There are many ways to iteratively plan your BI strategy. A common approach is to schedule planning revisions over periods that align with existing planning processes in the organization.

The following diagram depicts recommendations for how you can schedule planning revisions.



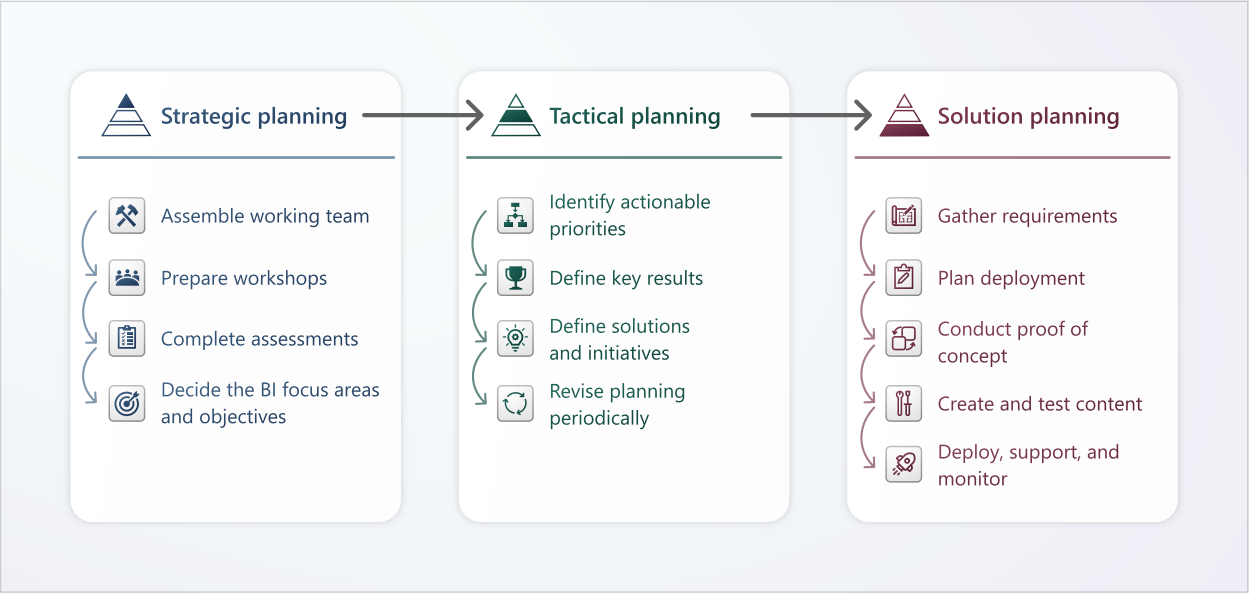
The diagram depicts how you can iteratively structure your BI strategy planning the following concepts.

Expand table

Item	Description
1	**Avoid detailed, long-term planning**: Long-term, detailed plans can become outdated as technology and business priorities change.
2	**Strategic planning (every 12-18 months)**: This high-level planning focuses on aligning business objectives and BI objectives. It's valuable to align this strategic planning with other annualized business processes, like budgeting periods.
3	**Tactical planning (every 1-3 months)**: Monthly or quarterly planning sessions focus on evaluating defining the specific, actionable, key results that are time-bound to the planning period. This planning should take iterative business feedback and changes in business or technology into account.
4	**Continuous improvement (every month)**: Monthly sessions focus on feedback and urgent changes that impact ongoing planning. If necessary, decision makers can make decisions, take corrective action, or influence ongoing planning.

How to plan a BI strategy

This series of articles presents a structured framework that helps you to plan the three levels of your BI strategy, as depicted in the following diagram.



The diagram shows three levels of BI strategy planning, which are each described in separate articles. We recommend that you read these articles in the following order.

1. **BI strategic planning**: This article describes how you can form a working team to lead the initiative to define your BI strategy. The working team prepares workshops with key stakeholders to understand and document the business strategy. The team then assesses the effectiveness of BI in supporting the business strategy. This assessment helps to define the strategic BI focus areas and objectives. After strategic planning, the working team proceeds to tactical planning.
2. **BI tactical planning**: This article describes how the working team can identify measurable, time-bound key results to achieve the BI objectives. As part of these key results, the working team creates a prioritized backlog of BI solutions and key initiatives. Finally, the working team commits to revise tactical planning every quarter. After tactical planning, you proceed to solution planning.
3. **BI solution planning**: This article describes how you design and build BI solutions that support the BI key results. You first assemble a project team that's responsible for a solution in the prioritized solution backlog. The project team then gathers requirements to define the solution design. Next, it plans for deployment and conducts a proof of concept (POC) of the solution to validate assumptions. If the POC is successful, the project team creates and tests content with iterative cycles that gradually onboard the user community. When ready, the project team deploys the solution to production, supporting and monitoring it as needed.

> **Tip**
>
> Before you read the BI strategy articles, we recommend that you're already familiar with the **Fabric adoption roadmap**. The adoption roadmap describes

considerations to achieve Fabric adoption and a healthy data culture. These BI strategy articles build upon the adoption roadmap.

Related content

In the next article in this series, learn about BI strategic planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: BI strategic planning

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to define your business intelligence (BI) focus areas and objectives through strategic planning. It's primarily targeted at:

- **BI and analytics directors or managers**: Decision makers who are responsible for overseeing the BI program and BI strategic planning.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for tactical planning, and for measuring and monitoring progress toward the BI goals.
- **Subject matter experts (SMEs) and content owners and creators**: The teams and individuals that champion analytics in a team or department and conduct BI solution planning. These teams and individuals are responsible for representing the strategy and data needs of their business area when you define the BI strategy.

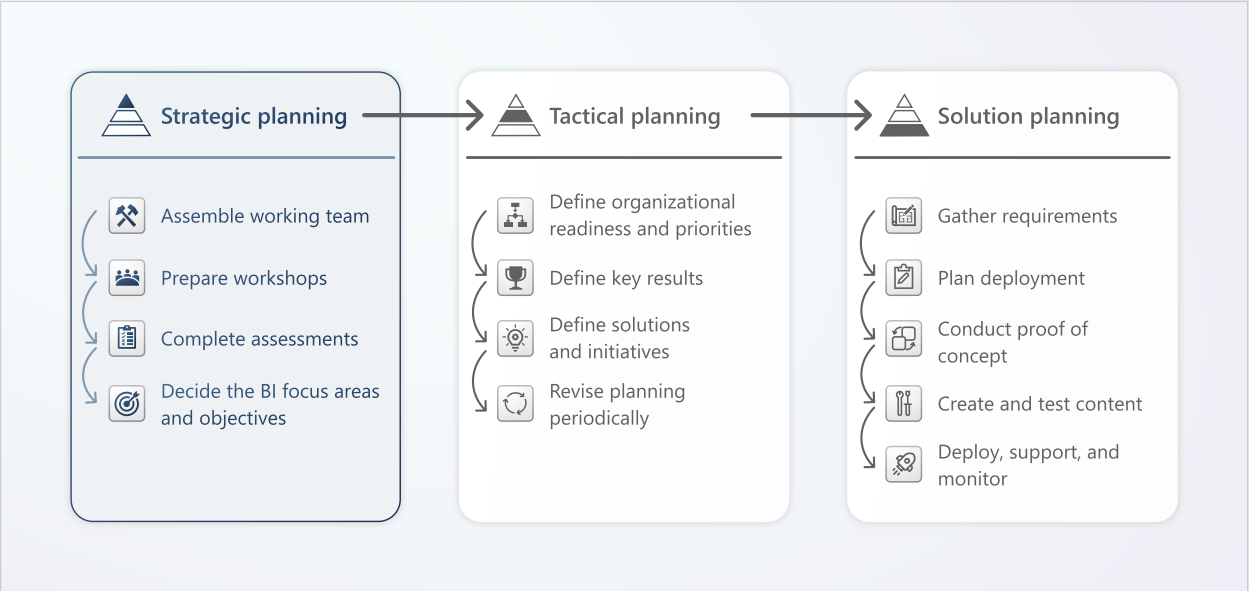
A BI strategy is a plan to implement, use, and manage data and analytics. As described in the BI strategy overview article, your BI strategy is a subset of your data strategy. It supports your business strategy by enabling business users to make decisions and take actions by using data and BI solutions more effectively.

In short, this article describes how you can perform strategic planning to define the focus areas and objectives of your BI strategy.

> **Note**
>
> In the objectives and key results (OKRs) framework, *objectives* are clear, high-level descriptions of what you want to achieve. In contrast, *key results* are specific, achievable outcomes to measure progress toward one of your objectives.
>
> Further, *initiatives* or *solutions* are processes or tools built to help you achieve one or more key results. Solutions address specific data needs for users. A solution can take many forms, such as a data pipeline, a data lakehouse, or a Power BI semantic model or report.

For more about OKRs, see Get to know OKRs (Microsoft Viva Goals).

The following high-level diagram depicts how to conduct BI strategic planning.



You take the following steps to define your strategic BI focus areas and objectives.

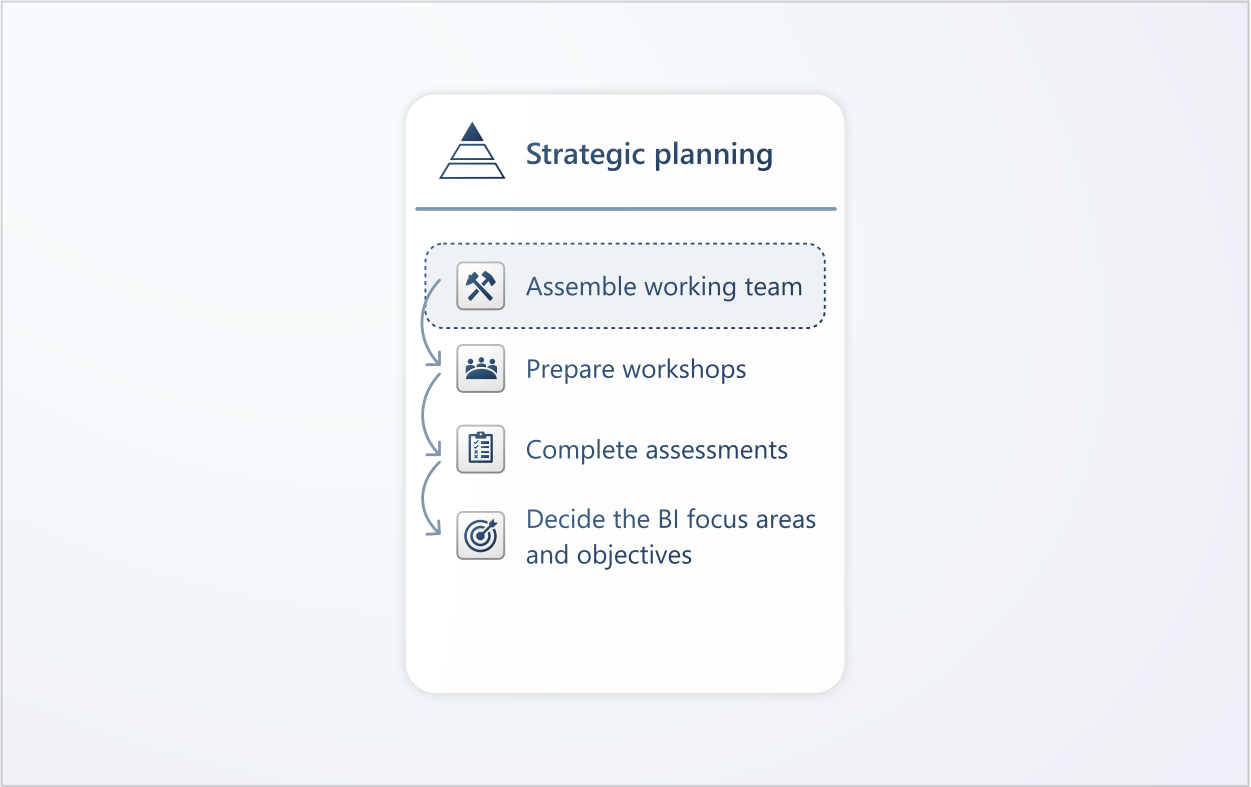
Expand table

Step	Description
1	Establish a working team to lead the BI strategy initiative.
2	Establish business alignment by conducting research and workshops to gather information about the business objectives and data needs, and also existing BI solutions and initiatives.
3	Complete a current state assessment by running a series of strategic planning workshops with key stakeholders.
4	Use the assessments and stakeholder input to decide on the strategic BI focus areas and objectives.

This article describes each step of the BI strategic planning process.

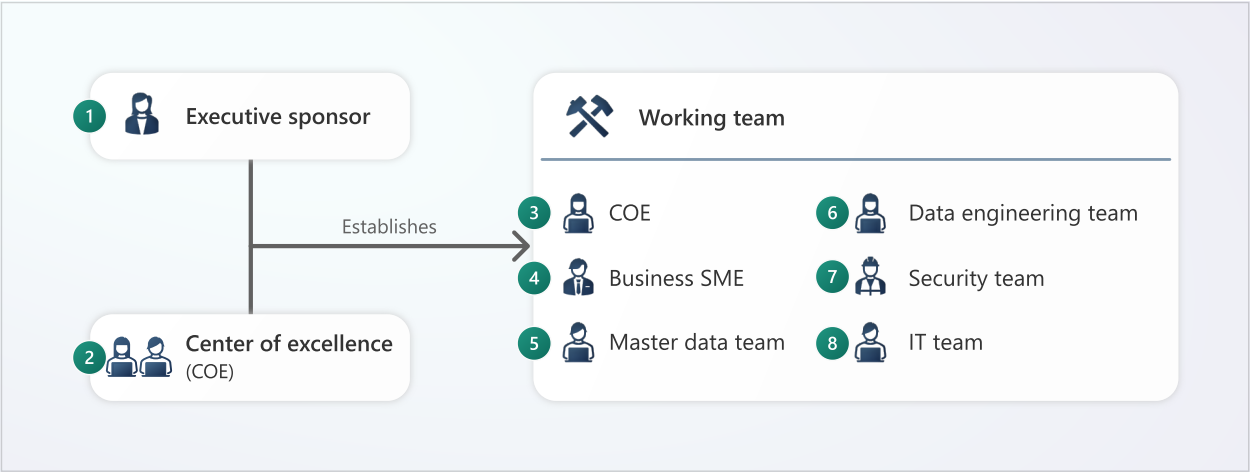
Step 1: Assemble a working team

Your first step when defining your BI strategy is to establish a *working team*. A working team leads the initiative to describe and plan the BI strategy. It's a cross-functional group of experts that's enabled by the support of the executive sponsor. The group should have a deep understanding of technical and business processes across the organization.



Ideally, the working team should represent each department, business unit, and region that's in scope for the initiative.

The following diagram depicts the following roles that appoint the working team members, and the types of members in a working team.



The diagram depicts the following roles.

Expand table

Item	Description
1	The executive sponsor typically provides top-down objectives and support of the working team, including funding. The executive sponsor might also appoint the working team members together with the Center of Excellence (COE).

Item	Description
2	A COE or central BI team confers with the executive sponsor to identify and appoint working team members. The COE might also provide guidance to the working team to support their activities.
3	COE members form part of the working team. They're responsible for using their BI expertise to drive BI information gathering and complete the current state assessments.
4	Business SMEs form part of the working team. They represent the interests of their department or business unit. SMEs are responsible for driving business strategy information gathering.
5	Functional team members, like those from a master data team, can form part of the working team. They're responsible for clarifying strategically important processes during information gathering.
6	Technical team members, like those from a data engineering team, can form part of the working team. They're responsible for clarifying strategically important systems during information gathering.
7	Security team members form part of the working team. They're responsible for identifying and clarifying compliance, security, and privacy requirements during information gathering.
8	Other IT team members can form part of the working team. They might identify other technical requirements related to areas such as networking or application management.

> **Note**
>
> Not all roles depicted in the diagram have to be present in the working team. Involve roles that are relevant for the scope and scale of your BI strategy initiative.

> **Important**
>
> Defining the BI strategy is a significant undertaking. It's important that working team members understand what's expected of them, and that they have the resources and time to fulfill their role. An engaged executive sponsor can help by clarifying focus areas and ensuring that all required resources are available.

Working team members are typically appointed and guided by an executive sponsor of BI and analytics, like the Power BI executive sponsor. Identifying and engaging an executive sponsor is the first step of a BI strategy initiative.

Identify and engage an executive sponsor

A key role of the executive sponsor is to help formulate strategic BI focus areas and objectives. The executive sponsor is an individual in a position of senior, strategic leadership who has an invested stake in BI efforts and the BI strategy. They provide top-down guidance and reinforcement by regularly promoting, motivating, and investing in the BI strategy.

In addition to the many activities listed in the adoption roadmap, an executive sponsor plays a key role in BI strategic planning by:

- **Supporting the working team and COE**: The executive sponsor takes a leading role in defining the BI strategy by providing top-down guidance and reinforcement.
- **Allocating resources**: They confirm funding, staffing, roles, and responsibilities for the working team.
- **Advocating for the initiative**: They advance the strategic initiative by:
  - Legitimizing working team activities, particularly when the working team faces resistance to change.
  - Promoting the BI strategy initiative with announcements or public endorsement.
  - Motivating action and change to progress the BI strategy initiative.
  - Representing the working team and sharing the BI strategic plan among C-level executives to obtain executive feedback.
- **Making strategic decisions**: They make decisions about what should be the focus areas, objectives, and key results.

> **Tip**
>
> Before assembling the working team, you should first identify and engage an **executive sponsor**. Work through **this checklist** to ensure that you take the necessary actions to ensure a sufficiently engaged executive sponsor.

Decide on the scope of the BI initiative

Because the working team contains members from different business areas, the composition of the working team will depend on the scope of your BI initiative. Typically, a BI strategy encompasses many areas of an organization. However, you should refine this scope to define the specific areas it should address. You might limit the scope of your BI strategy initiative for two reasons.

- **Practical reasons**: A successful BI strategy starts small and simple, achieving incremental growth as you experience success. When you first define the BI

strategy, focus on key areas so that you achieve quick wins to demonstrate value, while achieving sustainable, incremental progress.

- **Strategic reasons**: You can have distinct initiatives for different business areas. For example, different parts of the organization might require independent BI strategies because their business strategies are sufficiently different. These independent strategies should align with an overall BI strategy, whenever possible.

As part of the scoping exercise, you should also plan how you'll set expectations with stakeholders that the BI strategy will be defined iteratively.

Understand the working team purpose and responsibilities

Once you've identified and engaged an executive sponsor and clarified the scope of the BI initiative, you then assemble the working team. This team leads the initiative to define and plan the BI strategy.

The responsibilities of the working team include:

- **Planning and preparation**: The working team should plan and prepare the various aspects of the BI strategy initiative, such as:
  - Defining the timelines, deliverables, and milestones for the initiative.
  - Identifying stakeholders who can accurately describe the business objectives and goals of their respective departments.
  - Communication with stakeholders, the executive sponsor, and each other.
- **Information gathering**: The working team should gather sufficient information to accurately assess the current state of the BI implementation. Examples of information gathering activities include:
  - Conducting independent research about the business context and existing BI solutions or initiatives.
  - Running interactive workshops with stakeholders to elicit input about business objectives and data needs.
  - Documenting summarized findings and sharing conclusions.
- **Feedback and follow-up**: The working team summarizes the findings from the information gathered and proposes BI objectives, focus areas, and next steps. It gathers feedback and follows up by:
  - Assessing the current state of BI adoption and implementation.
  - Creating a prioritized list of business data needs.
  - Presenting their conclusions and proposed next steps to stakeholders and executive leadership.

> **Note**
>
> Because the working team communicates with stakeholders and business users, they're often considered ambassadors for Fabric, Power BI, and other BI initiatives in your organization. Ensure that your working team members understand and accept this responsibility.

Assemble and prepare the working team

Working team members should include representatives from different departments and business units. The following sections describe where you might source working team members.

> **Important**
>
> The working team should be comprised of people with credibility in the organization. These people should have knowledge of technical processes and business processes. A working team that consists only of consultants could indicate that the BI initiative isn't sufficiently understood or prioritized by the organization.

Center of Excellence members

You can source working team members from the Power BI Center of Excellence (COE), or a similar group of multidisciplinary BI experts. The main responsibility of COE members in the working team is to take advantage of their COE expertise to contribute to information gathering. Further, COE members can share workshop findings back to the COE to inform tactical planning decisions and actions.

Some organizations don't have a COE, possibly because the role of a COE is performed by their BI team or IT. In this case, consider adding members from the BI team or IT to the working team.

> **Note**
>
> Ensure that the working team doesn't consist only of members from your COE, central IT, or BI teams. A BI strategy encompasses many areas of the organization, and each of these areas should be well represented.

> **Tip**
>
> If you don't have a COE, consider establishing one while assembling the working team. Establishing a COE with the working team members could be a natural evolution of the working team's activities, once the BI strategic vision is defined. This approach is a good way to capture the knowledge and understanding that the working team gained during the BI strategy initiative.

Business subject matter experts (SMEs)

Working team members should include business SMEs. The main responsibility of business SMEs is to represent their business unit. You should include business SMEs in the working team to avoid assumptions that results in a BI vision that doesn't work for part of the organization.

Business SMEs in the working team must have a deep understanding of data needs and business processes within their business unit or department. Ideally, they should also understand the BI tools and technologies used to address these needs.

> **Note**
>
> It might not be practical to include every department, business unit, or region in the working team. In this case, ensure that you dedicate effort to identifying assumptions and exceptions for any unrepresented departments, business units, or regions.

Champions network

Working team members can include users from your existing champions network in the community of practice. A champion typically has exceptional knowledge of both BI tools and the data needs of their business area. They're often leaders in the community of practice. The main responsibility of champions in the working team is to promote information gathering, and to involve their community of practice in the initiative.

> **Note**
>
> Including champions can help to avoid making assumptions that can result in an inaccurate assessment of the current state of **Fabric adoption** and **implementation**.

Functional, IT, and security team members

A working team can include members from specific functional areas, especially when other expertise is required. The main responsibility of these members is to bring their expertise about specific important topics to the BI strategy.

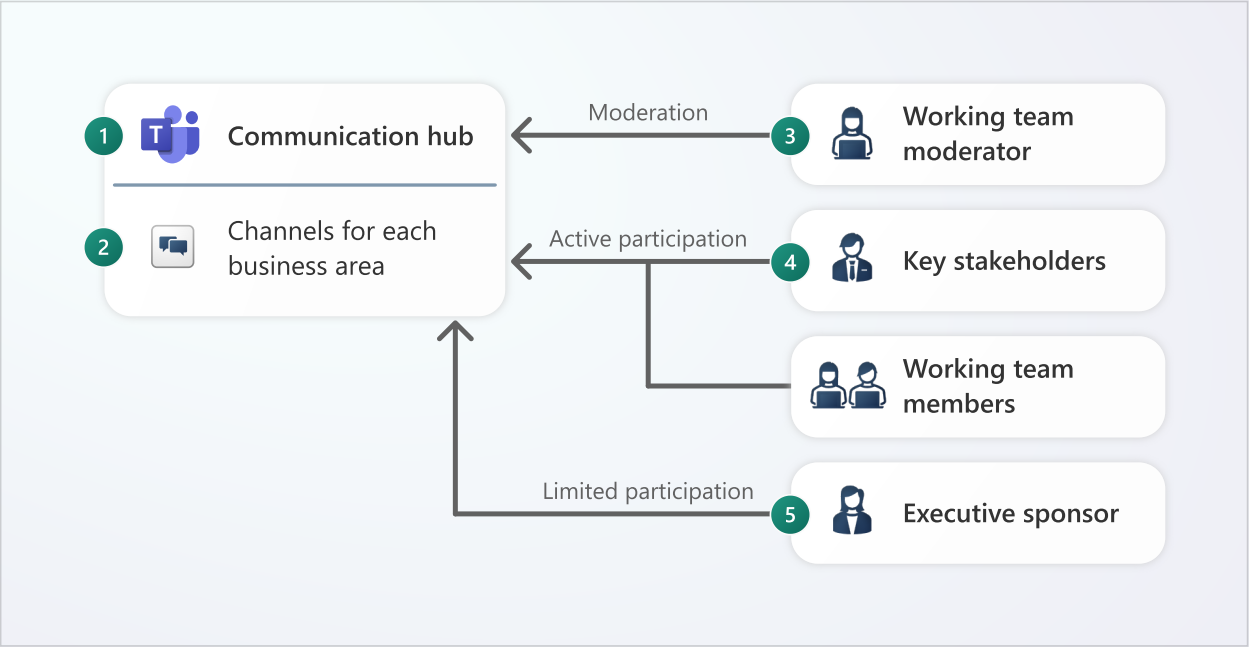
Here are some examples of when you might include members from functional areas in the working team.

- **Functional teams**: Include relevant representatives from functional teams in the working team. For example, if your organization uses one or more large enterprise resource planning systems (ERPs), then you should include an expert of these ERPs in the working team. This individual would be responsible for clarifying how the systems are used in the context of feedback provided during information gathering.
- **IT teams**: Include relevant IT experts in the working team. For example, your organization could have specific networking requirements, or a complex scenario involving multiple tenants. The IT experts would be responsible for describing specific requirements, which is particularly important in tactical planning. They can also help identify risks or pain points during information gathering.
- **Security teams**: Include members from security teams in the working team. For example, your organization could have specific compliance, security, or privacy requirements. These individuals would be responsible for describing security-related requirements when defining the future state. They can also help identify compliance risks and security threats during information gathering.

Create a communication hub

Efficient and structured communication between working team members and stakeholders is critical for your initiative to succeed. One way to improve communication is by centralizing it in a *communication hub*. A communication hub is a place that you use to consolidate communication, documentation, and planning about the BI strategy. It also helps promote collaboration between the working team and stakeholders.

The following diagram depicts how to use a communication hub to centralize BI strategic planning and input.



The diagram conveys the following concepts or processes.

Expand table

Item	Description
1	A communication hub is a central location in Microsoft Teams or a similar platform. Its purpose is to centralize communication, documentation, and planning.
2	The working team creates and manages different channels for each business area. The separation by business area should correspond to the top-level structure of the initiative. Each channel contains a repository of summarized findings, timelines, and discussions about the BI strategy.
3	Designated working team members curate and moderate the communication hub. Moderation ensures that the communication hub remains useful and current.
4	Key stakeholders and working team members actively participate in the communication hub.
5	The executive sponsor limits participation. For example, they might resolve disputes as they arise.

Tip

We recommend that you use the communication hub beyond the strategic planning workshops. Because the communication hub is a source of regular input and discussion from key business stakeholders, it can help your team keep the BI strategy relevant and up to date.

Communicate consistently and effectively

The working team should maintain and follow a concise, organized, and transparent process to define the BI strategy by using the communication hub.

Here are some recommendations to get the most value from the communication hub.

- **Have well-defined working team responsibilities**: Ensure that the working team has well-defined responsibilities for the communication hub, such as curating and moderating it. Having active and involved moderation ensures that the communication hub remains current, informative, and useful for the working team and key stakeholders.
- **Organize discussions and files**: Ensure that it's easy to find files or previous discussions in the communication hub by creating and maintaining a logical structure. An organized communication hub encourages its effective use.
- **Be concise in documents and posts**: Avoid overwhelming people with large volumes of information. Key stakeholders have limited time, so encourage people to publish posts and documents to the communication hub that are concise and easy to understand.
- **Be consistent in communication**: Ensure that the communication hub is used instead of alternative channels, like email. You should also strive to ensure that documents and updates are consistent in tone, format, and length.
- **Be transparent and foster a collaborative environment**: An effective communication hub has an active, collaborative social environment. It requires transparency from the working team who should be sharing regular updates and findings throughout the initiative.

> ⓘ **Important**
>
> Success of strategic planning relies on effective communication. Promoting concise and consistent communication benefits not only strategic planning, but also the broader adoption and implementation of BI initiatives across your organization.

Checklist - When establishing a working team, key decisions and actions include:

- ✓ **Involve an executive sponsor**: If there isn't an executive sponsor, identify and engage one before assembling the working team.

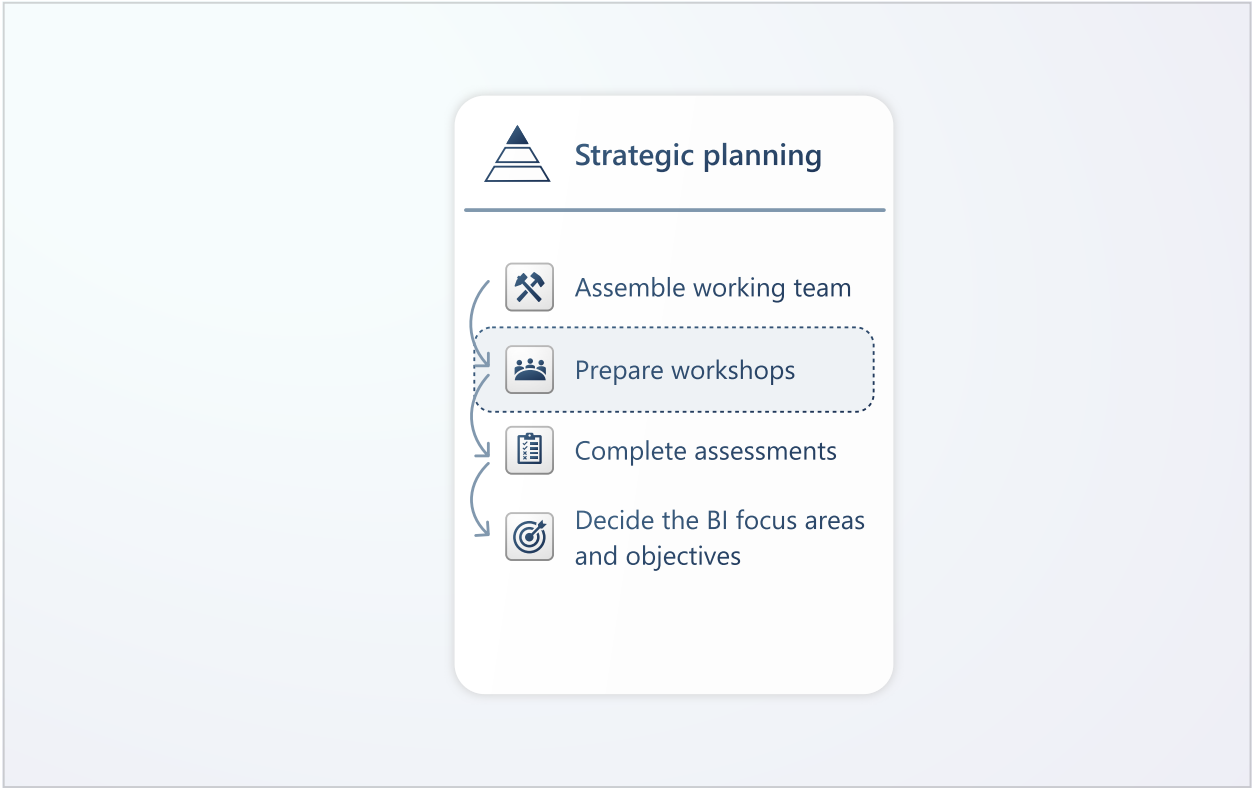
- ✓ **Decide on the scope of the BI initiative**: Together with the executive sponsor, determine which business areas the BI strategy will cover.
- ✓ **Communicate the initiative**: Have the executive sponsor raise awareness throughout the organization of the initiative to define the BI strategy.
- ✓ **Assemble the working team**: Appoint members who can provide sufficient coverage of the relevant business areas, technical areas, and compliance areas.
- ✓ **Set expectations of the working team members**: Clarify the time and effort requirements, and ensure that team members understand what's expected of them (and that they have the time and resources to fulfil their role).
- ✓ **Clarify working team roles and responsibilities**: Ensure that everyone in the working team knows what they should do to drive successful strategic planning.

Step 2: Plan workshops and conduct research

After you assemble the working team (step 1), the newly assembled working team can start the following activities to establish business alignment.

- **Conduct independent research**: The working team performs research into the business context and existing BI solutions or initiatives.
- **Plan workshops**: The working team prepares strategic planning workshops to collect input from key stakeholders about their business objectives and data needs.

These activities are prerequisites for the workshops and complete assessments (step 3).



Conduct independent research

The working team conducts research to document the current state of BI adoption and implementation. This research is used to complete assessments, but it also helps the working team to prepare for the workshops.

Research the business context

To define an effective BI strategy, the working team must understand the business objectives. By understanding the business objectives, the working team has the right business context to describe why people use data and BI tools, and comprehension of their desired outcomes. You should define data needs and use cases with respect to the business processes they support and the goals they address.

Business SMEs in the working team should use their expertise to lead the effort to describe the business context. However, it's important that all members of the working team participate. It's essential that the working team has a shared understanding of the business strategy. That way, the BI strategy focuses on addressing business needs instead of solving abstract, technical problems.

Research existing BI initiatives and solutions

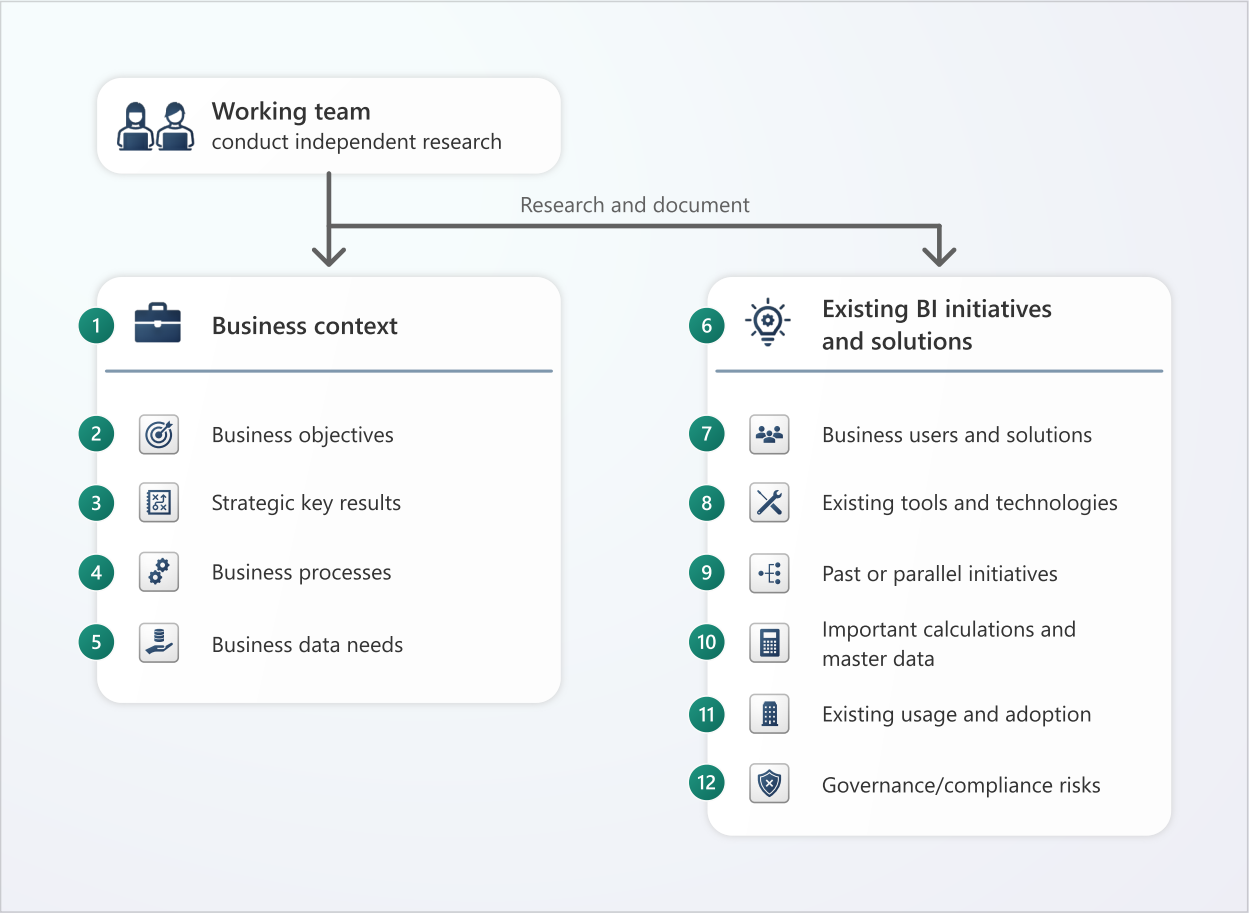
To define an effective BI strategy, the working team must also understand the current state of BI adoption and implementation. The current state describes how people use existing data and BI tools, and what data and tools are strategically important. You should identify the existing BI initiatives and solutions with respect to the business processes they support and goals they address. These solutions help illustrate what business users do today to address their data needs, so that you can assess whether it's effective.

COE members in the working team should use their expertise to lead the effort to describe the current state of BI adoption and implementation. An example of an activity that helps this effort is tenant-level auditing. Auditing allows the working team to collect an inventory of current BI initiatives and solutions to prepare for workshops.

> **Important**
>
> Ensure that the working team has a good understanding of the compliance requirements and information protection needed in your organization. Document these requirements during independent research, and ensure that they're well understood by everyone in the working team.

Topics to address with independent research

The following diagram depicts topics typically addressed with independent research.



The diagram depicts the following concepts and processes.

Expand table

Item	Description
1	The working team researches the business context to document and understand the business strategy. This research is led by business SMEs for their respective departments or business units.
2	The working team researches the business context by first identifying the business objectives.
3	The working team identifies specific business objectives that departments or business units have to make progress toward with their objectives.
4	Business processes are initiatives or plans created to work toward business objectives. The working team identifies the processes in place to help achieve the business objectives.
5	Business data needs are the data, tools, and solutions required to support business processes and strategic goals. The working team identifies the business data needs.

Item	Description
6	The working team researches any existing BI initiatives and solutions to understand the current state of BI adoption and implementation. COE members or BI experts lead this research.
7	The working team investigates strategically important BI solutions to understand how the organization currently addresses business data needs. Specifically, the working team identifies who the business users are, how they use the solutions. The working team also documents key data questions or problems that these solutions address, and also potential flaws, opportunities, and inefficiencies.
8	The working team surveys and documents the existing tools and technologies that the organization uses to address business data needs.
9	The working team identifies past or parallel initiatives to define the BI strategy. Past initiatives might contain valuable learnings, while parallel initiatives could be combined to avoid duplication of effort.
10	The working team identifies strategically important calculations and master data. These calculations and master data are critical to enabling the business to achieve their business objectives.
11	The working team assesses the usage and adoption of strategically important BI solutions among the user community.
12	The working team identifies any potential governance and compliance risks identified in existing BI solutions.

> **ⓘ Important**
>
> The topics and examples presented in this section are intended to guide you in conducting your own independent research. These topics aren't an exhaustive or required list. Use these topics as inspiration. We recommend that you use the maturity levels documented in the Fabric adoption roadmap to help you evaluate and focus upon areas that are most important for your organization and its business context.

Taken together, research on the business context and existing BI initiatives and solutions describe the current state of BI adoption and implementation. The working team verifies this research in workshops when capturing stakeholder input.

Plan workshops

While performing independent research, you should plan workshops with stakeholders. The purpose of these workshops is to gather input about the business objectives and

data needs. You also validate the conclusions from independent research in these workshops.

> **Note**
>
> This article uses the term *workshops* to describe interactive meetings with key stakeholders. The goal of the workshops is to gather input so you can accurately describe and understand the goals and data needs.

The following sections describe key considerations for planning and preparing workshops.

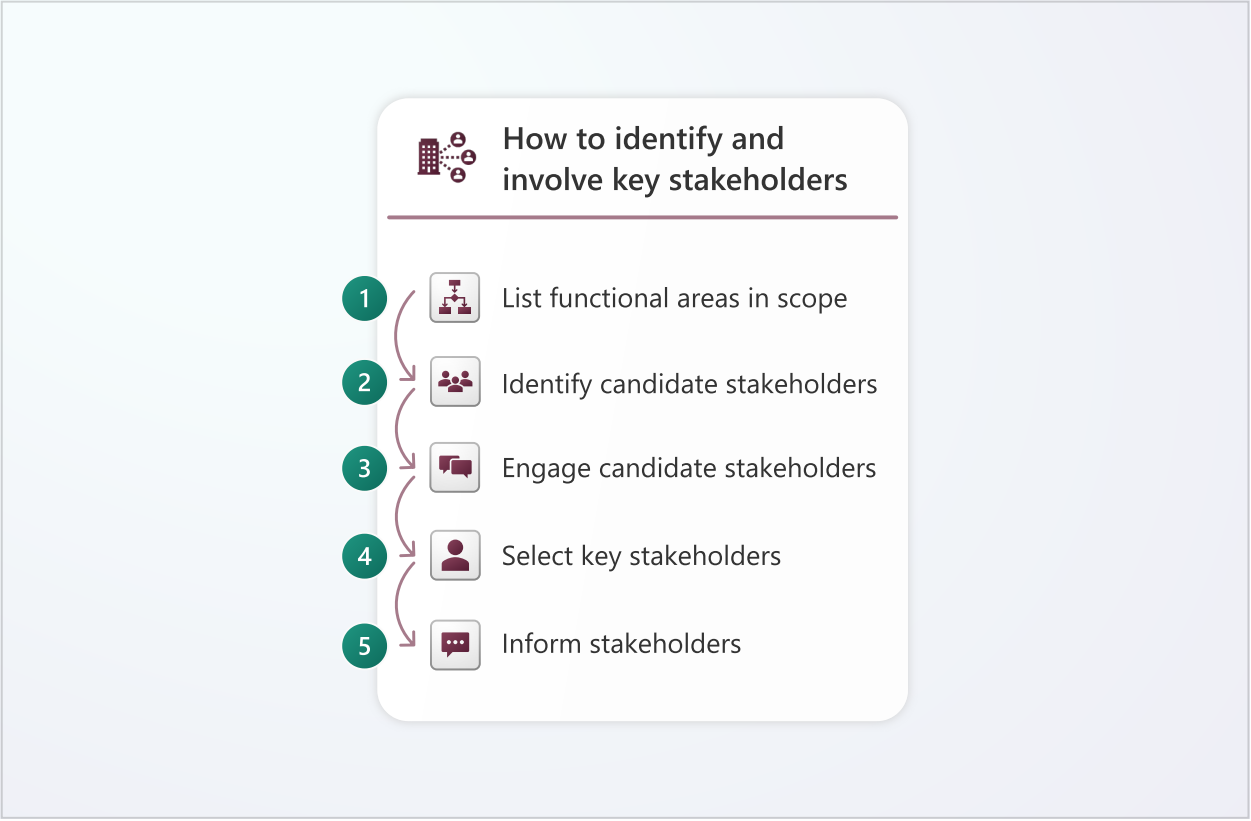
Involve the correct key stakeholders

Successful BI strategic planning requires the working team to involve the right stakeholders in the workshops. The working team should identify key stakeholders who have sufficient knowledge and credibility to represent their business area. In each workshop, the role of these stakeholders is to participate in discussions that are led by the working team. Stakeholders need to describe the business objectives and data needs for their areas, and the current state of data and analytics initiatives to support the business objectives.

Identifying the right stakeholders is essential in order to run successful workshops and gain an accurate understanding of the business areas in scope.

> **Warning**
>
> If you engage the wrong stakeholders, there's significant risk that the BI strategy won't align with strategic business objectives or support business users to achieve their goals.

The following diagram depicts the process to identify and inform the right key stakeholders about the BI strategy initiative.



The diagram depicts the following steps.

Expand table

Item	Description
1	List the functional areas (departments and business units) in scope for the BI strategy initiative.
2	For each functional area, identify two to three candidate key stakeholder representatives.
3	Engage with stakeholders to inform them of the initiative, and validate their selection. At this stage, candidate stakeholders can decline to participate and might suggest alternative people.
4	Select a final list of key stakeholders.
5	The executive sponsor informs key stakeholders and formally requests their participation. All further communication with the key stakeholders is posted to the communication hub.

When you initially request key stakeholder participation, ensure that you:

- Obtain approval from their manager, as appropriate.
- Explain the scope of the initiative, and its goals, timelines, and deliverables.
- Describe specifically why they've been asked to participate and what the desired outcomes of the initiative are.

- Outline the necessary time commitment and participation that you need from them.
- Communicate clearly and concisely.

> **ⓘ Important**
>
> Often, top-down BI initiatives limit stakeholders to executives and decision makers. While these individuals have a significant role to play (to obtain sufficient executive buy-in and strategic alignment), they aren't necessarily the right stakeholders. In this scenario, you risk defining a strategy isolated from the reality experienced by business users. This misalignment can result in strategies and solutions that don't meet the needs of day-to-day users, and consequentially they aren't used.
>
> To mitigate this risk, ensure that you involve stakeholders from various levels of the organization. When selecting key stakeholders, engage with different teams to briefly introduce the initiative and gather input on who the right stakeholders might be. This level of engagement not only raises awareness of the initiative, but it also enables you to involve the right people more easily.

Checklist - When planning workshops and conducting research, key decisions and actions include:

- ✓ **Agree on communication values**: Encourage all working team members to engage with concise, clear, and consistent communication throughout the initiative.
- ✓ **Set up the communication hub**: Create a central, structured hub for all communication, documentation, and planning. Document how the hub can be used effectively.
- ✓ **Research the business context**: With assistance from business SMEs, describe the business objectives for each of the business areas that are in scope.
- ✓ **Research existing BI initiatives and solutions**: Conduct tenant-level auditing and targeted investigation of strategically important solutions to describe the current state of BI adoption and implementation.
- ✓ **Select the right key stakeholders**: Engage representatives from each business area who have sufficient knowledge and credibility.
- ✓ **Invite key stakeholders to the communication hub**: When ready, onboard the key stakeholders to the communication hub and send meeting invitations for the workshops.

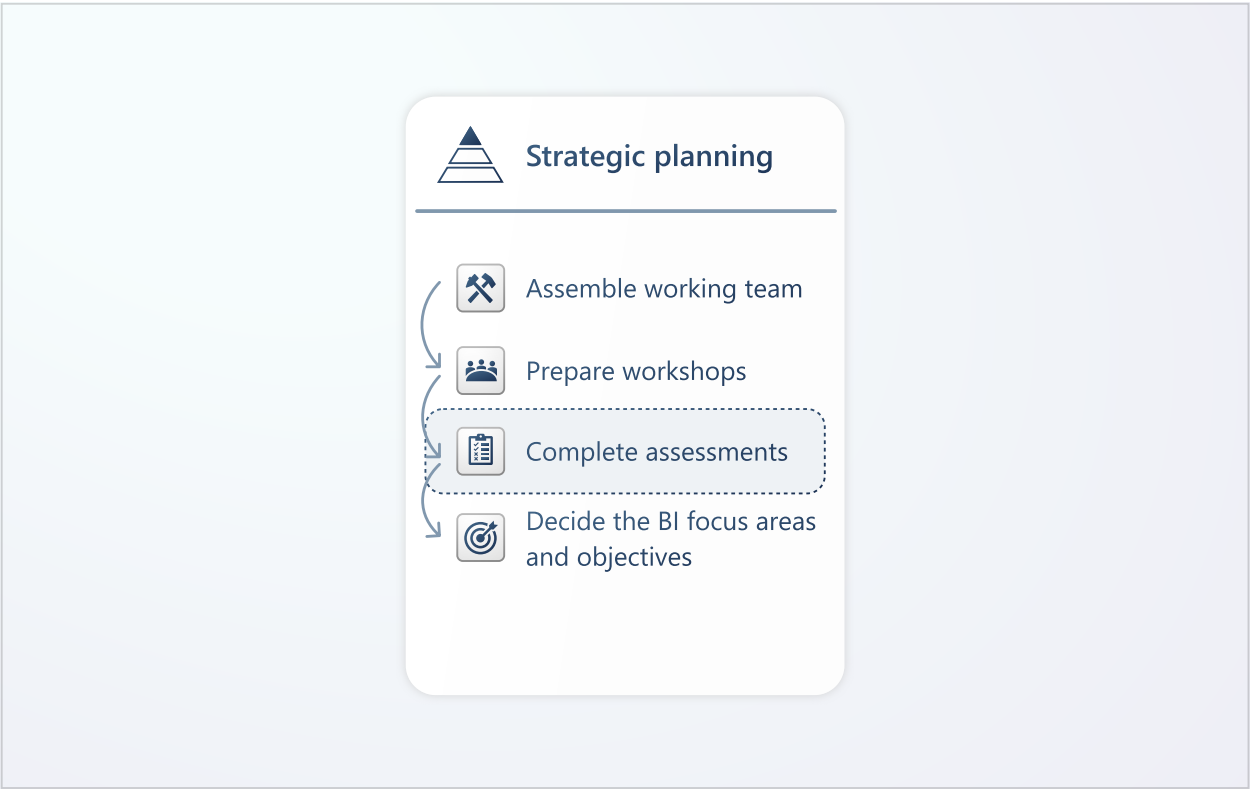
Step 3: Run workshops and complete assessments

After you complete independent research and workshop planning (step 2), you run the workshops and complete the assessments. The objective of the workshops is to use stakeholder input to document:

- The business objectives, strategy, and data needs of the in-scope business areas.
- The current state of BI adoption and implementation for the in-scope business areas.

The working team combines the stakeholder input together with independent research. These inputs should provide the working team with a sufficient understanding of the business strategy and the current state of BI adoption and implementation.

With this understanding, the working team evaluates the maturity and effectiveness of the current state of BI adoption and implementation. This evaluation is summarized in a data culture assessment and a technical assessment, which are the key outputs of the workshops. The goal of these assessments is to clearly identify both weaknesses and opportunities in the data culture, and technical areas that should be a focus for the BI strategy.



Important

If no working team members have experience running and moderating interactive meetings or workshops, the working team should first undertake training or seek support to help run the workshops.

Run workshops

The workshops are organized as a series of interactive sessions structured to effectively elicit and collect information from stakeholders. The number of sessions and how they're conducted will depend on the number of stakeholders, their location, time availability, and other factors.

The following sections describe the types of sessions you typically conduct when running workshops.

Introduction session

The introduction session is run by the working team, and it should involve all stakeholders and the executive sponsor. It introduces the initiative and clarifies the scope, goals, timeline, and deliverables.

The goal of this session is to set expectations about the purpose of the workshops and what's needed for the BI initiative to succeed.

Workshops

Workshops are interactive meetings between a few members of the working team and the key stakeholders. A member of the working team moderates the discussion, posing questions to stakeholders to elicit input. Stakeholders provide input about their business strategy, existing BI initiatives and solutions, and their data needs.

> **Note**
>
> While a moderator should be proficient in eliciting information, they don't require deep domain knowledge. Ideally, all workshops for a given business area should be led by the same moderator.

The goal of the workshops is to collect sufficient input from stakeholders to accurately describe their business objectives and data needs. A successful workshop concludes with stakeholders feeling that the working team members understand the business objectives and data needs. This stakeholder input is used together with the working

team's independent research to complete an assessment of the current state of BI adoption and implementation.

Here are some practical considerations to help you plan and organize effective workshops.

- **Keep workshop attendance focused**: Don't saturate meetings with too many attendees. Involving too many people can result in prolonged discussions, or discussions where only the most assertive personalities provide input.
- **Keep the discussion focused**: Take any debates, excessively specific questions, or remarks offline to discuss later in short one-on-one meetings. Similarly, identify and address any resistance directly, and involve the executive sponsor whenever necessary. Keeping the discussion focused ensures that workshops concentrate on the overall discussion of strategic planning, and they don't get distracted by small details.
- **Be flexible with preparation**: Depending on time and preference, you can use prepared material to conduct more effective discussion. However, understand that discussions could go in unexpected directions. If a session departs from your prepared material but still produces helpful input, don't force the discussion back to a fixed agenda. When stakeholders are focused on a different point, it means that it's important. Be flexible by addressing these points to capture the most valuable input.
- **Document stakeholder input**: During the workshops, you should document stakeholders' inputs about their business objectives and the BI strategy.
- **Document business data needs**: One outcome of workshop information gathering is a high-level list of the unmet business data needs. You should first organize the list from the highest to lowest priority. Determine these focus areas based on stakeholder input, and the impact the list items have on business effectiveness.

> ⓘ **Note**
>
> The list of prioritized data needs is a key outcome of strategic planning that later facilitates tactical planning and solution planning.

Complete assessments

The working team should combine independent research and the stakeholder input into summarized findings. These goal findings should convey an accurate description of the current state of BI adoption and implementation (for conciseness, referred to as *the current state*). For each business area in scope, these findings should describe:

- Business objectives.
- Business key results to measure progress towards their objectives.
- Business key initiatives, which are intended to achieve their key results.
- Business data needs to support the key initiatives.
- BI tools and solutions that people use to address their business data needs.
- How people use the tools and solutions, and any challenges that prevent them from using the tools and solutions effectively.

With an understanding of the current state, the working team should then proceed to assess the overall BI maturity and its effectiveness in supporting the business strategy. These assessments address specific data culture and technical areas. They also help you to identify weaknesses and opportunities that you'll focus on in your BI strategy. To address these weaknesses and opportunities, you define high-level, strategic BI objectives.

To help identify focus areas, the working team conducts two types of assessment: a data culture assessment and a technical assessment.

Contents of an assessment

Making a concise and accurate assessment of the current state is essential. Assessments should highlight the strengths and challenges of the organization's ability to use data to drive decisions and take actions.

An effective maturity assessment consists of the following content.

- **Maturity level**: Evaluate the overall maturity level on a five-point scale from 100 (initial) to 500 (efficient). The score represents a high-level, subjective assessment by the working team of the effectiveness in different areas.
- **Business cases**: Justify and illustrate the maturity level scores in the assessment. Concrete examples include actions, tools, and processes taken by business users to achieve their business objectives with data. The working team uses business cases together with summarized findings to support their assessment. A business case typically consists of:
  - A clear explanation of the desired outcome, and business data needs the current process aims to address.
  - An as-is description of how the general process is currently done.
  - Challenges, risks, or inefficiencies in the current process.
- **Supplemental information**: Support the conclusions, or documents significant details that are relevant to the BI and business strategy. The working team documents supplemental information to support later decision-making and tactical planning.

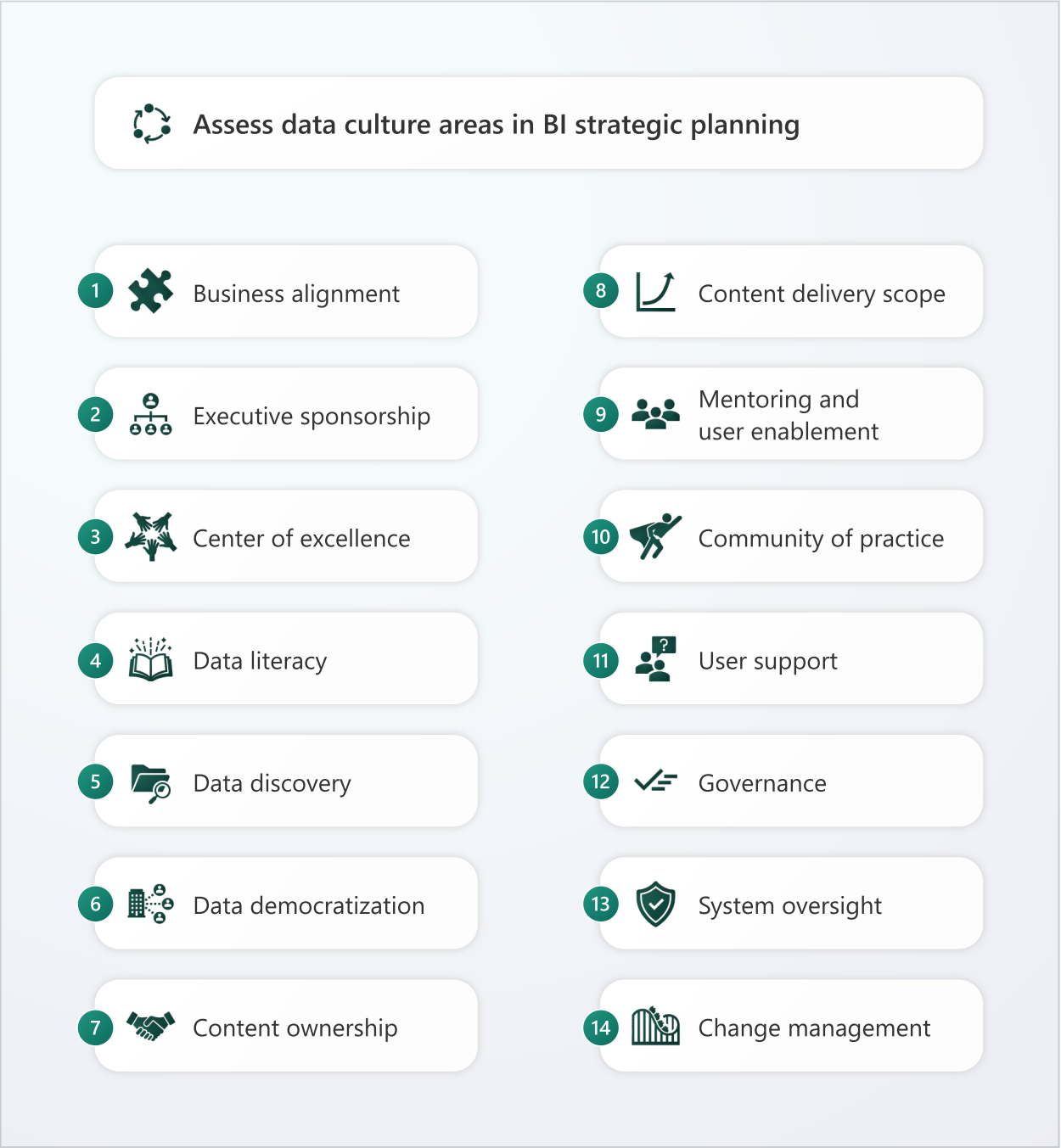
Complete the data culture assessment

The data culture assessment evaluates the current state of BI adoption. In order to complete this assessment, the working team performs the following tasks.

1. **Review summarized findings**: The working team reviews the inputs collected from conducting independent research and running workshops.
2. **Evaluate the maturity levels**: The working team proceeds through each of the data culture areas described in this section. Using the Fabric adoption roadmap, they evaluate the effectiveness of each area by assigning a maturity score.
3. **Justify the subjective evaluation with goal evidence**: The working team describes several key business cases and supporting information that justifies their evaluation of the maturity scores for each area.
4. **Identify weaknesses and opportunities**: The working team highlights or documents specific findings that could reflect a particular strength or challenge in the organization's data culture. It can be the lowest-scoring or highest-scoring areas, or any areas that they feel have a high impact on the organization's data culture. These key areas will be used to identify the BI focus areas and objectives.

> **Tip**
>
> Use the **Fabric adoption roadmap** to guide you when completing the data culture assessment. Also, consider other factors specific to your organizational culture and the ways your users work. If you're looking for more information, consult other reputable sources like the **Data Management Body of Knowledge (DMBOK)**.

The following diagram depicts how the working team assesses the organizational data culture in BI strategic planning for specific data culture areas.



The diagram depicts the following data culture areas.

Expand table

Item	Description
1	Business alignment: How well the data culture and data strategy enable business users to achieve business objectives.
2	Executive sponsorship: How effectively a person of sufficient credibility, authority, and influence supports BI solutions and initiatives to drive successful adoption.
3	Center of Excellence (COE): How effectively a central BI team enables the user community, and whether this team has filled all the COE roles.

Item	Description
4	Data literacy: How effectively users are able to read, interpret, and use data to form opinions and make decisions.
5	Data discovery: How discoverable the right data is, at the right time, for the people who need it.
6	Data democratization: Whether data is put in the hands of users who are responsible for solving business problems.
7	Content ownership and management: Whether there's a clear vision for centralized and decentralized ways that content creators manage data (such as data models), and how they're supported by the COE.
8	Content delivery scope: Whether there's a clear vision of who uses, or consumes, analytical content (such as reports), and how they're supported by the COE.
9	Mentoring and user enablement: Whether end users have the resources and training to effectively use data and improve their data literacy.
10	Community of practice: How effectively people with a common interest can interact with and help each other on a voluntary basis.
11	User support: How effectively users can get help when data, tool, or process issues arise.
12	Governance: The effectiveness of processes for monitoring user behavior to empower users, maintain regulatory requirements, and fulfill internal requirements.
13	System oversight: The effectiveness of everyday administrative activity concerned with enacting governance guidelines, empowering users, and facilitating adoption.
14	Change management: How effectively change is handled, including procedures that safeguard against disruption and productivity loss due to changes in solutions or processes.

To evaluate these data culture areas, see the Fabric adoption roadmap. Specifically, refer to the maturity level sections and *Questions to ask* sections, which guide you to perform assessments.

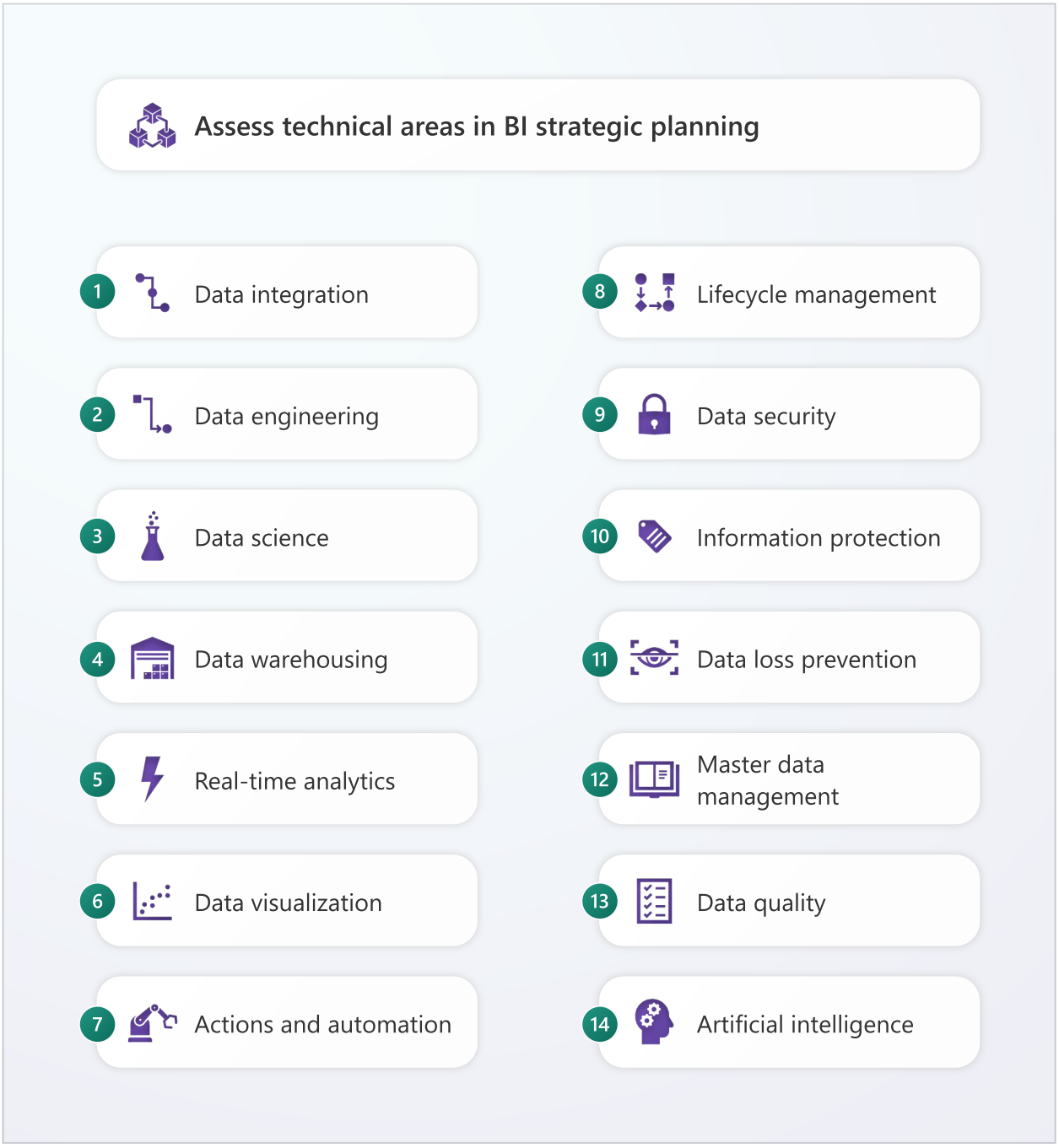
Complete the technical assessment

The technical assessment evaluates technical areas that strategically enable the success of BI implementation. The purpose of this assessment isn't to audit individual technical solutions or assess the entirety of technical areas related to BI. Instead, the working team describes the maturity level and general effectiveness for strategically critical areas, like those described in this section. To complete this assessment, the working team performs the following tasks.

1. **Identify technical areas**: The working team identifies specific technical areas that are relevant and strategically important to the success of BI to include in their assessment. Some examples of technical areas are described in this section and shown in the following diagram.
2. **Define maturity levels**: The working team defines the maturity levels to score the high-level effectiveness for each technical area in the assessment. These maturity levels should follow a consistent scale, such as those found in the template provided in the maturity levels of the Fabric adoption roadmap.
3. **Review summarized findings**: The working team reviews the collected inputs by conducting independent research and running workshops.
4. **Evaluate the maturity levels**: The working team evaluates the effectiveness of each area by assigning a maturity score.
5. **Justify the subjective evaluation with goal evidence**: The working team describes several key business cases and supporting information that justifies their evaluation of the maturity scores for each area.
6. **Identify weaknesses and opportunities**: The working team highlights or documents specific findings that could reflect a particular strength or challenge in the organization's BI implementation. It can be the lowest-scoring technical areas, or any areas that they feel have a high impact on the organization's strategic success with implementing BI tools and processes. These key areas will be used to identify the BI focus areas and objectives.

The following diagram depicts technical areas that you might assess when defining your BI strategy.

> **Note**
>
> If you're adopting Microsoft Fabric, be aware that many of these areas are represented as separate parts of the Fabric analytics platform.



The diagram depicts the following technical areas.

Expand table

Item	Description
1	Data integration: How effectively tools or systems connect to, ingest, and transform data from various sources to create harmonized views for analytical purposes. Evaluating data integration means equally assessing enterprise data pipelines and self-service data integration solutions, like dataflows in Power BI and Fabric.
2	Data engineering: How effective the current architectures are at supporting analytical use cases and adapting to changes in business data needs.
3	Data science: Whether the organization can use exploratory and sophisticated techniques to discover new insights and benefit from predictive or prescriptive analytics.

Item	Description
4	Data warehousing: The effectiveness of relational databases in modeling business logic to support downstream analytical use cases. Data warehousing is often considered together with data engineering.
5	Real-time analytics: Whether the organization can correctly identify, capture, and use low latency data to provide an up-to-date picture of systems and processes.
6	Data visualization: Whether visualizations can be used effectively to reduce the time-to-action of reporting experiences for business users. Effective visualizations follow best practices, directing attention to important, actionable elements, enabling users to investigate deeper or take the correct actions.
7	Actions and automation: How consistently and effectively tasks are automated and data alerts are used to enable manual intervention at critical moments in a system or process. You should also evaluate how actionable BI solutions are, meaning how effectively and directly they enable report users to take the right actions at the right time.
8	Lifecycle management: How effectively content creators can collaborate to manage and track changes in BI solutions for consistent, regular releases or updates.
9	Data security: Whether data assets comply with regulatory and organizational policies to ensure that unauthorized people can't view, access, or share data. Data security is typically evaluated together with information protection and data loss prevention.
10	Information protection: How well the organization mitigates risk by identifying and classifying sensitive information by using tools like sensitivity labels. Information protection is typically evaluated together with data security and data loss prevention.
11	Data loss prevention (DLP): Whether the organization can proactively prevent data from leaving the organization. For example, by using DLP policies based on a sensitivity label or sensitive information type. DLP is typically evaluated together with data security and information protection.
12	**Master data management** :Whether quantitative fields and business attributes are effectively managed, centrally documented, and uniformly maintained across the organization.
13	**Data quality**: Whether BI solutions and data are trustworthy, complete, and accurate according to the business user community.
14	**Artificial intelligence (AI)**: Whether the organization makes effective use of generative AI tools and models to enhance productivity in BI processes. Additionally, whether AI is used to deliver valuable insights in analytics workloads.

> ⓘ Note
>
> The technical areas depicted in the diagram aren't all necessarily part of BI; instead some are strategic enablers of a successful BI implementation. Further, these areas

don't represent an exhaustive list. Be sure to identify and assess the technical areas that are strategically important for your organization.

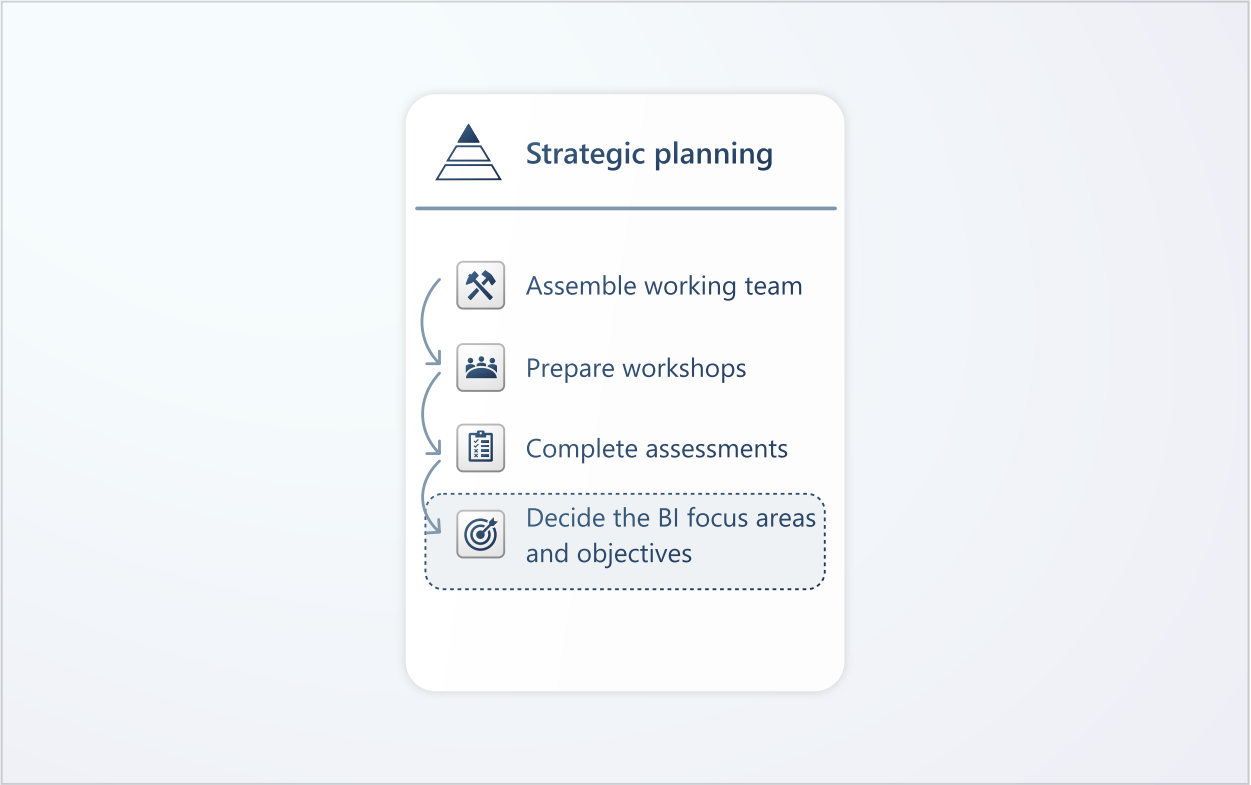
> ⊗ **Caution**
>
> When performing the technical assessment, don't assess details beyond the scope of strategic planning. Ensure that all activities that investigate the BI implementation focus directly on defining and evaluating the current state to define your BI focus areas and objectives.
>
> Getting too detailed in the technical assessment risks diluting key messages about the BI strategy. Always keep in mind the big picture questions like: *Where do we want to go?* and *How can BI effectively support the business?*

Checklist - When running workshops and completing assessments, key decisions and actions include:

- ✓ **Decide and communicate the workshop format**: Outline the number of sessions, their length, participants, and other relevant details for participating stakeholders.
- ✓ **Nominate a moderator from the working team**: Decide who from the working team will moderate the workshops. Their objective is to guide discussions and elicit information.
- ✓ **Collect input**: Organize the workshops so that you collect sufficient input about the business strategy and the current state of BI implementation and adoption.
- ✓ **Summarize findings**: Document the inputs that justify the assessments. Include specific business cases that illustrate strategically important processes and solutions.
- ✓ **Complete the maturity assessments**: Complete the relevant assessments for the current state of BI adoption and implementation.
- ✓ **Document business cases and supporting information**: goally document the evidence used to justify the maturity levels you assign in each assessment.

Step 4: Decide on the BI focus areas and objectives

After you run the workshops and complete assessments (step 3), the working team, together with the executive sponsor, decide on the BI focus areas and objectives to address in tactical planning.



> **Note**
>
> While the working team should be involved in clarifying and documenting focus areas and objectives, it isn't responsible for defining them. The executive sponsor and equivalent decision makers own these decisions. The executive sponsor and other decision makers have the authority to decide and allocate resources to deliver on these focus areas and objectives.

Decide on strategic focus areas

The assessments should clearly identify weaknesses and opportunities in the data culture or technical areas to focus on for the BI strategy. From the weaknesses and opportunities in the assessments, work with key decision makers, like your executive sponsor, to decide which of the areas are focus areas you'll aim to improve in the short-term. With this focus, you aim for sustainable, incremental progress toward your BI objectives.

Decide on strategic BI objectives

In the last step of BI strategic planning, for each of the focus areas, the working team defines several objectives to work toward in the next 12-18 months. Typically, these objectives represent the envisioned future state and maturity level growth.

> **Tip**
>
> For data culture areas, we recommend that you define your objectives by using the **Fabric adoption roadmap**. It can help you to identify the maturity level you should aim to achieve for your desired future state. However, it's not realistic to aim for a level 500 for each category. Instead, aim for an achievable maturity level increase in the next planning period.

For technical areas, we recommend that you define your objectives by using the maturity scales described in the technical assessment by the working team.

Examples of strategic BI objectives

Here are some examples of strategic BI objectives.

- Improve executive support of BI initiatives and solutions.
- Improve the effectiveness of the COE.
- Create a clear content ownership strategy and structure.
- Better understand and monitor user behavior with data to improve governance.
- Move from descriptive analytics solutions to predictive analytics solutions.
- Improve decision-making processes with more effective data visualization.
- Expand the number of effective content creators to improve time-to-delivery and the business value obtained from BI solutions.

Before you conclude strategic planning, the working team should align the decided BI focus areas and objectives with stakeholders and executives.

Align with stakeholders and executives

It's critical that the final assessments and decisions be shared with stakeholders. In the communication hub, stakeholders can asynchronously follow up on the progress of these deliverables and contribute feedback. However, you should conclude strategic planning by presenting the assessments and focus areas back to stakeholders and executives.

The following sections describe how you align with stakeholders and executives.

Conduct an alignment session

The alignment session is the final meeting for each business area. Each alignment session involves key stakeholders and the executive sponsor, who review the assessments made by the working team.

The goal of this session is to achieve consensus about the conclusions and assessments, and the agreed BI focus areas and objectives.

> **Note**
>
> Ensure that stakeholders understand that the BI strategy isn't final and unchanging. Emphasize that the BI strategy evolves alongside the business and technology. Ideally, the same stakeholders will continue to take part in this iterative exercise.

Prepare and present an executive summary

The executive summary is typically delivered by the executive sponsor to other executives responsible for the overall business strategy. The executive sponsor describes the assessment results and outlines the key challenges and opportunities that justify the focus areas identified. Importantly, the executive sponsor describes the next steps to start making actionable progress toward the future state.

The goal of this session is to obtain executive alignment and approval on the outcomes of strategic planning and the next steps.

Proceed with tactical planning

Once you've identified your BI focus areas and objectives, you have concluded strategic planning. The next step is to identify goals to help you make progress towards your BI objectives, which you do by conducting tactical planning.

Checklist – When deciding BI focus areas and objectives, key decisions and actions include:

- ✓ **Curate a list of business data needs and opportunities**: Create a consolidated, prioritized list of the business data needs, pain points, and opportunities. This output is used in tactical planning.

- ✓ **Decide on the strategic BI objectives**: Work with your executive sponsor and other decision makers to identify high-level BI objectives for the next 12-18 months.
- ✓ **Align with stakeholders**: Obtain consensus agreement that the assessments and other deliverables are accurate.
- ✓ **Align with executives**: Obtain approvals on the outcomes of strategic planning and the next steps.

Related content

In the next article in this series, learn how to conduct BI tactical planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: BI tactical planning

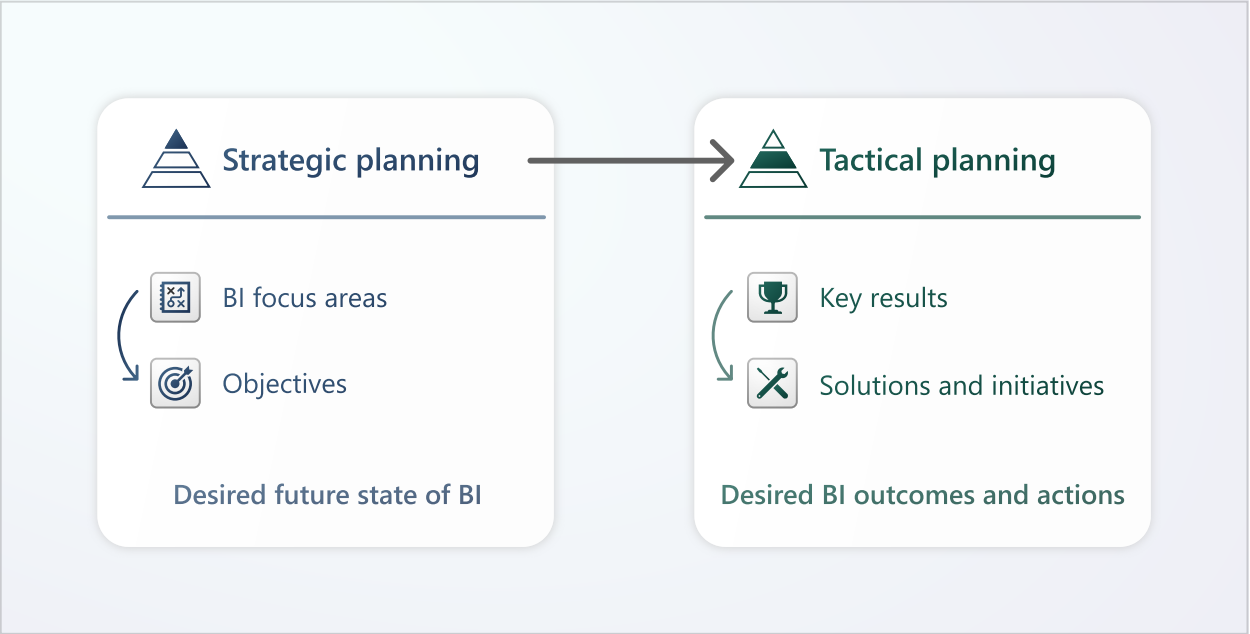
Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to identify your business intelligence (BI) key results and form actionable, time-bound plans to achieve incremental progress toward your strategic BI objectives. It's primarily targeted at:

- **BI and analytics directors or managers**: Decision makers who are responsible for overseeing the BI program and BI strategic planning.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for tactical planning, and for measuring and monitoring progress toward achieving BI key results.
- **Subject matter experts (SMEs) and content owners and creators**: The teams and individuals that champion analytics in a department and conduct BI solution planning.

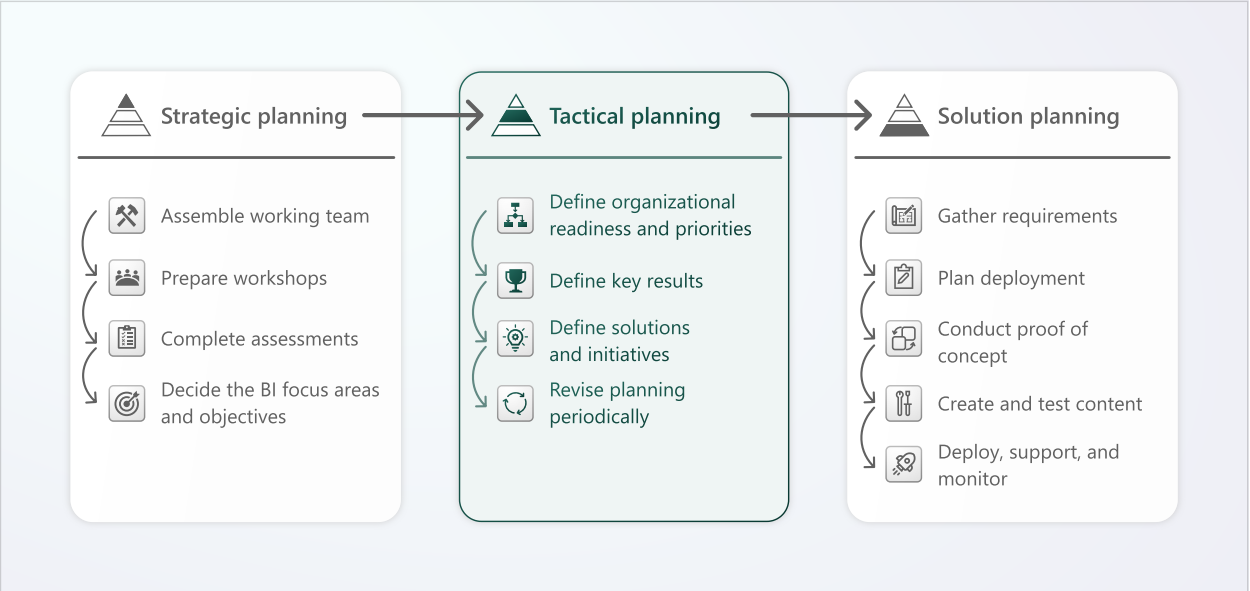
A BI strategy is a plan to implement, use, and manage data and analytics. You define your BI strategy by starting with strategic planning. In strategic planning, you assemble a working team to identify your BI focus areas and objectives. To make progress toward these objectives, the working team next conducts *tactical planning*. Tactical planning involves defining *key results*. Key results describe how you'll make progress toward one of your objectives. Key results are specific, actionable, and achievable within a defined period of time.



In short, strategic planning defines how you envision your long-term, desired future state, whereas tactical planning establishes what concrete, measurable, short-term actions you'll take to get to your desired BI outcomes.

> **Note**
>
> In the objectives and key results (OKRs) framework, *objectives* are clear, high-level descriptions of what you want to achieve. In contrast, *key results* are specific, achievable outcomes to measure progress toward one of your objectives.
>
> Further, *initiatives* or *solutions* are processes or tools built to help you achieve one or more key results. Solutions address specific data needs for users. A solution can take many forms, such as a data pipeline, a data lakehouse, or a Power BI semantic model or report.
>
> For more about OKRs, see Get to know OKRs (Microsoft Viva Goals).

The following high-level diagram depicts how to conduct BI tactical planning.



You take the following steps to conduct BI tactical planning.

Expand table

Step	Description
1	Identify and describe specific, actionable priorities that relate to your BI focus areas and objectives.
2	Define what success will look like, and what your key results are for the planning period. These key results are quantifiable goals that help you measure progress toward one of your objectives.
3	Prepare to reevaluate and assess planning periodically in subsequent periods.

Tip

Refer regularly your **assessments** from strategic planning when you conduct tactical planning. Focus on the key weaknesses to improve, and the opportunities to leverage.

Step 1: Define organizational readiness and priorities

The first step of tactical planning is to define your organizational readiness and to identify activities that you should prioritize in the current planning period.

To ensure that you define achievable and realistic key results, we recommend that you first evaluate your organizational readiness.

Evaluate organizational readiness

The key results you define during tactical planning should produce achievable outcomes during the planning period (the next 1-3 months). You should assess your organizational readiness to understand what achievable means for the context of your organization. Furthermore, you should also clearly define potential risks or threats that could prevent you from successfully achieving progress with your strategic objectives.

Assess organizational readiness by considering the factors described in the following sections.

Identify obstacles

You should identify any obstacles or dependencies that could hinder success or block progress in achieving the objectives that you defined in strategic planning. When you identify obstacles, describe how they might affect your activities in the next period. Define any relevant timelines, what actions could remove these obstacles, and who should perform these actions. You should also assess the risk of possible future obstacles that could prevent you from achieving your key results.

Here are some examples of obstacles.

- System migrations and other ongoing technical initiatives
- Business processes and planning, like fiscal year budgets
- Business mergers and restructuring
- Availability of stakeholders
- Availability of resources, including the available time of team members or SMEs
- Communication and change management activities to adequately inform and prepare business users about the BI strategy
- Skills of central team members and business users

Assess the necessary skills and knowledge

Teams and individuals in the organization should have the necessary skills and knowledge to achieve your key results. That's particularly true for central teams, like the COE or support teams that should lead by example. Confer with these teams about your objectives, and discuss openly what kind of activities and key results you could have in mind. Identify early on whether they require training or if there's gaps in the knowledge or profiles of the team.

To appraise the skills and knowledge of teams for organizational readiness, ask yourself the following questions.

- Do central teams (like the COE) understand the focus areas and objectives defined in strategic planning?
- Are special training programs needed for topics like security, compliance, and privacy?
- What new tools or processes might require user training? Who could organize this training?

> **Important**
>
> Improving the skills and competences of internal teams is particularly important when you migrate to Fabric or Power BI from other technologies. Don't rely exclusively on external consultants for these migrations. Ensure that internal team members have sufficient time and resources to upskill, so they'll work effectively with the new tools and processes.

Anticipate change management efforts

Change management is a crucial part of successful adoption and implementation. It's essential that you prepare and support people at all levels of the organization to successfully adopt new behaviors, tools, and processes for working with data. Consider

who will be responsible for change management activities and what resources are available to plan and complete these activities.

> **Important**
>
> Don't underestimate the importance of change management in making consistent progress toward your strategic objectives. Change management should be a priority to ensure successful evolution toward your desired future state.

Identify priorities

Once you've assessed your organizational readiness, you should proceed by identifying activities to prioritize for the current planning period. We recommend that you first prioritize time-sensitive, quick-win, and high-impact activities.

Time-sensitive activities

Some activities have a defined window of action, meaning that they must be addressed before a deadline or specific event occurs. Typically, these activities address problems that don't currently impact the business, but will impact the business at some time in the future (if left unaddressed). Alternatively, these activities can be linked to technology or business deadlines. You should identify and address these activities before the time window of action expires.

Here are some examples of time-sensitive areas you might prioritize when you define your key results.

- Tools, systems, or features that have a known decommission date.
- Business processes or initiatives that have a deadline.
- Known flaws or risks inherent in existing solutions or processes.
- Processes with a high degree of manual data handling and capacity constraints.
- The conclusion of a fiscal or budgeting period.

> **Important**
>
> While it's important to identify and act upon time-sensitive activities, you should also be careful to avoid a scenario where consistent urgency prevents you from achieving progress toward your high-level objectives.

Quick wins and high-impact activities

When assessing timelines and priorities, you should identify *quick wins*. Quick wins are short-term activities that deliver a significant long-term benefit. For instance, a quick-win might have few dependencies or not involve significant new designs or processes. The key benefit of a quick-win is that it can quickly demonstrate a return on the BI strategic initiative for the business. This return creates momentum that can lead to support of larger initiatives.

Quick wins can also be high-impact activities. In this case, they have the potential to make substantial advancements across many areas of the business.

Here are some examples of quick-win or high-impact areas you might prioritize when you define your key results.

- Minor changes that improve existing solutions for a large number of end users.
- Solution audits and optimizations that improve performance and reduce capacity usage and costs.
- Training initiatives for key users.
- Setting up a centralized portal to consolidate a user community of practice.
- Beginning to connect Power BI or Fabric champions to start a champions network for these individuals to share knowledge and practices.
- Creating shared, central themes, templates, and design guidelines for reports.

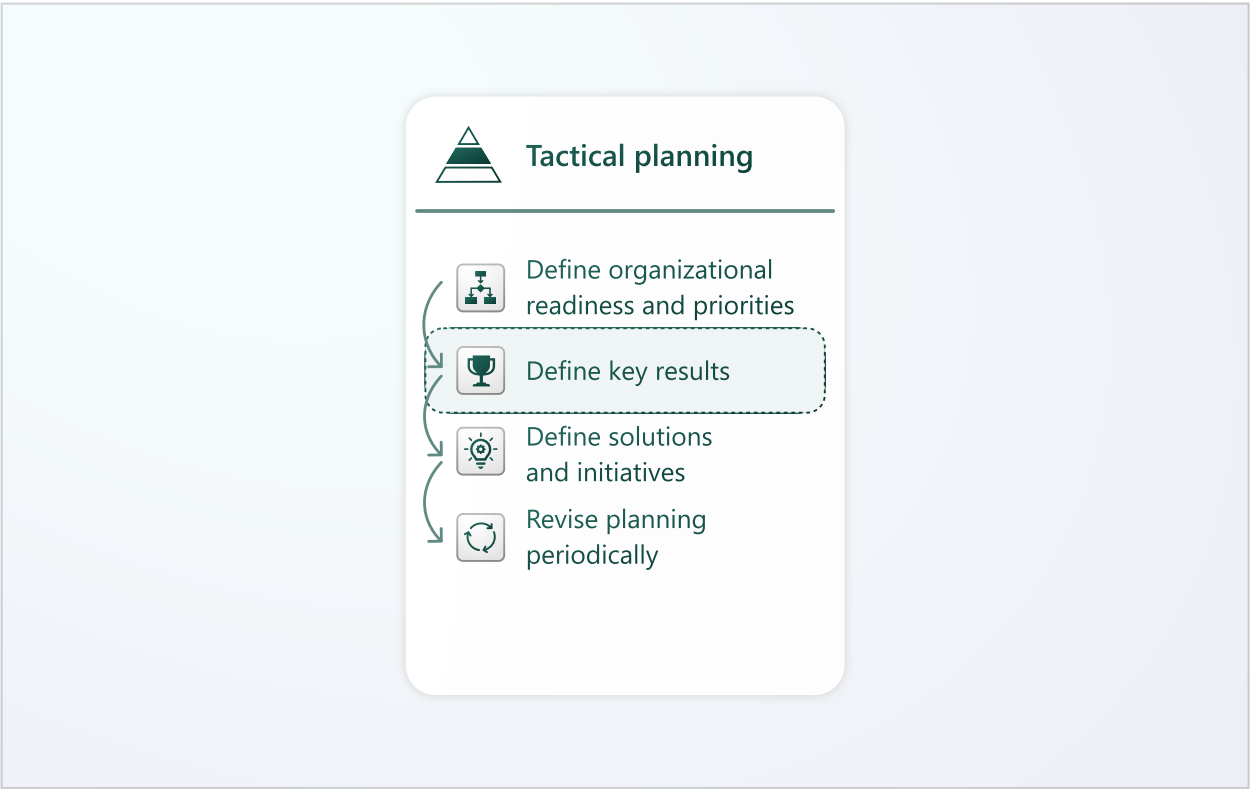
Checklist - When defining organizational readiness and priorities, key decisions and actions include:

- ✓ **Review BI focus areas and objectives**: Ensure that your strategic objectives are current, and that they're understood by everyone who participates in tactical planning.
- ✓ **Review the current state assessments**: The weaknesses and opportunities that the working team identified in the current state assessments directly inform the key results that you'll define.
- ✓ **Identify time-sensitive activities**: Identify any potential activities or areas that have a defined time period or urgency. Clarify the deadline and its importance for the business.
- ✓ **Identify quick wins**: Identify activities that require low effort or time investment to achieve a high impact. Justify why these activities are quick wins.
- ✓ **Identify high-impact activities**: Identify areas that have a significant impact on your BI strategy. Define why these areas have a high impact.

- ✓ **Assess organizational readiness**: Survey for potential risks or threats that you should address to successfully achieve your strategic objectives.

Step 2: Define key results

Once you understand your organizational readiness and any activities that you should prioritize, you're ready for the next step. In step 2 of tactical planning, you define key results to help you track progress toward your strategic objectives.



As mentioned in the previous section, key results are one of the main outcomes of tactical planning. Defining good key results is essential to facilitate focus on outcomes instead of output and to motivate progress toward achieving your objectives.

Ensure that your key results:

- Address one (or more) of the objectives of your BI strategy.
- Result in measurable and achievable outcomes within the tactical planning period.
- Relate to both your business strategy and BI strategy.
- Follow consistent criteria, like the SMART system, and that they're:
  - **Specific**: Target an explicit area of improvement.
  - **Measurable**: Are quantifiable so that you can monitor progress.
  - **Assignable**: Specify who's responsible for the key result.
  - **Realistic**: State whether you will achieve the key result, given the current level of organizational readiness and available resources. Set challenging yet achievable goals.

    - **Time-related**: Specify when you will achieve the results.

A fundamental aspect of your key results is that they help you to define and measure success for your BI strategy.

Define and measure success

It's important that you define what success will look like for your strategic objectives. There are several reasons why you should define and measure success.

- **Demonstrate progress**: A key element of clear success criteria is the ability to acknowledge progress and achievements. Good measures of success demonstrate a clear return on investment (ROI) in BI initiatives. While ROI can be challenging to measure, doing so drives motivation and allows leadership to acknowledge the realized business value of the BI strategy.
- **Continuous improvement**: Clear success criteria help you to (re-)evaluate your strategy. This evaluation should motivate your iterative tactical planning, together with user feedback and changes to the business or technology.
- **Corrective action**: A good definition of success is backed by measurable outcomes. Monitoring these measurable outcomes during operations can inform specific decisions and actions to adjust tactical planning, or intervene if you're heading off track.

There are two ways to track measurable achievement. In this article, we discuss key results, which are part of the OKR framework (objectives and key results), but organizations also use KPIs (key performance indicators), or a combination of OKRs together with KPIs. Both approaches are equally valid. What's most important is that you find an approach to measure progress toward your strategic objectives that works for you.

- **Key results**: Evaluate measurable success criteria that track progress toward strategic objectives.
- **KPIs**: Evaluate the success of a particular activity against a target. While KPIs typically measure performance against goals, key results measure outcomes. You can use KPIs together with OKRs.

> ⓘ **Note**
>
> Your measures of success should be closely aligned with business objectives. Ensure that your success criteria aren't specific to technical tasks.

> ⊗ **Caution**
>
> Measure a limited number of KPIs or key results. These metrics are only useful when you know what they measure and how you should act upon them. It's better to have a few strategic, valuable indicators than many which you can't regularly monitor or follow up.

Effectively use indicators

When you use measure progress toward your strategic objectives, you should regularly monitor them to track progress and take action where necessary.

Here are some general considerations to help you successfully measure and monitor key results and KPIs.

- **Report your indicators**: Create reporting solutions for your indicators that let you effectively monitor them. For instance, you can use Microsoft Viva Goals OKRs or scorecards and metrics in Power BI to measure and track progress.
- **Automate data collection**: If possible, ensure that data for indicators aren't collected manually. Find efficient ways to streamline and automate the collection of the data so that it's current, accurate, and reliable.
- **Track change**: Visualize the current indicator value, but also the trend over time. Progress is best demonstrated as a gradual improvement. If the indicator exhibits high volatility or variance, consider using a moving average to better illustrate the trend.
- **Assign an owner**: Ensure that a team or individual is responsible for measuring the indicator and keeping its data current.
- **Define an acceptable range**: Establish targets or an acceptable range of values to assign status (like *on track* or *off track*) to the indicator. When values fall outside the target or range, it should prompt someone to investigate or take corrective action.
- **Set up data-driven alerts**: Set up automated alerts that notify key teams or individuals, for example, by using Power Automate. That way, timely action can be taken when the indicator is off track.
- **Define actions and interventions**: Clearly describe how you'll use this information to take action, either to address issues or to justify moving to the next step in your BI strategy.

> ⓘ **Important**

Ensure that your chosen indicators genuinely reflect your desired outcomes. Regularly evaluate these indicators to survey for avoid incentivizing counterproductive behaviors. Consider Goodhart's Law, which states: *When a measure becomes a target, it ceases to be a good measure*.

Identify key results

You should identify and describe key results for adoption, governance, and implementation. Identify key results that you can achieve in the next period and that directly address the weaknesses and opportunities that you identified in your data culture and technical assessments. Take into account any activities or areas that you must prioritize, which were mentioned in step 1.

> **Tip**
>
> Refer to the relevant sections of the **Fabric adoption roadmap** and the **Power BI implementation planning** to help you define and describe your key results.

> **Important**
>
> When defining your key results, remember that the successful implementation of your BI strategy is more likely when you aim for an evolution instead of a revolution from your current state. *Evolution* implies that you strive for gradual change over time. Small but consistent, sustained progress is better than an abundance of change that risks disruption to ongoing activities.

Adoption

First, identify your adoption key results. These activities can address many areas, but typically describe the actions you'll take to improve overall organizational adoption and data culture.

Here are some examples of adoption key results that you could define for a period.

- Increase the usage of an endorsed central report or semantic model by a certain percent, or by a certain number of users.
- Identify one or more Power BI champions from each department or from each team within a department.

- Increase the positive response rate of business users to the question *BI tools and initiatives help me achieve my business objectives* by a certain percent.
- Guide a certain percent of sales teams through an introductory data literacy training program.
- Improve attendance of office hours Q&A sessions by an average of a certain number of people per session.
- Host a certain number of mentorship sessions with Power BI champions.
- Host a certain number of sessions introducing Power BI champions to the capabilities and use-cses of Microsoft Fabric.
- Dedicate a certain number of hours per week (on average) of the COE to user mentorship and enablement activities.

Governance

Next, identify your governance key results. These key results should describe how you'll sustainably enable users to answer business problems with data, while mitigating risks to data security or compliance. These governance key results should be motivated by, and closely tied to, your adoption key results.

Here are some examples of governance key results that you could define for a period.

- Reduce the number of workspaces or reports by a certain percent.
- Reduce export to Excel by a certain percent.
- Increase the number of workspaces delivering content from apps by a certain percent.
- Reduce the number of reports that are shared with the executive leadership to a specific number, or by a certain percent.
- Reduce the number of support tickets requesting access to data sources or tools by a certain percent.

> **ⓘ Important**
>
> If you don't have an effective process to monitor user activities and content, you should make this an immediate priority. An understanding of these activities and items informs better governance decisions and actions.

Implementation

Finally, identify your implementation key results. These key results should describe how you'll improve existing or future BI solutions, practices, and processes. These

implementation key results should support and align with both your adoption and governance key results.

Here are some examples of implementation key results that you could define for a period.

- Reduce the number of refresh failures by a certain percent.
- Reduce the time to investigate and resolve issues by a certain percent.
- Reduce the time to retrieve specific data or produce certain reports by a certain number of hours or days.
- Reduce the number of support incidents that relate to inaccurate data by a certain percent.
- Reduce the number of business-facing outages by a certain percent.

Checklist - When identifying your BI key results, key decisions and actions include:

- ✓ **Define how you'll measure success**: Decide if you'll use key results, KPIs, or a combination of both.
- ✓ **Identify adoption key results**: Identify key results that will help you realize your data culture vision and achieve the BI objectives for organizational adoption.
- ✓ **Identify governance key results**: Identify key results that will help you balance user enablement and risk mitigation.
- ✓ **Identify implementation key results**: Identify key results to either support defined adoption and governance key results or specific business data needs. Classify implementation key results as either initiatives or solutions.

Step 3: Define solutions and initiatives

Once you've defined your key results and you're sure that you can achieve them, you're ready to take the next step. In step 3 of tactical planning, you define solutions and initiatives you'll implement to help you achieve one or more key results.

The solutions and initiatives you'll implement have two purposes. They:

- **Support adoption and governance key results**: Describe the solutions you build and initiatives you enact to achieve your adoption and governance key results. These solutions help you work toward improving organizational adoption and user adoption.
- **Support business data needs**: Describe specific solutions you'll build to address the business data needs and priorities (such as those that are time-sensitive, quick wins, or high-impact). With these solutions, you should aim to achieve or improve solution adoption.

You can implement either solutions or initiatives.

- **Solutions**: Systems or tools built to directly address specific business problems or data needs for users. Examples of solutions include:
  - An actionable monitoring solution that allows governance teams to follow up on governance and adoption key results.
  - A unified data lakehouse that delivers business-ready data for consumption by content creators planning other downstream analytical solutions.
  - A Power BI app that addresses specific business data needs for content consumers.
- **Initiatives**: Processes, training resources, and policies that support your key results. Initiatives are typically non-technical instruments that support users or processes. Examples of initiatives include:

- Processes for self-service content creators so that they can request access to tools, data, or training.
- Governance data policies that describe how certain data should be accessed and used.
- A curated, moderated centralized portal for the user community of practice.

Here are some examples of BI objectives together with related key results and solutions or initiatives to achieve them for a specific planning period.

Expand table

Example objective	Sample key results	Example initiatives or solutions
Improve executive adoption and support of BI to promote a healthier data culture.	• Identify and engage one or more candidates for an executive sponsor. • Conduct three executive-led town hall or Q&A meetings about BI accomplishments and planned activities. • Hold three targeted mentoring sessions with the executive sponsor to improve their knowledge and understanding about relevant BI topics, and allowing them to lead by example.	• **Create a communication plan**: Create a communication plan with the Center of Excellence (COE), which will involve distributing a regular newsletter from the executive sponsor to share updates, announcements, and highlights from BI solutions and initiatives. • **Conduct an executive feedback survey**: Measure executive endorsement and sentiment with a brief survey of executives, including (but not limited to) the executive sponsor. The survey asks for quantitative feedback about the effectiveness, usability, and relevance of BI solutions.
Achieve a better balance of user enablement and risk mitigation in BI governance.	• Reduce the ratio of Power BI semantic models to reports by a certain percentage. This key result measures whether semantic models are reused for ad hoc analysis and reporting, or whether data is duplicated across models—a ratio close to one indicates that users might be creating a new semantic model for each report, which is a governance risk. • Reduce the ratio of exports to views by a certain percentage. This key result	• **Perform a tenant-wide audit**: An initial tenant-wide audit to gain visibility on general usage trends and anomalies. • **Create a tenant-wide monitoring solution**: Track critical solutions and risk-creating behaviors. • **Conduct targeted training of the top exporters**: Identify and engage the people in the user community who export data the

Example objective	Sample key results	Example initiatives or solutions
	measures how often users export data to files instead of using existing reports for their analysis—a ratio close to one indicates that users are regularly exporting data, which is a governance risk.	most frequently, and offer them several hours of training or mentorship in how to use Power BI reports or analyze in Excel pivot tables.
Improve data-driven decision making in sales teams so that users are more effective at using Power BI to make sales decisions and take actions.	• Have a certain number of users complete a data literacy training with a passing score. • Dedicate an average of four hours of the COE per week to mentorship activities of the user community with open office hours.	• **Run a data literacy training program**: Improve the data competences of the sales community. • **Hold weekly office hours**: Allow users to ask questions about central reports, or request guidance for their decentralized self-service BI solutions. • **Create a certified, centralized semantic model**: Deliver daily sales data, which sales teams can connect to in order to answer their questions and perform personal or team BI.

Create a backlog of initiatives and solutions

The working team should make a list of solutions and initiatives that will be implemented this period. For each key result, consider what initiatives or solutions will be implemented to achieve them. Then, order this list by priority, sorting each implementation from highest to lowest, so that it's clear what should be done first.

After tactical planning, content creators and owners work through this prioritized list (or *backlog*) to iteratively design and deliver BI solutions, which is the focus of the BI solution planning article.

When curating this implementation backlog, consider the following points.

- Justify the prioritization of the initiative or solution.
- Approximate the effort involved, if possible.
- Outline the anticipated scope of the initiative or solution.
- Describe relevant timelines and stakeholders.
- Refer to any existing documentation, research, or related solutions.
- Agree on who will design and implement the solution.

> **Important**
>
> While the solutions you plan aim to address the business data needs, it's unlikely you'll be able to address all of these needs immediately. Ensure that you plan to mitigate the potential impact of unmet business data needs that you won't address now. Try to assess the impact of these data needs and plan to either partially address them with quick wins or even stopgap solutions to at least temporarily alleviate the business impact.

Validate tactical planning

Once you've defined key results, solutions, and initiatives, you should get approval from executives and the key stakeholders before enacting your tactical planning. Present the outcomes of tactical planning to executives and key stakeholders. Highlight the expected benefits and relevant outcomes for the business should tactical planning be successful. Also, explain how the described BI key results support the business objectives and data needs identified in BI strategic planning. Use any feedback to adjust tactical planning, where necessary.

Checklist - When considering your desired future state, key decisions and actions include:

- ✓ **Create a list of initiatives and solutions to implement**: These initiatives and solutions should support one or more of your key results.
- ✓ **Prioritize the solution backlog**: Rank the list of initiatives and solutions from highest to lowest priority, so that you know what must be done first.
- ✓ **Define initial planning for each implementation**: Define the initial estimated scope, timelines, and responsible teams or individuals for implementing these intiatives or solutions.
- ✓ **Validate tactical planning**: Share the outcomes of tactical planning with executives and key stakeholders. Use feedback to revise tactical planning, where necessary.

Step 4: Periodically revise the plan

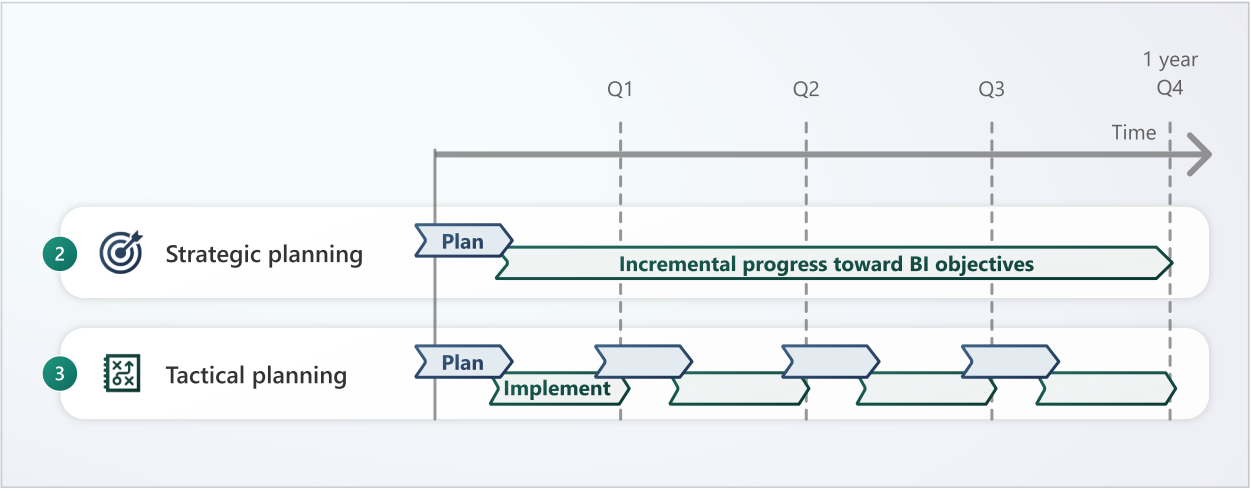
The business and technology context of your organization regularly changes. As such, you should periodically reevaluate and reassess your BI strategy and tactical planning.

The objective is to keep them relevant and useful for your organization. In step 4 of tactical planning, you take practical steps to iteratively reevaluate and reassess planning.

Prepare iterative planning and anticipate change

To ensure BI and business strategic alignment, you should establish continuous improvement cycles. These cycles should be influenced by the success criteria (your KPIs or OKRs) and the feedback that you regularly collect to evaluate progress.

We recommend that you conduct tactical planning at regular intervals with evaluation and assessment, as depicted in the following diagram.



The diagram depicts how you can iteratively revise the BI strategy to achieve incremental progress.

Expand table

Item	Description
1	**BI strategic planning**: Define and reassess your BI focus areas and objectives every 12-18 months. In between strategic planning sessions, strive for incremental progress toward your strategic objectives by achieving your key results defined in tactical planning. Additionally, in between strategic planning, you should collect feedback to inform future strategic decision-making.
2	**BI tactical planning**: Identify and reevaluate your key results quarterly, or every 1-3 months. In between, you implement these tactical plans by building BI solutions and launching BI initiatives. Additionally, in between tactical planning, you should collect feedback and monitor your KPIs or OKRs to inform future tactical decision-making.

Future key results and focus areas defined in your strategic and tactical planning are informed by using regular feedback and evaluation mechanisms, such as those described in the following sections.

Collect feedback about the business strategy

Business key results regularly change, resulting in new business data needs and changing requirements. For this reason, your tactical planning must be flexible and remain well aligned with the business strategy. To enable this alignment, you can:

- **Schedule business alignment meetings**: When conducting tactical planning, schedule strategic meetings with key business and data decision makers to assess what was done in the previous period. You should schedule these meetings to align with other key strategic business meetings. Discussions during these meetings provide an opportunity to revise strategic and tactical planning, and also to demonstrate and reflect upon progress.
- **Review feedback and requests**: Feedback and requests from the user community is valuable input to reevaluate your BI strategy. Consider setting up a communication hub, possibly with channels like *office hours*, or feedback forms to collect feedback.
- **Couple tactical planning with project planning**: Tactical planning can be integrated with your project planning processes. For example, you might integrate tactical planning with your *agile planning* processes. Agile planning is a project management approach that focuses on delivering value through iterative work cycles. Coupling tactical and agile planning helps to create a consistent, iterative structure around the operationalization of your BI strategy.

Tip

Creating structured processes to handle changing business key results can help to avoid reactive or spontaneous planning, especially to meet new, urgent business requests.

Anticipate change in technology

Tactical planning should address relevant technological changes. Technological changes can strongly impact on your planning and business processes. Many changes are also opportunities to improve existing or planned implementations. It's important to ensure that your planning is always current to ensure that you can best use the opportunities new technology provide. Not only does it help people remain effective, it helps your organization remain competitive in its market.

Here are some examples of technological changes that can affect your tactical planning.

- New products, features, or settings (including those in preview release)
- Decommissioned tools, systems, or features
- Changes in how the user community use tools or analyze data (such as generative AI)

To mitigate impact and capitalize on opportunities of change, you should regularly examine the technological context of your business. Consider the following points about responding to technological change.

- **Follow updates**: Keep current with new developments and features in Microsoft Fabric. Read the monthly community blog posts and keep pace with announcements at conference events.
- **Document key changes**: Ensure that any impactful changes are included in your tactical planning, and include relevant references. Call attention to any changes that have a direct or urgent impact on business data needs or BI key results.
- **Decide how to handle features in preview**: Clarify how you'll use new preview features that aren't yet generally available. Identify any preview features or tools that have a strategic impact in your organization or help you achieve strategic key results. Consider how you'll benefit from these preview features while identifying and mitigating any potential risks or limitations.
- **Decide how to handle new third-party and community tools**: Clarify your policy about third-party and community tools. If these tools are allowed, describe a process to identify new tools that have a strategic impact in your organization or help you achieve strategic key results. Consider how you'll benefit from these tools while identifying and mitigating any potential risks or limitations.

Proceed with solution planning

A key outcome of tactical planning is a prioritized backlog of key solutions and initiatives that you'll implement to help you achieve one or more key results. The next step is to plan and implement these initiatives or solutions.

Checklist - When planning to revise your strategic and tactical planning, key decisions and actions include:

- ✓ **Schedule periodic planning workshops**: At the end of each planning period, schedule workshops to assess progress and review the milestones attained.
- ✓ **Schedule regular workshops to re-align with business strategy**: Use workshops to align the BI strategy with the business strategy by having a cross-functional discussion between the working team and key stakeholders.
- ✓ **Create mechanisms for assessment and feedback**: Ensure that feedback relevant to the BI strategy is documented. Create forms, or encourage key stakeholders to use the communication hub to provide feedback and submit new requests.
- ✓ **Assign a team to own feedback**: Ensure that there's a team that has clear ownership of user feedback and requests. This team should respond to users to acknowledge their requests or request more detail.
- ✓ **Create a schedule to review requests**: Review feedback regularly, like every week. Identify priority requests before they become urgent and disrupt existing planning. Clearly and transparently communicate any rejected requests to users. Propose alternatives and workarounds so that users can continue their work without disruption.

Related content

In the next article in this series, learn how to conduct BI solution planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: BI solution planning

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to plan solutions that support your business intelligence (BI) strategy. It's primarily targeted at:

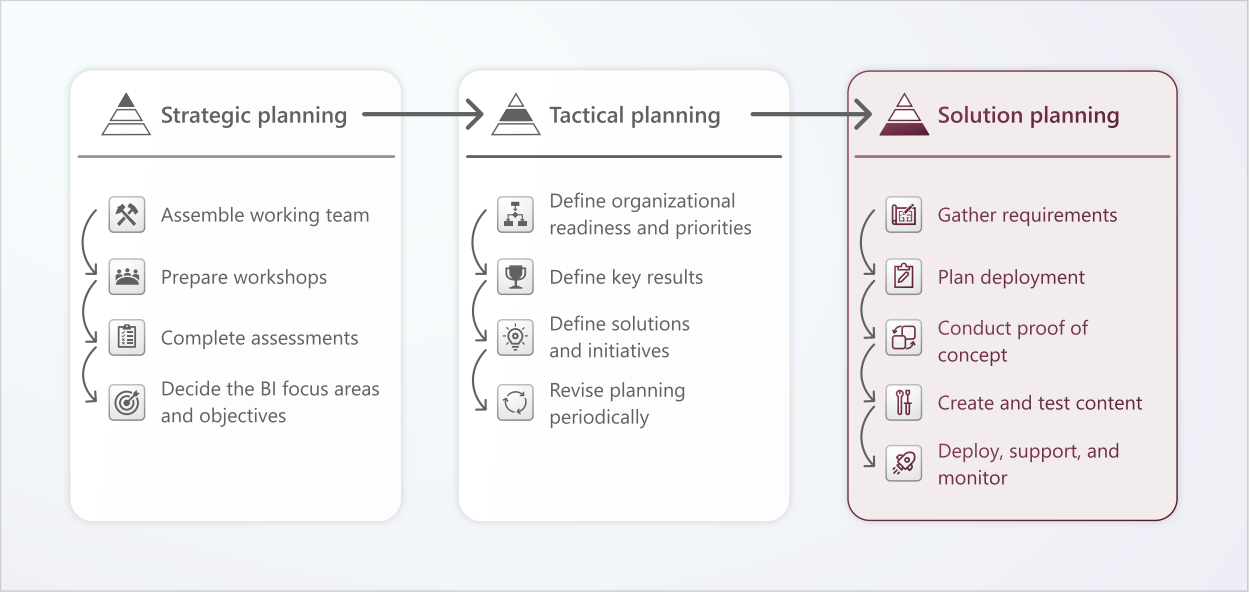
- **BI and analytics directors or managers**: Decision makers who are responsible for overseeing the BI program and strategically important BI solutions.
- **Center of Excellence (COE), IT, and BI teams**: The teams that design and deploy enterprise BI solutions for their organization.
- **Subject matter experts (SMEs) and content owners and creators**: The teams and individuals that champion analytics in a department and design and deploy solutions for self-service, departmental BI, or team BI usage scenarios.

A BI strategy is a plan to implement, use, and manage data and analytics. You define your BI strategy by starting with BI strategic planning. Strategic planning helps you to identify your BI focus areas and objectives. To determine the path to progress toward your BI objectives, you describe specific key results by using tactical planning. You then achieve progress toward these key results by planning and deploying BI solutions.

> **Note**
>
> In the objectives and key results (OKRs) framework, *objectives* are clear, high-level descriptions of what you want to achieve. In contrast, *key results* are specific, achievable outcomes to measure progress toward one of your objectives.
>
> Further, *initiatives* or *solutions* are processes or tools built to help you achieve one or more key results. Solutions address specific data needs for users. A solution can take many forms, such as a data pipeline, a data lakehouse, or a Power BI semantic model or report.
>
> For more about OKRs, see Get to know OKRs (Microsoft Viva Goals).

There are many approaches to plan and implement BI solutions. This article describes one approach that you can take to plan and implement BI solutions that support your BI strategy.

The following high-level diagram depicts how to conduct BI solution planning.



You take the following steps to conduct BI solution planning.

Expand table

Step	Description
1	Assemble a project team that gathers requirements and defines the design of the solution.
2	Plan for solution deployment by performing initial setup of tools and processes.
3	Conduct a solution proof of concept (POC) to validate assumptions about the design.
4	Create and validate content by using iterative development and validation cycles.
5	Deploy, support, and monitor the solution after it's released to the production environment.

> **Note**
>
> BI solution planning applies to both self-service BI and enterprise BI projects.

For more information, see the Power BI migration series. While the series is concerned with migration, the key actions and considerations are relevant to solution planning.

Step 1: Gather requirements

You commence solution planning by first gathering requirements and defining the solution design.

Note: Strategic and tactical planning is led by a *working team*, which leads the initiative. In contrast, solution planning is led by a *project team*, which consists of content owners and creators.

Gathering the right requirements is critical to achieve successful solution deployment and adoption. An effective way to gather requirements is to identify and involve the right stakeholders, collaboratively define the problem to be solved, and use that shared understanding of the problem to create a solution design.

Here are some benefits from using a collaborative approach to gather requirements.

- **User input produces more useful designs**: By engaging users in focused discussions to collect requirements, you can more effectively capture business data needs. For example, users can demonstrate to content creators how they use existing solutions and provide feedback about the perceived effectiveness of these solutions.
- **Avoid assumptions and mitigate change requests**: Discussions with users often reveal nuances, exceptions, and hidden complexities. These insights reduce the likelihood of late-stage requests, which can be costly to address.
- **Onboarding users increases solution adoption**: By involving users in design and early development, you provide them with an opportunity to influence the final result. Involvement can also give users a sense of intellectual ownership and

accountability for the solution. Highly involved users will be more likely to endorse the solution and lead their community of practice in using it effectively.

- **Designs set expectations for stakeholders and business users**: By producing mock-ups or illustrations of the solution design, you can clearly show stakeholders what the solution will deliver. It also helps by creating a mutual understanding of the expected project result. This process is also known as *design thinking*, and it can be an effective way to approach and understand complex problems.

You can take different approaches to engage users and gather requirements. For example, you can gather requirements with *business design* and *technical design* (described in detail in later sections of this article).

Business design is an approach to gather business requirements. It focuses on engaging business users in business design sessions to collaboratively design the solution. The output of a business design consists of solution mock-ups and descriptive design documentation.

Technical design is an approach to translate business requirements to technical requirements, and to address design assumptions. A technical design focuses on validating the business design and defining a technical approach to use. To validate the design, content creators typically engage with technical experts in focused discussions called technical design sessions, where necessary.

> **ⓘ Important**
>
> Collecting the wrong requirements is a common reason why implementations fail. Often, teams collect the wrong requirements because they engaged with the wrong stakeholders, like decision makers who provide top-down requests for solutions to be built.
>
> Engaging business users by using collaborative approaches like a business design can help you collect better requirements. Better requirements often lead to more efficient development and more robust solutions.

> **ⓘ Note**
>
> For some teams, adopting a structured requirements gathering process is a big change. Ensure that you manage this change, and that it doesn't disrupt solution planning. We recommend that you find ways to adapt these approaches to fit with the way your team works.

Prepare for solution planning

You should first prepare for solution planning by considering the factors described in the following sections.

- **Identify who will conduct solution planning**: As part of the BI tactical planning, the working team created a prioritized backlog of solutions. In solution planning, a project team is responsible for designing, developing, and deploying one or more solutions from the backlog. For each solution in the backlog, you should assemble a project team that will be responsible for the solution. In addition to running BI solution planning, the project team should:
  - Define timelines and milestones for solution planning.
  - Identify and involve the right stakeholders for requirements gathering.
  - Set up a centralized location for communication, documentation, and planning.
  - Engage stakeholders to gather requirements.
  - Communicate and coordinate with stakeholders and business users.
  - Orchestrate iterative development and testing cycles with business users.
  - Document the solution.
  - Onboard users to the solution by defining and enacting a training plan.
  - Provide post-deployment solution support.
  - Address reasonable user requests to change or update the solution after deployment.
  - Conduct solution handover after deployment, if necessary.
- **Centralize communication and documentation**: It's important that the project team centralizes communication and documentation for BI solution planning. For example, the project team should centralize requirements, stakeholder communication, timelines, and deliverables. Consider storing all documentation in a centralized portal.
- **Plan requirements gathering**: The project team should begin by planning the business design sessions to gather business requirements. These sessions take the form of interactive meetings, and they can follow a similar format to the strategic planning workshops.

> **Tip**
>
> Consider identifying and involving the support teams responsible for the solution early in the requirements gathering process. To effectively support the solution, the support teams will need a comprehensive understanding of the solution, its purpose, and the users. That's particularly important when the project team is comprised only of external consultants.

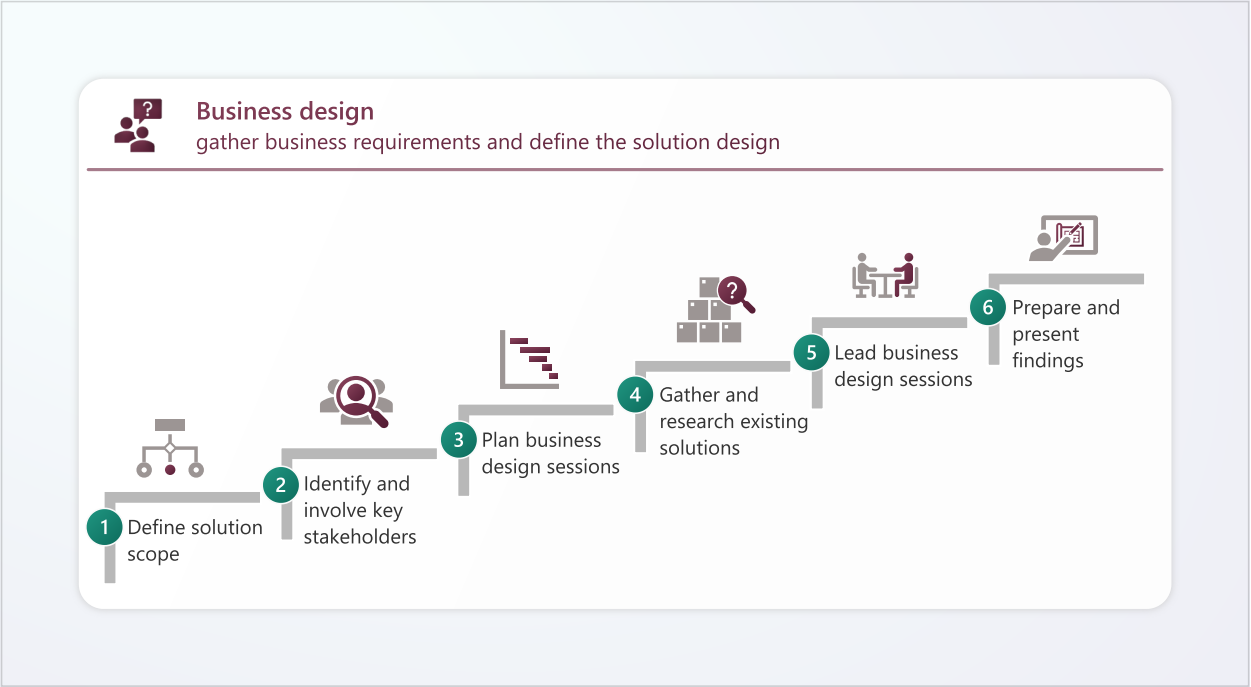
Gather business requirements

Gathering the right business requirements is critical to designing the right solution. To gather the right requirements and define an effective solution design, the project team can conduct business design sessions together with the business users.

The purpose of the business design sessions is to:

- Confirm the initial solution scope.
- Define and understand the problem the solution should address.
- Identify the right key stakeholders for the solution.
- Gather the right business requirements.
- Prepare a solution design that meets the business requirements.
- Prepare supporting design documentation.

The following diagram depicts how to gather business requirements and define the solution design by using a business design approach.



The diagram depicts the following steps.

Expand table

Item	Description
1	The project team begins the business design by confirming the solution scope that was first documented in tactical planning. They should clarify the business areas, systems, and data that the solution will encompass.
2	The project team identifies key stakeholders from the user community who will be involved in the business design sessions. Key stakeholders are users with sufficient

Item	Description
	knowledge and credibility to represent the subject areas of the solution.
3	The project team plans business design sessions. Planning involves informing stakeholders, organizing meetings, preparing deliverables, and engaging with business users.
4	The project team gathers and researches existing solutions that business users currently use to address existing business data needs. To accelerate this process, the project team can use relevant research from BI strategic planning, which has been documented in the communication hub.
5	The project team runs business design sessions with stakeholders. These sessions are small, interactive meetings, where the project team guides stakeholders to understand business data needs and requirements.
6	The project team concludes the business design by presenting a draft solution design to stakeholders and other users for feedback and approval. The business design is successful when the stakeholders agree that the design will help them achieve their business objectives.

The business design concludes with the following deliverables.

- **Draft solution designs**: Mock-ups, prototypes, or wireframe diagrams illustrate the solution design. These documents translate the requirements to a concrete design blueprint.
- **List of business metrics**: Quantitative fields expected in the solution, including business definitions, and expected aggregations. If possible, rank them by importance to the users.
- **List of business attributes**: Relevant attributes and data structures expected in the solution, including business definitions and attribute names. If possible, include hierarchies and rank the attributes by importance to the users.
- **Supplemental documentation**: Descriptions of key functional or compliance requirements. This documentation should be as precise as necessary, yet as concise as possible.

The business design deliverables are used in, and validated by, the technical design.

> **Tip**
>
> Solution mock-ups, prototypes, or wireframe diagrams can create a clear understanding of the expected result, both for developers and end users. Creating effective mock-ups doesn't require artistic skill or talent. You can use simple tools

like **Microsoft Whiteboard**, PowerPoint, or even just a pen and paper to illustrate the design.

Gather technical requirements

After completing the business design, the project team validates its outcomes by using a technical design. The technical design is an approach similar to the business design. While the business design focuses on business data needs, the technical design focuses on the technical aspects of a solution. A key outcome of the technical design is the solution plan, which describes the final solution design and informed estimates of the effort to implement it.

> **Note**
>
> Unlike the business design, the technical design is largely an independent investigation into source data and systems conducted by content creators and owners.

The purpose of a technical design is to:

- Validate the results of the business design.
- Address technical assumptions in the current design.
- Identify the relevant data sources in scope, and define the field calculations and field-source mappings for each data source.
- Translate the business requirements to technical requirements.
- Produce estimations of the effort required for the implementation.

The project team engages technical or functional stakeholders in limited, focused technical design sessions. These sessions are interactive meetings with the functional stakeholders to gather technical requirements. Stakeholders are responsible for specific functional areas required for the solution to work effectively.

Examples of stakeholders in a technical design could be:

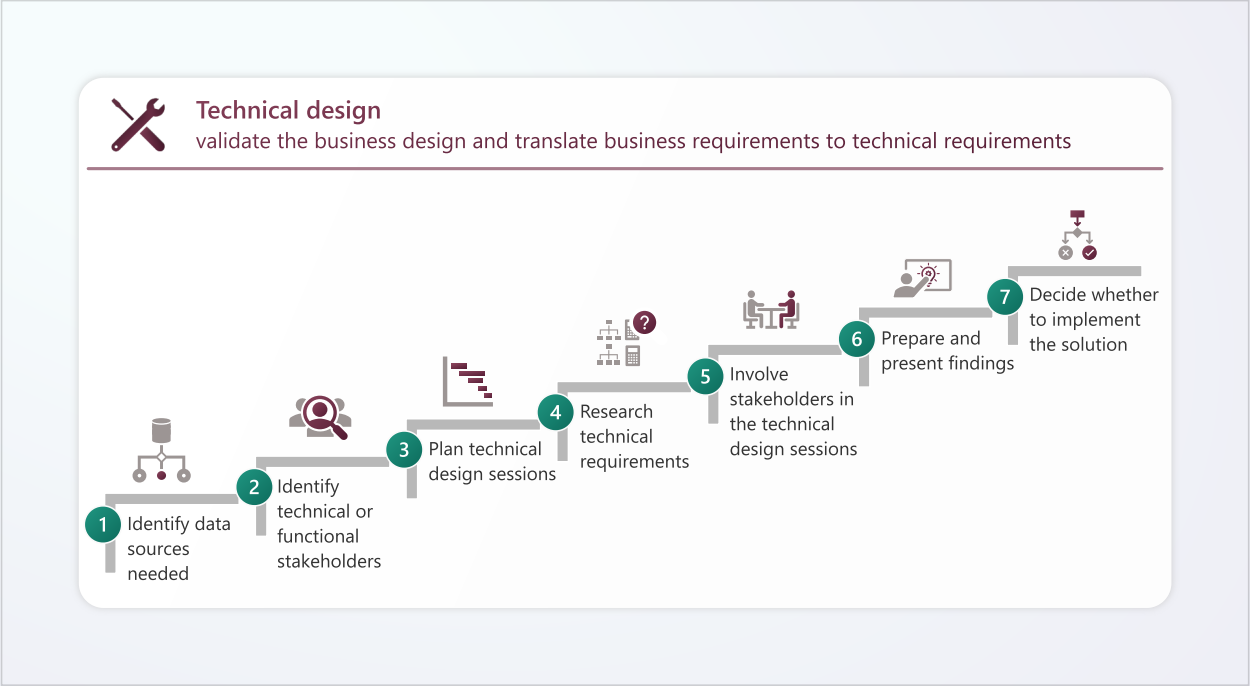
- **Security and networking teams**: Responsible for ensuring security and compliance of the data.
- **Functional teams and data stewards**: Responsible for curating the source data.
- **Architects**: Owners of specific platforms, tools, or technology.

The project team engages stakeholders in technical design sessions to address technical aspects of the solution. Technical aspects can include:

- **Data source connections**: Details about how to connect to, and integrate, data sources.
- **Networking and data gateways**: Details about private networks or on-premises data sources.
- **Field source mapping**: Data mappings of business metrics and attributes to data source fields.
- **Calculation logic**: A translation of business definitions to technical calculations.
- **Technical features**: Features or functionality needed to support business requirements.

> **Tip**
>
> The project team who conducted the business design should also conduct the technical design. However, for practical reasons, different individuals might lead the technical design. In this case, begin the technical design by reviewing the outcomes of the business design.
>
> Ideally, the individuals who lead the technical design should have a thorough understanding of the outcomes and the business users.

The following diagram depicts how to translate business requirements into technical requirements by using a technical design.



The diagram depicts the following steps.

Expand table

Item	Description
1	The project team begins the technical design by defining the data source scope based on the results of the business design. The data source scope declares which data are required to build the solution. To identify the right data sources, the project team consults with the business and functional SMEs.
2	The project team identifies technical or functional stakeholders to involve later in the technical design sessions.
3	The project team plans limited, focused sessions with functional stakeholders to address technical aspects of the solution. Planning involves informing stakeholders, organizing meetings, and preparing deliverables.
4	The project team researches technical requirements. Research includes defining field calculations and data source mappings, and also addressing the business design assumptions with technical analysis and documentation.
5	If necessary, the project team involves stakeholders in technical design sessions. Sessions focus on a specific, technical aspect of the solution, like security or data source connections. In these sessions, the project team gathers qualitative feedback from stakeholders and SMEs.
6	The project team prepares their findings by using a solution plan, which they present to stakeholders and decision makers. The plan is an iteration and extension of the business design outcomes that includes the final design, estimations, and other deliverables.
7	The technical design should conclude with a final meeting with stakeholders and decision makers to decide whether or not to proceed. This meeting provides a final opportunity to evaluate the solution planning before resources are committed to developing the solution.

> **Note**
>
> The technical design might reveal unexpected complexity that could make the solution planning infeasible given the current resource availability or organizational readiness. In this case, the solution should be reevaluated in the subsequent tactical planning period. Depending on the urgency of the business data needs, a decision maker, like the executive sponsor, might still want to proceed with a proof of concept, or only one part of the planned solution.

The technical design concludes with a solution plan, which consists of the following deliverables.

- **Tools and technologies**: List of the relevant technical instruments needed to implement the solution. The list typically includes relevant new license options (like Fabric capacity or Premium per user), features, and tools.

- **Defined list of business metrics**: Calculations and field-source mappings of the business metrics for all of the in-scope data sources. To produce this deliverable, the project team uses the list of business metrics created in the business design.
- **Defined list of business attributes**: Field-source mappings of the business attributes for all of the in-scope data sources. To produce this deliverable, the project team uses the list of business attributes created in the business design.
- **Revised designs**: Revisions to the solution design based on changes to, or invalid assumptions about, the business design. Revised designs are updated versions of the mock-ups, prototypes, or wireframe diagrams produced in the business design. If no revisions are necessary, communicate that the technical design validates the business design.
- **Estimate of effort**: Estimate of the resources needed to develop, support, and maintain the solution. The estimate informs the final decision about whether to proceed with implementing the solution, or not.

> **Important**
>
> Ensure that the project team notifies stakeholders of any changes or unexpected discoveries from the technical design. These technical design sessions should still involve relevant business users. However, ensure that stakeholders aren't unnecessarily exposed to complex technical information.

> **Tip**
>
> Because business objectives invariably evolve, it's expected that requirements will change. Don't assume that requirements for BI projects are fixed. If you struggle with changing requirements, it might be an indication that your requirements gathering process isn't effective, or that your development workflows don't sufficiently incorporate regular feedback.

Checklist - When gathering requirements, key decisions and actions include:

- ✓ **Clarify who owns solution planning**: For each solution, ensure that roles and responsibilities are clear for the project team.
- ✓ **Clarify the solution scope**: The solution scope should already be documented as part of BI tactical planning. You might need to spend additional time and effort to

clarify the scope before you start solution planning.

- ✓ **Identify and inform stakeholders**: Identify stakeholders for business designs and technical designs. Inform them in advance about the project and explain the scope, goals, required time investment, and deliverables from the business design.
- ✓ **Plan and conduct business design sessions**: Moderate the business design sessions to elicit information from stakeholders and business users. Request that users demonstrate how they use existing solutions.
- ✓ **Document business metrics and attributes**: By using existing solutions and input from stakeholders, create a list of business metrics and attributes. In the technical designs, map the fields to the data source and describe the calculation logic for quantitative fields.
- ✓ **Draft the solution design**: Create iterative mock-ups based on stakeholder input that visually reflect the expected solution result. Ensure that mock-ups accurately represent and address the business requirements. Communicate to business users that the mock-ups must still be validated (and possibly revised) during the technical design.
- ✓ **Create the solution plan**: Research source data and relevant technical considerations to ensure that the business design is achievable. Where relevant, describe key risks and threats to the design, and any alternative approaches. If necessary, prepare a revision of the solution design and discuss it with the stakeholders.
- ✓ **Create effort estimates**: As part of the final solution plan, estimate the effort to build and support the solution. Justify these estimates with the information gathered during the business design and technical design sessions.
- ✓ **Decide whether to proceed with the plan**: To conclude the requirements gathering process, present the final plan to stakeholders and decision makers. The purpose of this meeting is to determine whether to proceed with solution development.

Step 2: Plan for deployment

When the project team finishes gathering requirements, creating the solution plan, and receiving approval to proceed, it's ready to plan for solution deployment.

Deployment planning tasks differ depending on the solution, your development workflow, and your deployment process. A deployment plan typically pertains to many activities involving the planning and setup of tools and processes for the solution.

Plan to address key areas

The project team should plan for key areas of solution deployment. Typically, planning should address the following areas.

- **Compliance**: Ensure that all the compliance criteria identified in requirements gathering will be addressed by specific actions. Assign each of these actions to specific people, and clearly define the delivery timeframe.
- **Security**: Decide how different layers of solution access will be managed, and any data security rule requirements. Review whether the solution security will be more or less strict than standard content in the tenant.
- **Data gateways**: Evaluate whether the solution needs a data gateway to connect to data sources. Determine whether specific gateway settings or high availability clusters are necessary. Plan who will be able to manage gateway connections via the gateway security roles, and how to monitor the gateways. For more information, see Provision gateway access.
- **Workspaces**: Decide how to set up and use workspaces. Determine whether the solution requires lifecycle management tools like Git integration and deployment pipelines, and whether it requires advanced logging with Azure Log Analytics.
- **Support**: Establish who's responsible for supporting and maintaining the solution after production deployment. If the individuals responsible for support are

different than the project team, involve these individuals in development. Ensure that whoever will support the solution understands the solution design, the problem it should address, who should use it, and how.

- **User training**: Anticipate the efforts needed to train the user community so they can effectively use the solution. Consider whether any specific change management actions are necessary.
- **Governance**: Identify any potential governance risks for the solution. Anticipate the effort needed to enable users to effectively use the solution, while mitigating any governance risk (for example, by using sensitivity labels and policies).

Conduct initial setup

The project team should perform initial set up to commence development. Initial set up activities can include:

- **Initial tools and processes**: Perform first-time setup for any new tools and processes needed for development, testing, and deployment.
- **Identities and credentials**: Create security groups and service principals that will be used to access tools and systems. Effectively and securely store the credentials.
- **Data gateways**: Deploy data gateways for on-premises data sources (enterprise mode gateways) or data sources on a private network (virtual network, or VNet, gateways).
- **Workspaces and repositories**: Create and set up workspaces and remote repositories for publishing and storing content.

> **Note**
>
> Deployment planning differs depending on the solution and your preferred workflow. This article describes only on high-level planning and actionable items.

For more information about deployment planning, see Plan deployment to migrate to Power BI.

Checklist - When planning solution deployment, key decisions and actions include:

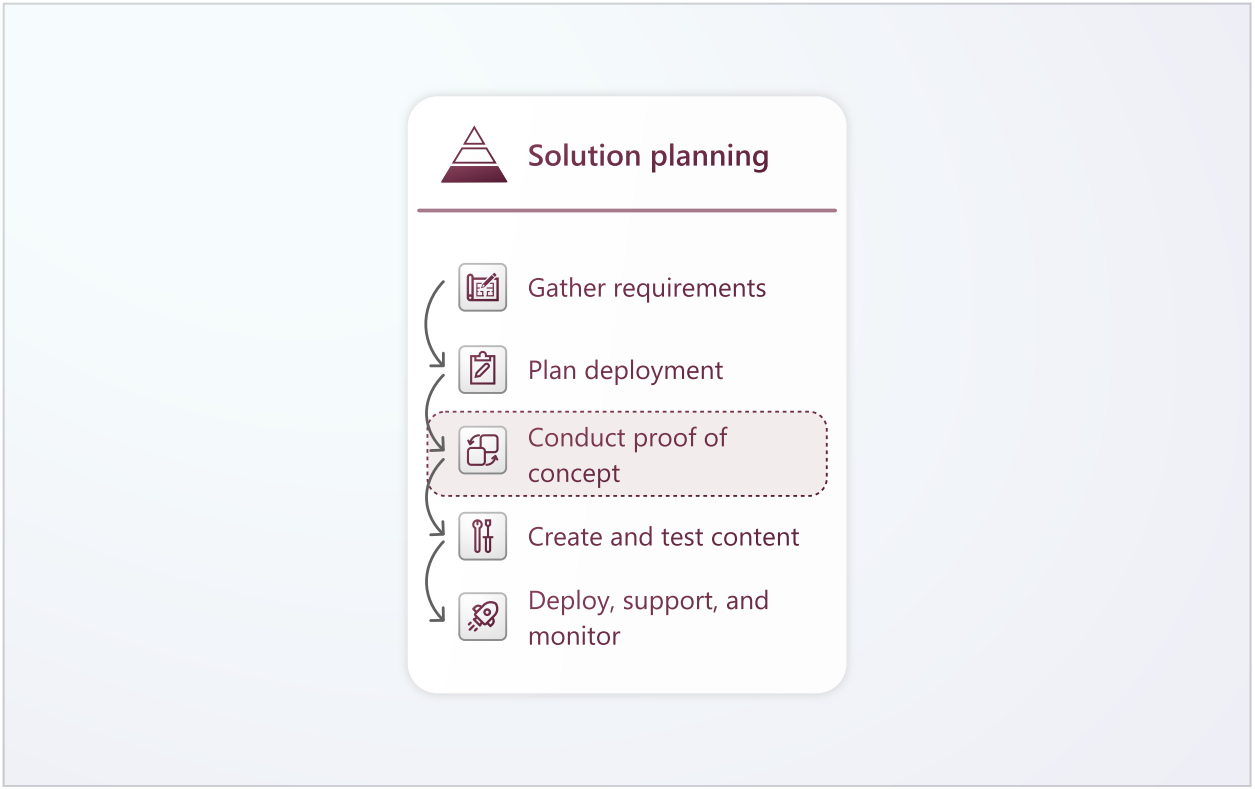
- ✓ **Plan for key areas**: Plan to address the processes and tools that you need to successfully develop and deploy your solution. Address both technical areas (like

data gateways or workspaces) and also adoption (like user training and governance).

- ✓ **Conduct initial setup**: Establish the tools, processes, and features that you need to develop and deploy the solution. Document the setup to help others who will need to do a first-time setup in the future.
- ✓ **Test data source connections**: Validate that the appropriate components and processes are in place to connect to the right data to start the proof of concept.

Step 3: Conduct a proof of concept

The project team conducts a solution *proof of concept (POC)* to validate outstanding assumptions and to demonstrate early benefits for business users. A POC is an initial design implementation that's limited in scope and maturity. A well-run POC is particularly important for large or complex solutions because it can identify and address complexities (or exceptions) that weren't detected in the technical design.



We recommend factoring in the following considerations when preparing a POC.

- **Goals and scope**: Describe the purpose of the solution POC and the functional areas it will address. For example, the project team could decide to limit the POC to a single functional area, or a specific set of requirements or features.
- **Source data**: Identify what data will be used in the POC. Depending on the solution, the project team might decide to use different types of data, such as:
  - Production (real) data
  - Sample data

    - Generated synthetic data that resembles actual data volumes and complexity observed in production environments
- **Demonstration**: Describe how and when the project team will demonstrate the POC to stakeholders and users. Demonstrations could be given during regular updates, or when the POC fulfills specific functional criteria.
- **Environment**: Describe where the project team will build the POC. A good approach is to use a distinct *sandbox* environment for the POC, and deploy it to a development environment when it's ready. A sandbox environment has more flexible policies and fluid content, and it's focused on producing quick results. In contrast, a development environment follows more structured processes that enable collaboration, and it focuses on completing specific tasks.
- **Success criteria**: Define the threshold for when the POC is successful and should move to the next iteration and enter formal development. Before starting the POC, the project team should identify clear criteria for when the POC is successful. By setting these criteria in advance, the project team defines when the POC development ends and when iterative development and validation cycles begin. Depending on the goals of the POC, the project team could set different success criteria, such as:
    - Approval of the POC by stakeholders
    - Validation of features or functionality
    - Favorable review of the POC by peers after a fixed development time
- **Failure**: Ensure that the project team can identify failure of the POC. Identifying failure early on will help to investigate root causes. It can also help to avoid further investment in a solution that won't work as expected when it's deployed to production.

> **⊗ Caution**
>
> When the project team conducts the POC, they should remain alert for assumptions and limitations. For example, the project team can't easily test solution performance and data quality by using a small set of data. Additionally, ensure that the scope and purpose of the POC is clear to the business users. Be sure to communicate that the POC is a first iteration, and stress that it's not a production solution.

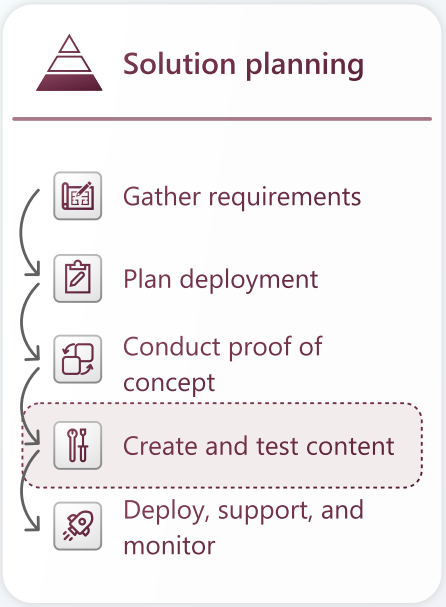
> **ⓘ Note**
>
> For more information, see Conduct proof of concept to migrate to Power BI.

Checklist - When creating a POC, key decisions and actions include:

- ✓ **Define the goals**: Ensure that the goals of the POC are clear to all people who are involved.
- ✓ **Define the scope of the POC**: Ensure that creating the POC won't take too much development effort, while still delivering value and demonstrating the solution design.
- ✓ **Decide what data will be used**: Identify what source data you'll use to make the POC, justifying your decision and outlining the potential risks and limitations.
- ✓ **Decide when and how to demonstrate the POC**: Plan to show progress by presenting the POC to decision makers and business users.
- ✓ **Clarify when the POC ends**: Ensure that the project team decides on a clear conclusion for the POC, and describe how it'll be promoted to formal development cycles.

Step 4: Create and validate content

When the POC is successful, the project team shifts from the POC to creating and validating content. The project team can develop the BI solution with iterative development and validation cycles. These cycles consist of iterative releases, where the project team creates content in a development environment and releases it to a test environment. During development, the project team gradually onboards the user community in a pilot process to early (beta) versions of the solution in the test environment.

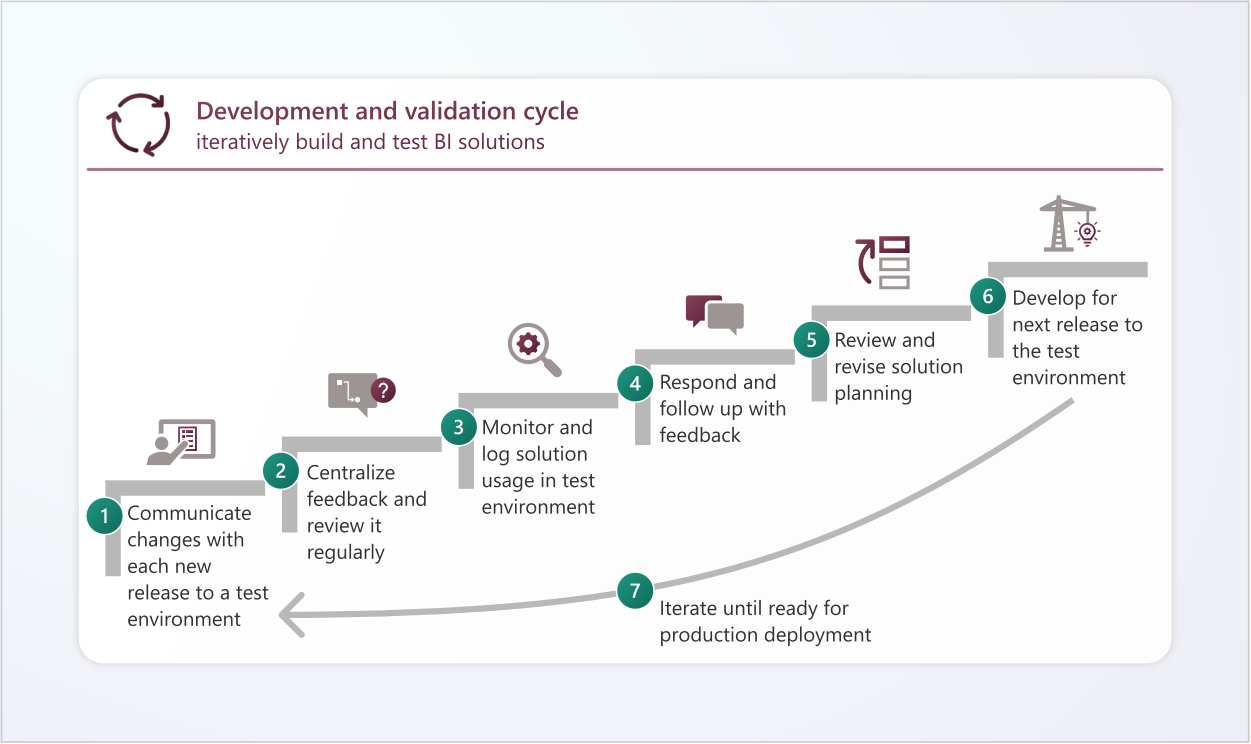


Tip

Iterative delivery encourages early validation and feedback that can mitigate change requests, promote solution adoption, and realize benefits before the production release.

Iterative development and validation cycles proceed until the project team arrives at a predefined conclusion. Typically, development concludes when there are no more features to implement or user feedback to address. When the development and validation cycles conclude, the project team deploys the content to a production environment with the final production release.

The following diagram depicts how the project team can iteratively deliver BI solutions with development and validation cycles.



The diagram depicts the following steps.

Expand table

Item	Description
1	The project team communicates each release to the user community, describing changes and new features. Ideally, communication includes a solution demonstration and Q&A, so users understand what's new in the release, and they can provide verbal feedback.
2	During validation, users provide feedback via a central tool or form. The project team should review feedback regularly to address issues, accept or reject requests, and inform upcoming development phases.
3	The project team monitors usage of the solution to confirm that users are testing it. If there isn't any usage, the project team should engage with the user community to understand the reasons why. Low usage can indicate that the project team needs to take further enablement and change management actions.
4	The project team promptly responds to user feedback. If the project team takes too long to address feedback, users might quickly lose motivation to provide it.
5	The project team incorporates accepted feedback into the solution planning. If necessary, they review the planning priorities to clarify and delegate tasks before the next development phase begins.
6	The project team continues development of the solution for the next release.
7	The project team iterates through all steps until they reach a predefined conclusion, and the solution is ready for production deployment.

The following sections describe key considerations for using iterative development and validation cycles to deliver BI solutions.

Create content

The project team develops the solution by following their normal development workflow. However, they should consider the following points when creating content.

- During each development cycle, update documentation to describe the solution.
- Conclude each development cycle with an announcement to the user community. Announcements should be posted to the centralized portal, and they should provide brief descriptions of changes and new features in each release.
- With each release, consider organizing sessions to demonstrate changes and new features to the user community, and to answer any verbal questions.
- Define when iterative development and validation cycles will conclude. Ensure that there's a clear process to deploy the solution to the production environment, including a transition to support and adoption activities.

Validate content

Each iterative development cycle should conclude with content validation. For BI solutions, there are typically two kinds of validation.

- **Developer validation**: Solution testing is done by content creators and peers. The purpose of developer validation is to identify and resolve all critical and visible issues before the solution is made available to business users. Issues can pertain to data correctness, functionality, or the user experience. Ideally, content is validated by a content creator who didn't develop it.
- **User validation**: Solution testing is done by the user community. The purpose of user validation is to provide feedback for later iterations, and to identify issues that weren't found by developers. Formal user validation periods are typically referred to as user acceptance testing (UAT).

> **Important**
>
> Ensure that any data quality issues are addressed during developer validation (before UAT). These issues can quickly erode trust in the solution, and they can harm long-term adoption.

> **Tip**

When conducting user validation, consider occasional, short calls with key users. Observe them when they use the solution. Take notes about what they find difficult to use, or what parts of the solution aren't working as expected. This approach can be an effective way to collect feedback.

Factor in the following considerations when the project team validates content.

- **Encourage user feedback**: With each release, request users provide feedback, and demonstrate how they can effectively do so. Consider regularly sharing examples of feedback and requests that have led to recent changes and new features. By sharing examples, you're demonstrating that feedback is acknowledged and valued.
- **Isolate larger requests**: Some feedback items require more effort to address. Ensure that the project team can identify these items and discuss whether they'll be implemented, or not. Consider documenting larger requests to discuss in later tactical planning sessions.
- **Begin change management activities**: Train users how to use the solution. Be sure to spend extra effort on new processes, new data, and different ways of working. Investing in change management has a positive return on long-term solution adoption.

When the solution reaches a predefined level of completeness and maturity, the project team is ready to deploy it to production. After deployment, the project team transitions from iterative delivery to supporting and monitoring the production solution.

> **Note**
>
> Development and testing differ depending on the solution and your preferred workflow.

This article describes only high-level planning and actionable items. For more information about iterative development and testing cycles, see Create content to migrate to Power BI.

Checklist - When creating and validating content, key decisions and actions include:

- ✓ **Use an iterative process to plan and assign tasks**: Plan and assign tasks for each release of the solution. Ensure that the process to plan and assign tasks is flexible

and incorporates user feedback.

- ✓ **Set up content lifecycle management**: Use tools and processes to streamline and automate solution deployment and change management.
- ✓ **Create a tool to centralize feedback**: Automate feedback collection by using a solution that's simple for you and your users. Create a straightforward form to ensure that feedback is concise yet actionable.
- ✓ **Schedule a meeting to review feedback**: Meet to briefly review each new or outstanding feedback item. Decide whether you'll implement the feedback or not, who will be responsible for the implementation, and what actions to take to close the feedback item.
- ✓ **Decide when iterative delivery concludes**: Describe the conditions for when the iterative delivery cycles will conclude, and when you'll release content to the production environment.

Step 5: Deploy, support, and monitor

When ready, the project team deploys the validated solution to the production environment. The project team should take key adoption and support actions to ensure that the deployment is successful.

To ensure a successful deployment, you perform the following support and adoption tasks.

- **Communicate the final release**: The executive sponsor, a manager, or another person with sufficient authority and credibility should announce the release to the

user community. Communication should be clear, concise, and include links to the relevant solutions and support channels.

- **Conduct training for content consumers**: Training should be available for content consumers during the first weeks after release to production. Training should focus on clarifying the solution scope, answering user questions, and explaining how to use the solution.
- **Address feedback and requests**: Consider providing users with a channel to submit feedback and requests to the project team. Ensure that reasonable feedback and requests are discussed and, when appropriate, implemented during the post-deployment support period. Acting on feedback and requests after the production release is important. It indicates an agile solution that responds to changing business needs.
- **Plan to connect with the user community**: Even after the post-deployment support period ends, ensure that solution owners regularly meet with the user community. These meetings are valuable sources of feedback for revising your BI strategy. Also, they help support solution adoption by enabling users.
- **Handover actions**: Members of the project team might not be responsible for maintaining the solution. In this case, the team should identify who's responsible and perform a handover. The handover should occur soon after the release to production, and it should address both the solution and the user community.

> ⊗ **Caution**
>
> Failing to conduct an effective handover can lead to preventable issues with solution support and adoption during its lifecycle.

After deployment, the project team should plan to proceed to the next solution in the prioritized solution backlog. Ensure that you collect any new feedback and requests and make revisions to tactical planning—including the solution backlog—if necessary.

Checklist – When considering solution deployment, key decisions and actions include:

- ✓ **Create a communication plan**: Plan how to communicate the release, training, and other solution support or adoption actions. Ensure that any outages or issues are communicated and promptly addressed in the post-deployment support period.
- ✓ **Follow through with a training plan**: Train users to use the solution. Ensure that the training includes both live and recorded training sessions for several weeks after

release.

- ✓ **Conduct handover activities**: If necessary, prepare a handover from the development team to the support team.
- ✓ **Conduct solution office hours**: After the post-deployment support period, consider holding regular office hours sessions to answer questions and collect feedback from users.
- ✓ **Set up a continuous improvement process**: Schedule a monthly audit of the solution to review potential changes or improvements over time. Centralize user feedback and review feedback periodically between audits.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Tenant setup

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This tenant setup article introduces important aspects to know about setting up your Fabric tenant, with an emphasis on the Power BI experience. It's targeted at multiple audiences:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization.
- **Microsoft Entra administrators**: The team that's responsible for overseeing and managing Microsoft Entra ID.

Fabric is part of a larger Microsoft ecosystem. If your organization is already using other cloud subscription services, such as Azure, Microsoft 365, or Dynamics 365, then Fabric operates within the same Microsoft Entra *tenant*. Your organizational domain (for example, *contoso.com*) is associated with Microsoft Entra ID. Like all Microsoft cloud services, your Fabric tenant relies on your organization's Microsoft Entra ID for identity and access management.

> **Tip**
>
> Many organizations have an on-premises Active Directory (AD) environment that they synchronize with Microsoft Entra ID in the cloud. This setup is known as a hybrid identity solution, which is out of scope for this article. The important concept to understand is that users, groups, and service principals must exist in Microsoft Entra ID for a cloud-based suite of services like Fabric to work. Having a hybrid identity solution will work for Fabric. We recommend talking to your Microsoft Entra administrators about the best solution for your organization.

For more information about the responsibilities of a Fabric administrator, see Tenant administration.

Microsoft Entra tenant

Most organizations have one Microsoft Entra tenant, so it's commonly true that a Microsoft Entra tenant represents an organization.

Usually, Microsoft Entra ID is set up before a Fabric implementation begins. However, sometimes it's when you provision a cloud service that you become aware of the importance of Microsoft Entra ID.

> **Tip**
>
> Because most organizations have one Microsoft Entra tenant, it can be challenging to explore new features in an isolated way. You might qualify for a non-production developer tenant through the **Microsoft 365 Developer Program**; for details, see the **FAQ**. Alternatively, you can **sign up for a 1-month free trial or purchase a Microsoft 365 plan**.

Unmanaged tenant

A *managed tenant* has a global administrator assigned within Microsoft Entra ID. If a Microsoft Entra tenant doesn't exist for an organizational domain (for example, *contoso.com*), when the first user from that organization signs up for a Fabric trial or account, an *unmanaged tenant* is created in Microsoft Entra ID. An unmanaged tenant is also known as a shadow tenant, or a self-service-created tenant. It has a basic configuration, allowing the cloud service to work without assigning a global administrator.

To properly manage, configure, and support Fabric, a managed tenant is required. There's a process that a system administrator can follow to take over an unmanaged tenant so that they can manage it properly on behalf of the organization.

> **Tip**
>
> The administration of Microsoft Entra ID is a broad and deep topic. We recommend that you assign specific people in your IT department as system administrators to **securely manage Microsoft Entra ID** for your organization.

Checklist - When reviewing your Microsoft Entra tenant for use with Fabric, key decisions and actions include:

- ✓ **Take over tenant**: If applicable, initiate the process to take over an unmanaged tenant.
- ✓ **Confirm the Microsoft Entra tenant is managed**: Verify that your system administrators actively manage your Microsoft Entra tenant.

Tenant ID for external users

You must consider how users will access your tenant when you have external users (such as customers, partners, or vendors) or when internal users must access another tenant outside of your organization. To access a different organizational tenant, a modified URL is used.

Every Microsoft Entra tenant has a globally unique identifier (GUID) known as the *tenant ID*. In Fabric, it's known as the *customer tenant ID (CTID)*. The CTID is appended to the end of the tenant URL. You can find the CTID in the Fabric portal by opening the *About Microsoft Fabric* dialog window. It's available from the *Help & Support (?)* menu, which is located at the top-right of the Fabric portal.

Knowing your CTID is important for Microsoft Entra B2B scenarios. URLs that you provide to external users (for example, to view a Power BI report) must append the CTID parameter in order to access the correct tenant.

If you intend to collaborate with or provide content to external users, we recommend setting up custom branding. Use of a logo, cover image, and theme helps users identify which organizational tenant they're accessing.

Checklist - When granting external users permission to view your content, or when you have multiple tenants, key decisions and actions include:

- ✓ **Include your CTID in relevant user documentation**: Record the URL that appends the tenant ID (CTID) in user documentation.
- ✓ **Set up custom branding in Fabric**: In the Fabric admin portal, set up custom branding to help users identify the organizational tenant.

Microsoft Entra administrators

Fabric administrators periodically need to work with the Microsoft Entra administrators.

The following list includes some common reasons for collaboration between Fabric administrators and Microsoft Entra administrators.

- **Security groups**: You'll need to create new security groups to properly manage the Fabric tenant settings. You might also need new groups to secure workspace content or for distributing content.
- **Security group ownership**: You might want to assign a group owner to allow more flexibility in who can manage a security group. For example, it could be more efficient to permit the Center of Excellence (COE) to manage the memberships of certain Fabric-specific groups.
- **Service principals**: You might need to create a Microsoft Entra app registration to provision a service principal. Authenticating with a service principal is a recommended practice when a Fabric administrator wants to run unattended, scheduled scripts that extract data by using the admin APIs, or when embedding content in an application.
- **External users**: You'll need to understand how the settings for external (guest) users are set up in Microsoft Entra ID. There are several Fabric tenant settings related to external users, and they rely on how Microsoft Entra ID is set up. Also, certain security capabilities for the Power BI workload only work when using the planned invitation approach for external users in Microsoft Entra ID.
- **Real-time control policies**: You can choose to set up real-time session control policies, which involves both Microsoft Entra ID and Microsoft Defender for Cloud Apps. For example, you can prohibit the download of a Power BI report when it has a specific sensitivity label.

For more information, see Collaborate with other administrators.

Checklist - When considering how to work with your Microsoft Entra administrators, key decisions and actions include:

- ✓ **Identify your Microsoft Entra administrators**: Make sure you know the Microsoft Entra administrators for your organization. Be prepared to work with them as needed.
- ✓ **Involve your Microsoft Entra administrators**: As you work through your implementation planning process, invite Microsoft Entra administrators to pertinent meetings and involve them in relevant decision-making.

Location for data storage

When a new tenant is created, resources are provisioned in Azure, which is Microsoft's cloud computing platform. Your geographic location becomes the *home region* for your tenant. The home region is also known as the *default data region*.

Home region

The home region is important because:

- The performance of reports and dashboards depends, in part, on users being in proximity to the tenant location.
- There could be legal or regulatory reasons that the organization's data be stored in a specific jurisdiction.

The home region for the organization's tenant is set to the location of the first user that signs up. If most of your users are located in a different region, that region might not be the best choice.

You can determine the home region for your tenant by opening the *About Microsoft Fabric* dialog window in the Fabric portal. The region is displayed next to the *Your data is stored in* label.

You might discover that your tenant resides in a region that isn't ideal. You can use the Multi-Geo feature by creating a capacity in a specific region (described in the next section), or, you can move it. To move your tenant to another region, your global Microsoft 365 administrator should open a support request.

The relocation of a tenant to another region isn't a fully automated process, and some downtime is involved. Be sure to take into consideration the prerequisites and actions that are required before and after the move.

> **Tip**
>
> Because a lot of effort is involved, when you determine that a move is necessary, we recommend that you do it sooner rather than later.

Checklist - When considering the home region for storing data in your tenant, key decisions and actions include:

- ✓ **Identify your home region**: Determine the home region for your tenant.
- ✓ **Initiate the process to move your tenant**: If you discover that your tenant is located in an unsuitable geographic region (that can't be solved with the Multi-Geo feature), research the process to move your tenant.

Other specific data regions

Some organizations have *data residency* requirements. Data residency requirements typically include regulatory or industry requirements for storing data in a specific geographic region. *Data sovereignty* requirements are similar, but more stringent because the data is subject to the laws of the country or region in which the data is stored. Some organizations also have *data localization* requirements, which dictate that data created within certain borders needs to remain within those borders.

Regulatory, industry, or legal requirements can require you to store certain data elsewhere from the home region (described in the previous section). In these situations, you can benefit from the Multi-Geo feature by creating a capacity in a specific region. In this case, you must assign workspaces to the correct capacity to ensure that the workspace data is stored in the desired geographic location.

Multi-Geo support enables organizations to:

- Meet data residency requirements for data at rest.
- Improve the ability to locate data near the user base.

> ⓘ **Note**
>
> The Multi-Geo feature is available with any type of capacity license (except shared capacity). It's not available with Premium Per User (PPU) because data stored in workspaces assigned to PPU is always stored in the home region (just like shared capacity).

Checklist - When considering other specific data regions for your tenant, key decisions and actions include:

- ✓ **Identify data residency requirements**: Determine what your requirements are for data residency. Identify which regions are appropriate, and which users might be involved.
- ✓ **Investigate use of the Multi-Geo feature**: For specific situations where data should be stored elsewhere from the home region, investigate enabling Multi-Geo.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

> **Tip**
>
> To learn how to manage a Fabric tenant, we recommend that you work through the **Administer Microsoft Fabric** module.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: User tools and devices

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces key considerations for planning user tools and managing devices to enable and support Power BI consumers and authors in the organization. This article is targeted at:

- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who need to decide which tools to use for creating Power BI content.
- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization.
- **IT and infrastructure teams**: Technical staff who installs, updates, and manages user devices and machines.
- **Content creators and content owners**: Users who need to communicate with colleagues and make requests for what they need to have installed.

One important aspect of analytics adoption is ensuring that content consumers and content creators have the software applications they need. The effective management of tools–particularly for users who create content–leads to increased user adoption and reduces user support costs.

Requests for new tools

How you handle requests for new tools and software applications is a governance decision. Many users who are new to the organization, or are just getting started with analytics, don't know what to request. To simplify the process, consider handling the following requests together:

- Software requests
- User license requests
- Training requests

- Data access requests

Software installations are usually the responsibility of the IT department. To ensure an optimal user experience, it's critical that IT collaborate with the Center of Excellence (COE) on key decisions and processes, such as:

- **Process for users to request software installation.** There are several ways to handle software installation requests:
  - Common tools can be included in a standard machine setup. IT teams sometimes refer to it as the *standard build*.
  - Certain applications might be installed automatically based on job role. The software that's installed could be based on an attribute in the user profile in Microsoft Entra ID.
  - For custom requests, using a standard request form works well. A form (rather than email) builds up a history of requests. When prerequisites or more licenses are required, approval can be included in the workflow.
- **Process for installing software updates.** The timely installation of software updates is important. The goal is to stay as current as possible. Be aware that users can read online what's possible and might become confused or frustrated when newer features aren't available to them. For more information, see Client tools later in this article.

Checklist - When planning for how to handle requests for new tools, key decisions and actions include:

- ✓ **Decide how to handle software requests**: Clarify who's responsible for receiving and fulfilling new requests for software installation.
- ✓ **Confirm whether prerequisites are required**: Determine what organizational prerequisites exist related to training, funding, licensing, and approvals prior to requesting software to be installed.
- ✓ **Create a tracking system**: Create a system to track the status and history of software requests.
- ✓ **Create guidance for users**: Provide documentation in the centralized portal for how to request new tools and software applications. Consider co-locating this guidance with how to request licenses, training, and access to data.

Plan for consumer tools

In an organization, many users are classified as *consumers*. A consumer views content that's been created and published by others.

The most common ways that a consumer can access Power BI content include:

Expand table

Software	Target audience
Power BI service	Content consumers view content by using a web browser (such as Microsoft Edge).
Teams	Content consumers who view content that's been published to the Power BI service by using the Power BI app for Microsoft Teams. This option is convenient when users spend a lot of time in Teams. For more information, see Guide to enabling your organization to use Power BI in Microsoft Teams.
Power BI Mobile Application	Content consumers who interact with content that's been published to the Power BI service (or Power BI Report Server) using iOS, Android, or Windows 10 applications.
OneDrive/SharePoint viewer	Content consumers who view Power BI Desktop (.pbix) files that are stored in OneDrive or SharePoint by using a web browser. This option is a useful alternative to sharing the original Power BI Desktop files. The OneDrive/SharePoint viewer is most suitable for informal teams who want to provide a friendly, web-based, report consumer experience without explicitly publishing .pbix files to the Power BI service.
Custom application	Content consumers who view content from the Power BI service that's been embedded in a custom application for your organization or for your customers.

> **Note**
>
> This list isn't intended to be an all-inclusive list of ways to access Power BI content.

Because the user experience can vary slightly between different web browsers, we recommend that you document browser recommendations in your centralized portal. For more information, see Supported browsers for Power BI.

Checklist - When planning for consumer tools, key decisions and actions include:

- ✓ **Use a modern web browser**: Ensure that all users have access to a modern web browser that's supported for Power BI. Confirm that the preferred browser is updated regularly on all user devices.
- ✓ **Decide how Teams should be used with Power BI**: Determine how users currently work, and to what extent Teams integration is useful. Set the *Enable Teams integration* and the *Install Power BI app automatically* tenant settings in the Fabric admin portal according to your decision.
- ✓ **Enable and install the Teams app**: If Teams is a commonly used tool, enable the Power BI app for Microsoft Teams. Consider pre-installing the app for all users as a convenience.
- ✓ **Decide whether viewing Power BI Desktop files is permitted**: Consider whether viewing Power BI Desktop files stored in OneDrive or SharePoint is allowed or encouraged. Set the *Users can view Power BI files saved in OneDrive and SharePoint* tenant setting according to your decision.
- ✓ **Educate users**: Provide guidance and training for content creators on how to make the best use of each option, and where to securely store files. Include recommendations, such as preferred web browsers, in your centralized portal.
- ✓ **Conduct knowledge transfer with the support team**: Confirm that the support team is prepared to answer frequently asked questions from users.

Plan for authoring tools

Some users are considered *content creators*. A content creator authors and publishes content that's viewed by consumers.

There are several tools that content creators can use to author Power BI content. Some tools are targeted at *self-service* content creators. Other tools are targeted at *advanced* content creators.

> **Tip**
>
> This section introduces the most common authoring tools. However, an author doesn't need all of them. When in doubt, start by only installing Power BI Desktop.

Available tools for authoring

The following table lists the most common tools and applications that are available for content creators.

Expand table

Software	Target audience
Power BI service	Content consumers and creators who develop content by using a web browser.
Power BI Desktop	Content creators who develop data models and interactive reports that will be published to the Power BI service.
Power BI Desktop Optimized for Report Server	Content creators who develop data models and interactive reports that will be published to Power BI Report Server (a simplified on-premises report portal).
Power BI Report Builder	Report creators who develop paginated reports that will be published to the Power BI service or to Power BI Report Server.
Power BI App for Teams	Content creators and consumers who interact with content in the Power BI service, when their preference is to remain within the Microsoft Teams application.
Power BI Mobile Application	Content creators and consumers who interact with and manage content that's been published to the Power BI service (or Power BI Report Server) using iOS, Android, or Windows 10 applications.
Excel	Content creators who develop Excel-based reports in workbooks that might include PivotTables, charts, slicers, and more. Optionally, Excel workbooks can be viewed in the Power BI service when they're stored in SharePoint or OneDrive for work or school.
Third-party tools	Advanced content creators can optionally use third-party tools and extend the built-in capabilities for purposes such as advanced data model management and enterprise content publishing.

Choose an authoring tool

When choosing an authoring tool, there are some key factors you should consider. Some of the following decisions can be made once, whereas other decisions need to be evaluated for each project or solution that you create.

- **Is browser-based authoring desirable?** To improve ease of use and reduce friction, Power BI (and other Fabric workloads) supports browser-based functionality for both content consumption as well as content creation. That's an advantage because a web browser is readily available to all users, regardless of the desktop operating system they use (including Mac users).
- **What's the desired development experience?** Consider that Power BI Desktop can be used to create data models and interactive reports, whereas Power BI Report

Builder is a design tool for creating paginated reports. Also, third-party tools offer extra functionality to developers that isn't available in Power BI Desktop. Because development experience differs among tools, the requirements for each specific solution should factor into your decision on which tool to use.

- **What's the desired publishing experience?** Advanced content creators and content owners might prefer to publish content by using a third-party tool (such as ALM Toolkit to compare and merge models). The requirements for each specific solution should be considered.
- **What's the preferred way to access and/or manage semantic models?** Rather than using the standard Power Query experience, advanced content creators might prefer to read and/or write to semantic models with their tool of choice by using the XMLA endpoint. The requirements for each specific solution should be considered.
- **How easily can you keep client tools updated?** Some organizations find it challenging to install frequent updates of client applications. In this case, users might prefer to use a web browser whenever possible.
- **What are the skills and expertise of the users?** There might be existing knowledge and preferences that impact which tool is selected. This choice impacts both initial development activities, and also whoever will support users and maintain existing solutions.
- **How will versioning be managed?** Version control can be accomplished in multiple ways. When working in a client tool, self-service users might prefer to use OneDrive or SharePoint, whereas more advanced users might prefer Git integration with client tools. When working in the Power BI service, Git workspace integration is available.

> **Tip**
>
> We recommend that you adopt one method of working and then consistently use that method. For example, when content creators are inconsistent about using Power BI Desktop versus the Power BI service for report creation, it becomes much harder to determine where the original report resides and who's responsible for it.

When to use each authoring tool

The remainder of this section considers when to use the most common authoring tools.

Web-based authoring

The capabilities in the Power BI service for authoring and editing content are continually evolving (alongside capabilities for viewing, sharing, and distributing content). For content creators that use a non-Windows operating system (such as macOS, Linux, or Unix), web-based authoring in the Power BI service is a viable option. Web-based authoring is also useful for organizations that aren't capable of keeping Power BI Desktop updated on a timely basis.

> **Note**
>
> Because the Power BI service is a web application, Microsoft installs all updates to ensure it's the latest version. That can be a significant advantage for busy IT teams. However, it's also important to closely monitor when releases occur so that you're informed about feature changes.

There are some types of Power BI items that can be created in the web-based experience, such as:

- Dataflows
- Datamarts
- Paginated reports
- Power BI reports
- Dashboards
- Scorecards

A Fabric solution can be created end-to-end in a browser. The solution can include Power BI items, and also non-Power BI items (such as a lakehouse).

> **Important**
>
> When choosing to create content in the browser, it's important that you educate content creators where to save content. For example, it's easy to save a new report to a personal workspace, but that's not always an ideal choice. Also, it's important to consider how versioning will be handled (such as Git integration).

Power BI Desktop

Because it's a free application, Power BI Desktop is a great way for content creators to get started with developing data models and creating interactive reports. Power BI Desktop allows you to connect to many data sources, combine data from multiple data sources, clean and transform data, create a data model, add DAX calculations, and build

reports within a single application. Power BI Desktop is well-suited to building *interactive reports* with a focus on exploration.

Here are some points to consider when using Power BI Desktop.

- You can create reports within Power BI Desktop or the Power BI service. Due to this flexibility, a consistent process for how and where to develop content is necessary.
- Using version control is considered a best practice. One option for self-service content creators is to save files created by Power BI Desktop in a location with versioning enabled (such as OneDrive or SharePoint) that can be secured to authorized users. Advanced content creators might prefer to use Git integration.
- Power BI Desktop is available as a Windows desktop application. Optionally, it's possible to run Power BI Desktop in a virtualized environment.
- Power BI Desktop is usually updated every month. Regular updates allow users to access new features quickly. However, rolling out frequent updates in a large organization requires planning. For more information, see Client tools later in this article.

> ⓘ **Note**
>
> There are many options and settings in Power BI Desktop that significantly affect the user experience. Not all settings can be programmatically maintained with group policy or registry settings (described later in this article). One key setting relates to preview features that users can enable in Power BI Desktop. However, preview features are subject to change, have limited support, and might not always work in the same way in the Power BI service (during the preview period).
>
> We recommend that you only use preview features to evaluate and learn new functionality. Preview features shouldn't be used for mission-critical production content.

Power BI Desktop for Report Server

Like the standard version of Power BI Desktop, content creators can use Power BI Desktop for Report Server to create .pbix files. It supports publishing content to Power BI Report Server. New versions align with the release cadence of Power BI Report Server, which is usually three times per year.

It's important that content creators use the correct report server version of Power BI Desktop to avoid compatibility issues after content has been published to Power BI

Report Server. You can manually download and install Power BI Desktop for Report Server from the Microsoft Download Center.

For users who publish content to *both* the Power BI service and Power BI Report Server, there are two options.

- **Option 1**: Only use Power BI Desktop for Report Server because it produces files that can be published to both the Power BI service and the report server. New authoring features will become available to users approximately every four months (to remain consistent with the Power BI Report Server release cadence).
  - **Pros**:
    - Content creators only need to use one tool.
    - Content creators are assured that the content they publish is compatible with the report server.
    - Fewer tools are simpler to manage.
  - **Cons**:
    - Some features that are only supported in the Power BI service aren't available in the Report Server version of Power BI Desktop. Therefore, content creators might find it limiting.
    - New features are slower to become available.
    - Preview features aren't available.
- **Option 2**: Run both versions—Power BI Desktop, and Power BI Desktop for Report Server—side by side.
  - **Pros**:
    - All features in standard Power BI Desktop are available to be used.
    - New features for the standard Power BI Desktop are available more quickly.
    - Preview features for the standard Power BI Desktop are available to use, at the content creator's discretion.
  - **Cons**:
    - Content creators must be prepared for complexity because they need to remember which version to use when, based on the target deployment location. The risk is that when a .pbix file from the newer version is inadvertently published to Power BI Report Server, it might not function correctly. For example, data model queries fail, data refresh fails, or reports don't render properly.
    - Content creators need to be aware of the default behavior when they directly open .pbix files (instead of opening them from within Power BI Desktop).

Microsoft Excel

Many business users are proficient with Microsoft Excel and want to use it for data analysis by using PivotTables, charts, and slicers. There are other useful Excel features as well (such as cube functions) that allow greater flexibility and formatting when designing a grid layout of values. Some content creators might also prefer to use Excel formulas for some types of calculations (instead of DAX calculations in the data model), particularly when they perform data exploration activities.

Here are several ways to efficiently use Excel with Power BI.

- **Connect Excel to a Power BI semantic model**: This capability is known as an *Excel live connection* (when you start from Excel) or as *Analyze in Excel* (when you start from the Power BI service). Connecting Excel to a Power BI semantic model is best suited to report creators who prefer using Excel for creating visualizations that are connected to an existing shared semantic model. The advantage of this approach is that it's a connection—rather than an export of data—so the data in the Excel workbook is refreshable.
- **Connect Excel to featured tables in a Power BI semantic model**: If you prefer to connect Excel to a subset of tables within a Power BI semantic model (rather than the entire shared semantic model), you can use featured tables. This option works well when you need to relate data that's in Excel to data that's stored in Power BI.
- **Export to Excel with a live connection**: When viewing a visual, you can export a table of refreshable data to Excel. This technique is useful when you want to further explore the data by using a PivotTable in Excel.
- **Create an Excel data model**: The Excel data model (formerly known as *Power Pivot*) is a native feature of Excel. It uses the same database engine as Power BI for storing imported semantic models, and the same Power Query functionality to get data. However, in Excel, the functionality is updated much less frequently than Power BI. It's useful for content creators who create small models and have a strong preference for working in Excel. Optionally, you can import your workbook from SharePoint or OneDrive for work or business. That allows you to view the workbook in the Power BI service. Or you can create a new Power BI semantic model that's synchronized with the data in the workbook (when it's stored in OneDrive for work or school).

There are other ways to work with Excel. These options are less optimal, and so you should use them only when necessary.

- **Export to Excel**: Many users have established a habit of exporting data to Excel from reports or dashboards. While Power BI supports this capability, it should be used cautiously and in moderation because it results in a static set of data. To ensure that data exports to Excel aren't overused, users in the organization should

be educated on the downsides of exports and administrators should track exports in the user activity data.

- **Get source data from Excel**: Excel can be used as a data source when importing data to Power BI. This capability works best for small projects when a user-friendly Excel-based solution is required to maintain source data. It can also be useful to quickly conduct a proof of concept (POC). However, to reduce the risk associated with Excel data sources, the source Excel file should be stored in a secure, shared location. Also, column names shouldn't be changed to ensure data refreshes succeed.

> **Tip**
>
> We recommend that you primarily encourage the use of Excel as a live connection.

Here are some important points to consider when determining whether Excel is an appropriate authoring tool.

- Certain prerequisites must be in place to allow users to connect to a Power BI semantic model from Excel.
- In some organizations, users have the 32-bit version of Excel installed rather than the 64-bit version. The 64-bit version of Excel can support larger data volumes, and generally performs better than the 32-bit version. All data providers must also align with this choice.
- Some features in Power BI Desktop aren't available in the Excel data model, or they're released on a significantly slower cadence. Therefore, complex modeling requirements might not be (easily) possible in Excel.
- Some connectors and data sources that are available in Power BI Desktop aren't available in Excel.

> **Tip**
>
> Many organizations have existing Excel solutions that can be modernized by connecting the Excel file to a Power BI shared semantic model (rather than using a data export). Live connectivity saves users from repeating tedious steps, prevents data from becoming stale, and ensures data security is consistently applied when users refresh the Excel data.

Power BI Report Builder

Power BI Report Builder is a tool for creating a paginated report (.rdl) file. Paginated reports can be deployed to either the Power BI service or Power BI Report Server. If you have experience creating reports in SQL Server Reporting Services (SSRS), you'll find it's a similar report creation experience.

Paginated reports are best suited to highly formatted, or print-ready, reports such as financial statements. They're also suitable for reports that are intended to be printed or for PDF generation, and when user input (with report parameters) is required.

> **Tip**
>
> For other scenarios that favor choosing paginated reports, see **When to use paginated reports in Power BI**.

Here are some important points to consider when deciding on using Power BI Report Builder.

- Approach working in Power BI Report Builder with a different mindset than when you work in Power BI Desktop. A paginated report always focuses on the creation of one individual report (conversely, a semantic model created in Power BI Desktop can serve many different reports).
- Developing paginated reports involves more skill than creating Power BI reports. However, the main benefit is fine-grained control over data retrieval, layout, and placement.
- A paginated report is concerned with both data retrieval and layout. You're required to develop a query (known as a *dataset*) to retrieve data from an external data source, which might involve writing a native query statement (in DAX, T-SQL, or other language). The dataset belongs to one report, so it can't be published and used by other paginated reports.
- Report consumers become accustomed to the built-in interactivity of Power BI reports. However, report interactivity isn't a strength of paginated reports. Attempting to achieve similar interactivity in paginated reports can be challenging or impossible.
- If you need to access data by using a database stored procedure (such as an Azure SQL Database stored procedure), that's possible with paginated reports.
- There are some feature differences and unsupported capabilities depending on whether the paginated report is published to the Power BI service or Power BI Report Server. We recommend that you conduct a proof of concept to determine what's possible for your target environment.

> **Tip**

For more information, see **Paginated reports in Power BI FAQ** and **Design tips for reports in Power BI Report Builder**.

Third-party tools

Advanced content creators can choose to use third-party tools, especially for enterprise-scale operations. They can use third-party tools to develop, publish, manage, and optimize data models. The goal of these tools is to broaden the development and management capabilities available to semantic model creators. Common examples of third-party tools include Tabular Editor, DAX Studio, and ALM Toolkit. For more information, see the advanced data model management usage scenario.

> **Note**
>
> The use of third-party tools has become prevalent in the global Power BI community, especially by advanced content creators, developers, and IT professionals.

There are three main ways to use third-party tools for semantic model development and management.

- **Use an external tool to connect to a local data model in Power BI Desktop**: Some third-party tools can connect to the data model in an open Power BI Desktop file. When registered with Power BI Desktop, these tools are known as external tools and extend the native capabilities of Power BI Desktop.
- **Use the XMLA endpoint to connect to a remote data model in the Power BI service**: Some third-party tools can use the *XML for Analysis (XMLA)* protocol to connect to a semantic model that's been published to the Power BI service. Tools that are compliant with the XMLA protocol use Microsoft client libraries to read and/or write data to a data model by using Tabular Object Model (TOM) operations.
- **Use a template file to connect to a local data model in Power BI Desktop**: Some third-party tools distribute their functionality in a lightweight way by using a Power BI Desktop template (.pbit) file.

Some third-party tools are proprietary and require a paid license (such as Tabular Editor 3). Other community tools are free and open source (such as Tabular Editor 2, DAX Studio, and ALM Toolkit). We recommend that you carefully evaluate the features of each tool, cost, and its support model so you can sufficiently support your content creators.

> **Tip**
>
> Some organizations find it easier to get a new tool approved that's fully supported (even when there's a licensing cost). However, other organizations find it easier to get a free open-source tool approved. Your IT department can provide guidance and help you do the necessary due diligence.

Checklist - When planning for authoring tools, key decisions and actions include:

- ✓ **Decide which authoring tools to encourage**: For self-service creators and advanced content creators, consider which of the available tools will be actively promoted for use in the organization.
- ✓ **Decide which authoring tools will be supported**: For self-service creators and advanced content creators, consider which of the available tools will be supported and by whom.
- ✓ **Evaluate the use of third-party tools**: Consider which third-party tools will be allowed or encouraged for advanced content creators. Investigate the privacy policy, licensing cost, and the support model.
- ✓ **Create guidance for content creators**: Provide guidance and training to help users choose and use the appropriate authoring tool for their circumstances.

Manage and set up devices

This section describes considerations for installing and updating tools and applications and setting up user devices.

Client tools

IT often uses the term *client tools* to refer to software that's installed on client machines (user devices). The most common Power BI software installed on a user device is Power BI Desktop.

Because Microsoft usually updates Power BI Desktop every month, it's important to have a seamless process for managing installations and updates.

Here are several ways that organizations can manage installations and updates of Power BI Desktop.

Expand table

Type of installation	Supports automatic updates	Description
Microsoft Store	Yes	Power BI Desktop is distributed from the Microsoft Store. All updates, including bug fixes, are automatically installed. This option is an easy and seamless approach, provided that your organization doesn't block some (or all) apps from the Microsoft Store for some (or all) users.
Manual installation	No	You can manually download and install an executable (.exe) file from the Microsoft Download Center. However, be aware that the user who installs the software must have local administrator rights—in most organizations, those rights are restricted. If you choose to use this approach (and it isn't managed by IT), there's a risk that users will end up with different versions of Power BI Desktop installed, possibly resulting in compatibility issues. Also, with this approach, every user will need to be notified to install quick fix engineering (QFE) releases, also known as *bug fixes*, when they come out.
IT-managed systems	Depends upon the setup	You can use a variety of IT-managed organizational deployment methods, like Microsoft System Center or Microsoft Application Virtualization (App-V). This option is best suited for organizations that need to manage many installations at scale or in a customized way.

It's important that user devices have adequate system resources. To be productive, content creators who work with large data volumes might need system resources that exceed the minimum requirements—especially memory (RAM) and CPU. IT might have suggested machine specifications based on their experience with other content creators.

All content creators collaborating on Power BI development should use the same version of the software—especially Power BI Desktop, which is usually updated every month. We recommend that you make updates automatically available to users because:

- Multiple content creators who collaborate on a Power BI Desktop file are assured of being on the same version. It's essential that creators who work together on the same .pbix file use the same software version.
- Users won't have to take any specific action to obtain updates.
- Users can take advantage of new capabilities, and their experience is aligned to announcements and documentation. It can impact adoption and user satisfaction when content creators learn about new capabilities and features, yet they experience long delays between software updates.

- Only the latest version of Power BI Desktop is supported by Microsoft. If a user has an issue and files a support ticket, they'll be asked by Microsoft support to upgrade their software to the latest version.

In addition to Power BI Desktop (described previously), you might need to install and manage other Microsoft tools or third-party tools on user devices, including mobile devices. For a list of possible tools, see Available tools for authoring earlier in this article.

Users who create and manage files located in Fabric OneLake might also benefit from OneLake File Explorer. This tool allows them to conveniently upload, download, edit, or delete files in OneLake by using Windows file explorer.

> ⓘ **Note**
>
> Your IT department might have managed device policies in place. These policies could control what software can be installed, and how it's managed.

Client tool prerequisites

Content creators that have client tools installed, such as Power BI Desktop, might require specific prerequisite software or packages.

- **WebView2**: (Mandatory) For content creators running Power BI Desktop, the Microsoft Edge WebView2 Runtime is a prerequisite. WebView2 allows the embedding of web technologies (such as HTML, CSS, and JavaScript) in Power BI Desktop in a secure way. WebView2 will already be installed if the user device has the latest version of Windows or has Microsoft 365 applications installed and monthly updates are enabled.
- **.NET Framework**: (Mandatory) For content creators running Power BI Desktop or a third-party tool, the .NET Framework is a prerequisite. The .NET Framework is a technology that supports building and running Windows apps. Power BI Desktop requires a specific version, or later.
- **Microsoft Edge**: (Mandatory) For content creators running Power BI Desktop, the Edge browser is a prerequisite.
- **Python and R packages**: (Optional) Python and R scripts can be used in multiple ways with Power BI, when enabled by the tenant setting. Scripts can be used to create Python visuals or R visuals. Scripts can also be created in the Query Editor; in this case, a personal gateway is required because Python and R are not supported in the standard data gateway. Python packages or R packages are a prerequisite. To avoid incompatibilities, IT should manage which packages get installed, where

they're installed, and that the versions installed match what's supported in the Power BI service.

Data connectivity components

Depending on your data sources, you might need to install drivers, connectors, or providers on user devices. These components enable data connectivity when a user works in a client tool (such as Power BI Desktop) or a third-party tool.

- **Drivers**: A *driver* is a software component that connects to other systems. For example, to connect to an Oracle database, you might need the Oracle Data Access Client software. Or, to connect to SAP HANA, you might need an ODBC driver.
- **Custom connectors**: A custom data source connector might be required when connecting to a legacy or proprietary system.
- **Excel provider**: The Analyze in Excel provider allows users to create visualizations in Excel while connected to an existing shared semantic model that's been published to the Power BI service.
- **Analysis Services client libraries**: When connecting to an Analysis Services source, a client library needs to be installed.
- **Access Database OLE DB provider**: When connecting to an Access database, an OLE DB provider needs to be installed.

> **Important**
>
> For data sources that require connectivity through a gateway, the same drivers, connectors, and providers will need to be installed on each **data gateway** machine. Missing components on a data gateway are a common reason for data refresh failures once content has been published to the Power BI service.

> **Tip**
>
> To simplify delivery to a larger number of users, many IT teams deploy the most common drivers, connectors, and providers as part of a standard user device setup.

Version control tools

Content creators that have client tools installed, such as Power BI Desktop, should also have a way to save *versions*, or historical copies, of files. Access to previous versions is particularly helpful when a change needs to be rolled back.

There are two main ways to handle versioning of development files.

- **Teams, OneDrive for Business, SharePoint**: Self-service content creators often save files in Teams, OneDrive for work or school, or SharePoint. Users find these tools are familiar and simple to use. Shared libraries can be organized, secured for appropriate coworkers, and versioning is built in.
- **Source control plug-ins**: Advanced content creators might need to integrate with a source control tool. For example, this typically involves installing Git for source control, then using a source control management tool like Visual Studio Code to commit content changes to a remote repository, such as Azure DevOps Repos. For Power BI Desktop, developer mode can be used. In this mode, content is saved as a Power BI project (.pbip) file, which is compatible for use with your preferred source control system. When working with Fabric, Git integration is supported for working with a client tool.

For more information, see Strategy for file locations.

Custom visuals

Power BI custom visuals, which developers can create by using the Power BI visuals SDK, allow Power BI report creators to work beyond the built-in core visuals. A custom visual can be created and released by Microsoft, software developers, vendors, or partners.

To use a custom visual in Power BI Desktop, it must first be installed on the machine of the content creator. There are several ways to distribute visuals to users.

- **AppSource**: AppSource is a marketplace for applications, add-ins, and extensions for Microsoft software. Visuals are distributed in AppSource using a Power BI visual (.pbiviz) file. A visual might be distributed freely or requires a license.
  - **Advantages**:
    - It's simple for users to search for, and locate, visuals in AppSource.
    - All reports and dashboards are automatically updated to use the latest version of custom visuals that have been sourced from AppSource.
    - Supports the use of certified visuals.
    - Microsoft performs basic validations of visuals published to AppSource. The extent of the review depends on whether the visual is certified or not.
  - **Potential disadvantages**:
    - When each content creator downloads what they need from AppSource, it can lead to incompatibilities when users have different versions installed.

    - A content creator might download a visual that hasn't yet been tested or approved for use in the organization.
    - The developer of the visual needs to follow a strict publishing process. Although it strengthens security and improves stability, the process can make it challenging to release a bug fix quickly,
- **Import a visual file**: A content creator can import a visual file into Power BI Desktop.
    - **Advantages**:
        - Visuals that are available publicly, or privately distributed, can be installed. That includes internally developed visuals or proprietary visuals purchased from a vendor.
        - Allows a way to obtain a visual file outside of AppSource.
    - **Potential disadvantages**:
        - Without a centralized system, it can be difficult for content creators to know what visuals have been approved for use in the organization.
        - When each content creator imports the visual file they have, it can lead to incompatibilities when users have different versions installed.
        - Updates aren't automatically propagated to user devices. Reports in local Power BI Desktop files aren't updated until each user updates their visual files.
        - Doesn't support the use of certified visuals.
- **Organizational visuals**: The organizational visuals repository is a centralized area in the Fabric admin portal for managing visuals.
    - **Advantages**:
        - Content creators don't have to manage visual files. Instead, a Fabric administrator centrally manages the version of a visual that's available for all users. Version consistency is ensured for all users and all reports.
        - Visuals that are available publicly or privately distributed can be installed. That includes internally developed visuals or proprietary visuals purchased from a vendor.
        - Visuals can be tested and pre-approved for use in the organization. This verification process reduces the risk that non-approved visuals are used. It also allows greater flexibility for setting which specific version of a visual is approved for use.
        - All reports and dashboards are automatically updated to use the latest version (when a visual file is updated in the admin portal or made available in AppSource).
        - If a visual that's currently in use by the organization is deemed to be no longer trustworthy, it can be disabled or deleted from the organizational visuals repository. In this case, the visual won't be rendered in reports and dashboards.

    - Allows the use of non-certified visuals from AppSource. That's useful when you've set the tenant setting to block uncertified visuals, yet a specific non-certified visual has been validated and approved for use in the organization.
- **Potential disadvantages**:
    - Organizational visuals need to be managed centrally by a Fabric administrator.
    - Centralization correlates to reduced user flexibility and the potential for delays in updating the version of a visual.
    - Some features aren't available when a visual isn't certified (which requires importing from AppSource).

> **Important**
>
> If your organization is highly concerned about data privacy and data leakage, consider governing all custom visuals through the organizational visuals repository.

> **Tip**
>
> How you distribute custom visuals is a **governance** consideration. We recommend that you carefully evaluate the features of each visual, considering its cost and support model so you can sufficiently support your content creators.
>
> Also, before you approve the use of a new custom visual, it's critical that you evaluate any security and data privacy risks because:
>
> - Visuals execute JavaScript code and have access to the data that they visualize.
> - Visuals can transmit data to an external service. For example, a visual might need to transmit data to an API to run an AI algorithm or to render a map. Just because a visual transmits data to an external service, it doesn't mean it's untrustworthy. A visual that transmits data can't be certified.

For more information, see Govern organizational visuals.

Group policy settings

Group policy provides centralized management and configuration of operating systems, applications, and user settings of Windows machines and the network environment. It helps IT roll out and manage consistent user accounts and machine settings. For Power

BI Desktop, the most common use of group policy is to manage custom visuals (described in the previous section).

You can specify whether uncertified visuals are allowed or blocked in Power BI Desktop. To ensure users have a consistent experience in both Power BI Desktop and the Power BI service, it's important that custom visuals be managed consistently in two places.

- **Tenant setting**: The *Add and use certified visuals only (block uncertified)* tenant setting allows or blocks using custom visuals when users create or edit reports in the Power BI service.

- **Group policy**: The group policy setting controls the use of custom visuals when users create or edit reports in Power BI Desktop. If a content creator spent considerable time creating content in Power BI Desktop that can't be displayed in the Power BI service (due to a misaligned tenant setting), it would result in a significant amount of user frustration. That's why it's important to keep them both aligned.

You can also use group policy to specify whether data exports are allowed or blocked from custom visuals.

Registry settings

The Windows operating system stores machine information, settings, and options in the Windows registry. For Power BI Desktop, registry settings can be set to customize user machines. Registry settings can be updated by group policy, which helps IT set up default settings that are consistent for all users (or groups of users).

Here are several common uses of registry settings related to Power BI Desktop.

- Disable notifications that a software update is available. That's useful when you're certain that IT will obtain the Power BI Desktop update, perform validations, and then push updates to user devices through their normal process.
- Set the global privacy level. It's wise to set this setting to *Organizational* as the default because it can help to avoid data privacy violations when different data sources are merged.
- Disable the Power BI Desktop sign-in form. Disabling the form is useful when organizational machines are automatically signed in. In this case, the user doesn't ever need to be prompted.
- Tune Query Editor performance. This setting is useful when you need to influence query execution behavior by changing defaults.
- Disable the external tools ribbon tab. You might disable the ribbon tab when you know you can't approve or support the use of external tools.

> **Tip**
>
> Usually, the goal isn't to significantly limit what users can do with tools. Rather, it's about improving the user experience and reducing support needs.

Mobile device management

Many users like to interact with Power BI content on a mobile device, such as a tablet or a phone, whether they're at home or traveling. The Power BI mobile apps for iOS, Android, and Windows are primarily designed for smaller form factors and touch screens. They make it easier to interact with content that's been published to the Power BI service or Power BI Report Server.

You can specify app protection policies and device protection policies for managed and unmanaged devices by using Microsoft Intune. Intune is a software service that provides mobile device and application management, and it supports mobile application management (MAM) policies. Policies can be set at various levels of protection.

Optionally, a mobile device management (MDM) solution from Microsoft 365, or a third party, might also be used to customize the behavior of Power BI mobile apps. The Power BI app for Windows also supports Windows Information Protection (WIP).

Here are several ways that you might choose to use MAM and MDM policies.

- Specify data protection settings.
- Encrypt application data when the app isn't in use.
- Selectively wipe data when a device if it gets lost.
- Prevent saving of data to a personal storage location.
- Restrict actions to cut, copy, and paste.
- Prevent printing of organizational data.
- Require biometric data, or an access PIN, to open the mobile app.
- Specify the default behavior when a user selects or taps in a mobile app.

For more information about securing devices and data, see the Power BI security whitepaper.

Checklist - When managing devices, key decisions and actions include:

- ✓ **Determine how Power BI Desktop will be updated**: Consider how to install Power BI Desktop (and other client tools). Whenever possible, ensure that updates are automatically installed.
- ✓ **Identify the necessary client tool prerequisites**: Ensure that all prerequisite software and packages are installed and updated regularly.
- ✓ **Identify the necessary data connectivity components**: Ensure that all drivers, connectors, and providers that are required for data connectivity are installed and updated regularly.
- ✓ **Determine how to handle custom visuals**: Decide how custom visuals will be handled from AppSource and other sources. Set the *Allow visuals created from the Power BI SDK* tenant setting and the *Add and use certified visuals only* tenant setting to align with your decisions. Consider creating a process that allows users to request a new custom visual.
- ✓ **Set up group policy settings**: Set up group policy to ensure that custom visuals are managed the same way in Power BI Desktop as they are in the Power BI service.
- ✓ **Set up registry settings**: Set up the registry settings to customize user machines, when applicable.
- ✓ **Investigate mobile device management**: Consider using app protection policies and device protection policies for mobile devices, when appropriate.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Subscriptions, licenses, and trials

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces the key considerations for planning subscriptions, licenses, and trials for Power BI and Fabric. This article is targeted at:

- **Billing administrators**: The administrators who are responsible for purchasing subscriptions and analyzing costs.
- **Azure administrators**: The administrators who are responsible for purchasing and managing Azure subscriptions and services.
- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization.
- **License and user management administrators**: The administrators who are responsible for assigning licenses (that have been purchased) to users.
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI and supporting Power BI users in the organization. These teams make key decisions and collaborate with the Fabric administrators.
- **Content owners and creators**: This article might also be relevant for self-service content creators who need to obtain licenses so that they can create, publish, and manage content.

One key aspect of managing Microsoft Fabric is to ensure that users have access to the capabilities that they need. To this end, you must purchase and manage subscriptions, licenses, and trials for your organization. Managing subscriptions, licenses, and trials is necessary to ensure that both content creators and content consumers can use Fabric and Power BI.

> **Note**
>
> Licensing is an important topic that can be complex, especially when your organization is implementing Fabric or Power BI for the first time. While this article describes key decisions and considerations about subscriptions, licenses, and trails,

we recommend that you refer to the following supplementary articles and resources for more detailed and practical information.

- Power BI pricing: This webpage provides the latest information about the pricing of the different licenses available for Power BI and Fabric in your region, as well as a feature comparison.
- Power BI service per-user and capacity-based licenses: This article provides detailed information about the different licenses available to use Power BI.
- Licensing the Power BI service for users in your organization: This article (and related articles) provides practical information about how to purchase and assign licenses for Power BI.
- Microsoft Fabric concepts and licenses: This article provides detailed information about the different capacity licenses available to use Fabric, including information about the different stock-keeping units (SKUs) for each license. It also describes the difference between a Premium capacity and a Fabric capacity, both with respect to the SKUs and capabilities.
- Buy a Microsoft Fabric subscription: This article provides practical information about where and how you can purchase Fabric capacity licenses for your organization. It also describes the difference between pay-as-you-go subscriptions (Azure SKUs) and reserved instances with monthly or yearly billing (Microsoft 365 SKUs).

Important

At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.

For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

The following table introduces key concepts used by this article.

Expand table

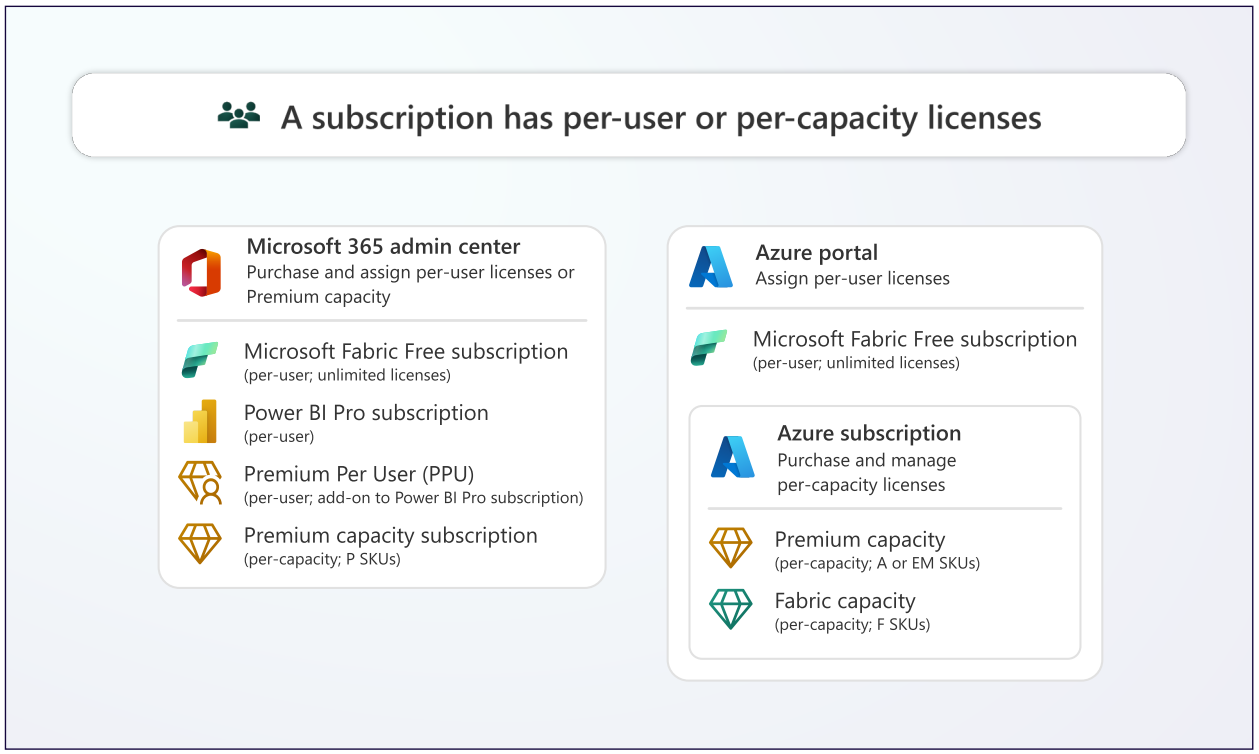
Concept	Description	Example
Tenant	Fabric operates within the organization's Microsoft Entra tenant. An organization will typically have a single *tenant* (though some large organizations might have multiple tenants).	At Contoso, one tenant exists for *contoso.com*.
Subscription	One or more *subscriptions* must be active for a tenant. Each subscription has a start date and an end date that references a product: • Per-user subscriptions are managed in the Microsoft 365 admin center. A specific number of licenses are available within each subscription. • A Power BI Premium capacity subscription is managed in the Microsoft 365 admin center. • A Fabric capacity subscription is managed in the Azure portal.	Contoso has four active subscriptions: • Unlimited Free licenses. • 100 Pro licenses. • 15 PPU licenses. • Three capacity licenses.
Per-user license	A *per-user license* is based on a subscription. All users require a user license, which can be Fabric Free, Power BI Pro (Pro), or Premium Per User (PPU).	Contoso has 450 active users: • All 450 users are assigned a Fabric Free license. • 92 users are assigned a Power BI Pro license (leaving eight available on the subscription). • 15 users are assigned a PPU license (leaving none available on the subscription).
Capacity license	A *capacity license* is based on a subscription. Workspaces are assigned to a capacity, which determines the capabilities and resources that are available to the content and users of the workspace.	Contoso has three active capacities: • Two Fabric capacities. • One Power BI Premium capacity.
SKU (stock-keeping unit)	*SKUs* are a product ID for the subscription that was purchased. For capacity, there are two ways to refer to SKUs. • **Grouping**: The type of capacity that's purchased. For example, an *F SKU* refers to a Fabric capacity in general. • **Specific**: The specific SKU for a capacity designates the level of its compute power. For	Contoso has three active capacities: • One F16 capacity. • One F64 capacity. • One P1 capacity.

Concept	Description	Example
	example, an *F64 capacity* has a specific set of compute resources (such as CPU and memory) that are available to all workspaces that are assigned to that capacity.	
Trial	A trial license allows you to try out features. A trial can be activated for either a per-user license or a Fabric capacity license.	Contoso has active trials: • Two users have an active PPU trial. • One Fabric capacity trial is active.

> **Note**
>
> The *subscriptions* referred to in this article relate to costs for a product. It's a different concept than ***report subscriptions***, which are reports delivered on a schedule.

> **Important**
>
> Speak with your Microsoft account manager when you have any questions about your organization's subscriptions and licenses for Fabric, Power BI, or Microsoft 365.

Licensing options can be mixed and matched based on the workload for individual users and groups. The following diagram depicts how a subscription has per-user or per-capacity licenses, which you manage either from the Microsoft 365 admin center or the Azure portal.



In the Microsoft 365 admin center, administrators can purchase and assign per-user licenses or Premium capacity.

- The organization has a Microsoft Fabric Free subscription with unlimited Free licenses. Billing administrators can assign these licenses to users in the organization.
- Billing administrators can purchase Power BI Pro subscriptions and assign Power BI Pro licenses to users in the organization.
- Billing administrators can purchase Power BI Premium Per User (PPU) licenses as add-ons to an existing Power BI Pro subscription. These administrators can then assign PPU licenses to users in the organization.
- Billing administrators can purchase Power BI Premium capacity subscriptions (P SKUs) for the organization. These licenses are per-capacity and not per-user.

In the Azure portal, Azure administrators can assign per-user licenses and purchase and manage per-capacity licenses.

- Azure administrators assign per-user licenses, like Microsoft Fabric Free, in the Azure portal in Microsoft Entra ID.
- Azure administrators manage their Azure subscription, where they can purchase and manage per-capacity licenses.
  - Billing administrators can purchase Premium capacity subscriptions (A SKUs or EM SKUs), which are linked to, and billed as part of, an Azure subscription.
  - Billing administrators can purchase Fabric capacity subscriptions (F SKUs), which are linked to, and billed as part of, an Azure subscription.

> **Note**
>
> Many of the capabilities described in this article aren't available to Fabric administrators. Instead, Fabric administrators must **collaborate with other administrators** who have additional permissions to view (or update) subscriptions, billing, and licenses.

The remainder of this article describes licensing considerations for both per-user licensing and capacity licensing.

Review and manage per-user licensing

Every user who works with Fabric requires a user license (either Free, Pro, or PPU, which will be described in Step 2). Even if you intend to purchase capacity licenses (covered later in this article), per-user licenses are required so that each user can access Fabric. This access is facilitated by integration with Microsoft Entra ID.

Step 1: Review user licenses

It's important that you first understand the current state of user subscriptions and licenses. Your billing administrators can help by confirming any user subscriptions you currently have, and how user licenses are assigned.

Here are two common ways that you can compile a list of user subscriptions and licenses.

- View the Billing area of the Microsoft 365 admin center.
- Programmatically extract the data by using the relevant Microsoft Graph REST APIs.

> **Tip**
>
> You can also use this information as part of your **tenant-level auditing**. Specifically, see this section about retrieving data about **users and groups**.

As you conduct your review, compile the following information.

- Active subscriptions for user licenses:
  - Microsoft Fabric Free
  - Power BI Pro

    - Power BI Premium Per User
- Subscription status
- Subscription start and end dates
- Subscription quantities:
    - Total purchased
    - Number of licenses assigned to users
    - Number of available licenses
- Subscription cost:
    - Pricing for each subscription
    - Organizational pricing information (if applicable)
    - Who approved the purchase (if applicable)
- User trials that are currently active

> **ⓘ Important**
>
> Make sure that you include existing self-service subscriptions in your list so that you get the full picture. For more information, see Manage self-service purchases.

Step 2: Decide on user licenses

After you've reviewed user licenses, you should next make some key decisions about how to assign and manage these licenses in your organization.

> **ⓘ Important**
>
> Your decisions for per-user licenses, with or without capacity (described in the next section of this article), will significantly impact what capabilities are available for authors and consumers.

Decide which per-user licenses you need

You should determine which user licenses are required. Every user needs to be assigned a Microsoft Fabric (Free) license *or* a Power BI Pro license. If they're creating or viewing content that's published to a workspace that uses *Premium per user* license mode, then they also need a Power BI Premium Per User (PPU) license.

Here's each of the user license types and their uses.

- **Microsoft Fabric (Free) license**: A Free license that has no subscription cost. It can be used in different ways:

    - **Personal BI**: A Free user can use their personal workspace in the Fabric portal. Because the purpose is personal BI, no report distribution, sharing, or collaboration features are available to the Free user.
    - **Consuming BI**: A Free user can view content that's been deployed to a workspace that's assigned to a capacity (minimum of F64 or P1). This enterprise BI use case is significant when you have a large number of report consumers who don't author BI content, because you don't have to purchase a Power BI Pro license for these consumers. For more information, see Report consumer security planning.
    - **Fabric authoring**: A Free user can create and share Fabric (non-Power BI) items in a workspace that's assigned to an F SKU.
- **Power BI Pro license**: A Power BI Pro license is required to author Power BI content. It's required for any form of sharing, collaboration, or content distribution. For more information, see Content creator security planning.
- **Power BI Premium Per User (PPU) license**: A PPU license provides all Pro license capabilities and includes certain Premium features, on a per-user basis. It's a good choice for smaller organizations and teams that want to use specific features but don't need the full set of Fabric capabilities. For more information, see Power BI Premium Per User.

> **Tip**
>
> You can mix and match user licenses with capacity licenses. For example, you might have development, test, and production workspaces that rely on specific **enterprise content publishing** approaches. Since the development and test workspaces have very few users, those workspaces might be assigned a smaller capacity size or PPU license mode (if they don't require a Fabric experience or capabilities). The production workspace could use a capacity license to support many consumers (with Free licenses). That way, you might potentially reduce cost, while segregating the development and test content from the production workload.

Decide prerequisites for obtaining a user license

Consider whether there are any requirements that must be met before a user license is assigned.

Here are some examples of prerequisites.

- Requiring the user to acknowledge an organizational data policy (such as a data privacy or a data handling policy) prior to being assigned a user license.

- Requiring the user to complete an initial training session or certification before a license is granted.
- Implementing a workflow, such as:
  - Require manager approval.
  - Confirm which subscription to assign the license from.
  - Approve where to charge the cost.
  - Confirm that the user's role and responsibilities justify a license (and access to data).

> **Tip**
>
> Don't introduce too many obstacles that could prevent users from getting a license. If it's too difficult, busy professionals might not bother to request a license. Instead, to get work done, they'll find another way, which might involve suboptimal workarounds. For instance, without a license, people might share files on a file system or via email when better—and more secure—approaches are available.

Decide the process for handling user license requests

Determine whether you need to implement a custom license request process that's defined by your organization. This request could handle scenarios such as:

- Requiring prerequisites (described in the previous section).
- Integrating with an existing license management platform.
- Informing users of training offerings and available help as part of the user license assignment process.
- Determining how costs will be allocated, or how chargebacks will be done.

> **Tip**
>
> You can set a custom URL for licensing requests in the ***Publish Get Help Information*** tenant setting. The URL can direct users to a form, or to your internal licensing request page.

Decide how user subscriptions will be purchased

It's important to plan exactly how your process to purchase subscriptions will work.

Here are some questions you should consider.

- Are subscriptions purchased individually or in bulk?
  - In smaller organizations, you might choose to purchase each subscription (and assign the license) upon request. That approach works well when there's a low volume of requests.
  - In larger organizations, it's often efficient to purchase subscriptions in batches (such as 50 Pro licenses, for example). This approach works well when sufficient funds are available in the budget, and you expect the licenses to be assigned (and used) soon.
- Do you have an Enterprise Agreement (EA)? For example, your organization procured a Microsoft 365 subscription that includes 500 enterprise E5 licenses. In that case, each user that's assigned an E5 license will have a Power BI Pro license (note that access to individual apps can be removed for users when necessary).
- Is purchasing a centralized function that's handled by one department? Or is self-service purchasing permitted?

> **Tip**
>
> We recommend that you prioritize **mentoring and user enablement** and **user support** activities. These activities become even more important when licenses are widely distributed to users throughout your organization.

Decide whether trials are enabled

An important governance decision is whether user trials are allowed. An in-product trial provides users with a way to try out features before making a commitment to buy a license. There are two types of trials that a user could initiate: a Premium Per User (PPU) trial and a Fabric capacity trial.

When trials are enabled, a *Start trial* button is displayed in the portal that permits a user to begin a Fabric trial. Also, a user can be prompted to start a trial while they're working. For example, if a Free user tries to create a workspace or share a report, they'll be prompted to start a PPU trial by selecting the *Try free* button. Similarly, if a Pro user tries to view content in a PPU workspace, they'll be prompted to start a trial by selecting the *Try free* button.

The ability to use in-product trials is controlled by the *Users can try Microsoft Fabric paid features* tenant setting. Its behavior is closely correlated with how self-service purchasing works (described next). For more information, see Users can try Microsoft Fabric paid features.

> **Note**
>
> The trial experience is intended to be a convenience that allows users to continue with their normal workflow. Generally, disabling trials isn't recommended. Restricting trials might encourage users to seek workarounds, perhaps by exporting data or working outside of supported tools and processes.

Consider disabling trials only when:

- There are significant cost concerns that would make it unlikely to grant the user a full license at the end of the trial period.
- Prerequisites are required for obtaining a license (such as approval, justification, or a training requirement) that must be met prior to beginning a trial or obtaining a license.
- There's a valid need, such as a regulatory requirement, to control access to Fabric closely.

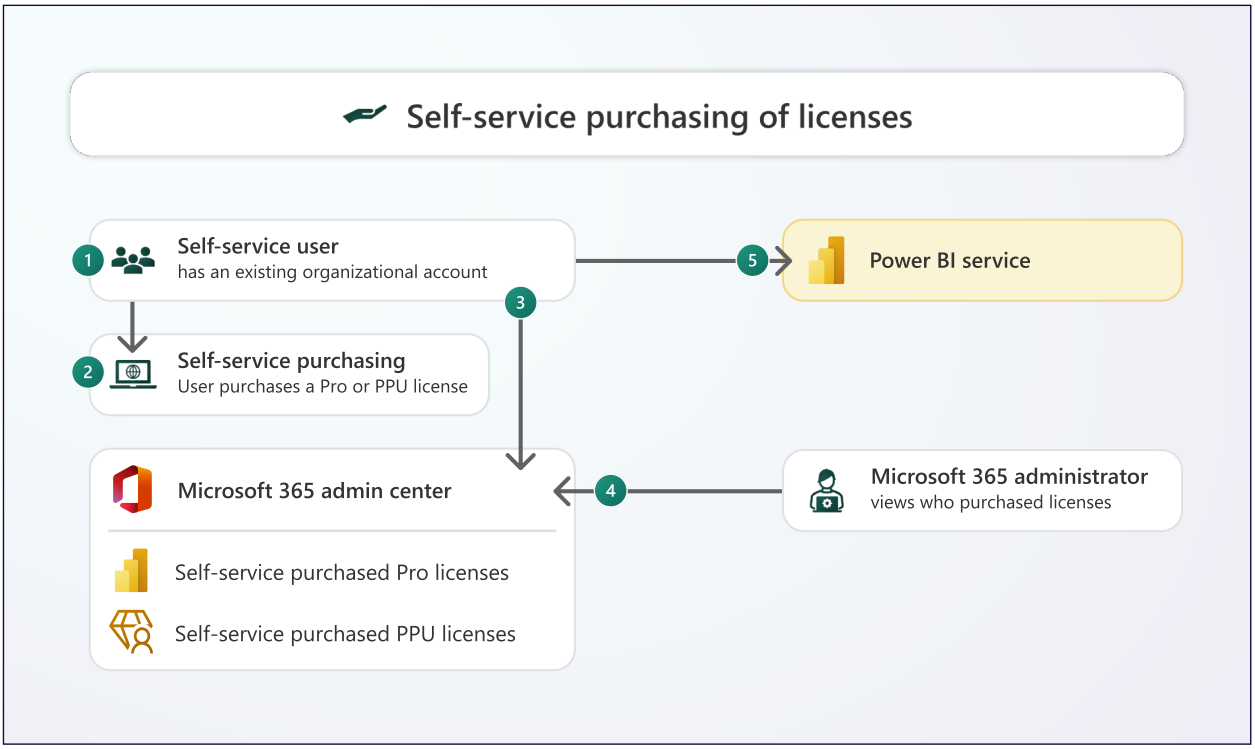
Decide whether self-service purchasing is enabled

Users can purchase licenses on their own when *self-service purchasing* is enabled. In this case, a user can purchase a license while they're working. For example, if a Pro user tries to view content in a PPU workspace, they could choose to purchase a license by selecting the *Upgrade account* or the *Buy now* button. For more information, see Sign up for or purchase the Power BI service as an individual.

Self-service purchasing is useful for:

- Larger organizations with decentralized business units that have purchasing authority and want to handle payment directly with a credit card.
- Organizations that intend to make it as easy as possible to purchase subscriptions on a monthly commitment.

The following diagram depicts how self-service purchasing works (when there isn't a custom URL for license requests).



The diagram depicts the following processes and steps in the self-service purchasing of licenses.

Expand table

Item	Description
1	A self-service user can purchase their own license only when they have an existing organizational account.
2	The user can purchase their own license when self-service publishing is supported by the organization.
3	The user has a limited view of the licenses purchased from the Microsoft 365 admin center that they can assign to others within their scope of responsibility.
4	Microsoft 365 administrators have a holistic view of all the licenses acquired through self-service publishing in the Microsoft 365 admin center.
5	Users with a self-service purchased license can access and use Power BI.

Consider disabling self-service purchasing when:

- Centralized procurement processes are in place to meet regulatory, security, and governance requirements.
- Discounted pricing is obtained through an Enterprise Agreement (EA).
- Existing processes are in place to handle intercompany chargebacks.
- Existing processes are in place to handle group-based licensing assignments.

- Prerequisites are required for obtaining a license, such as approval, justification, training, or a governance policy requirement.

There are many considerations when planning for self-service purchasing.

Here are some other factors to consider.

- Is there an existing organization-wide policy in place for self-service purchases and trials?
- Does Fabric follow the existing organizational policy for self-service purchases and trials?
- Should purchasing and trials both be enabled, both be disabled, or a combination? The user's experience will depend on how you combine the settings for purchasing and trials.
- Should users who try to purchase a license be directed to a specific page?
  - With a custom URL for licensing requests, the user is immediately directed to that page when they select the *Upgrade account* or the *Buy now* button. You can provide further instructions or ask them to submit details in a form. Using a custom URL is a way to disallow the use of self-service purchasing by redirecting the user elsewhere.
  - Without a custom URL, a user choosing to purchase a license is directed to Microsoft 365 to complete their purchase.

Decide how to handle licensing for external users

You might need to work with guest users who are external to your organization. Guests could include customers, partners, or vendors. They could also be consultants or external collaborators. This topic could also apply to organizations that have multiple legal entities or tenants due to mergers and acquisitions.

Here are some considerations when you're planning how to handle licensing for guest users.

- How will your process for assigning user licenses be different when an external user is involved?
- Does the external user work for an organization that has Microsoft Entra ID set up? In that case, their credentials can be managed by their home tenant. For more information, see Strategy for external users.
- Which external users are consumers only versus those who need to create and publish content?
- In what situations will a license be provided by the external user (known as bring-your-own-license, or BYOL). In what circumstances will a license be provided by

your organization? For more information, see the licensing topic in Distribute Power BI content to external guest users using Microsoft Entra B2B.

- Which type of guest invitation process will you use? There are different capabilities for occasional versus planned guest user invitations. The user experience is different as well. For more information, see Guest invitation process.

> **Tip**
>
> For more information, see the **Microsoft Entra B2B white paper**. It's a good resource to learn about strategies for handling external users.

Step 3: Update user licenses

At this point, information is available on your existing subscriptions and licenses, and you've made purposeful decisions. Now you're ready to make any updates that are necessary.

> **Important**
>
> Be sure to coordinate changes with your Microsoft account manager should you have any questions or need clarification.

The following topics are actions that might be appropriate.

Increase or decrease user subscription quantity

Based on the information you've gathered, you might choose to adjust your existing user subscriptions. For example, you might choose to increase or decrease the quantity of your Pro or PPU subscriptions.

> **Note**
>
> Adjustments to your user subscriptions might correlate with other changes to a capacity subscription. Capacity licensing is covered later in this article.

Assign or unassign user licenses

You might need to assign or unassign user licenses. For example, you might identify that you need to assign more Pro licenses to certain users, or unassign PPU licenses from

other users.

Take over self-service purchases

If your objective is to centrally manage all subscriptions, you might need to take over a purchase previously made by a self-service user. For more information, see Decide whether self-service purchasing is enabled.

Adjust tenant settings

Based on your decisions for handling user licensing and trials, you might need to adjust some tenant settings in the Fabric admin portal.

You might need to update the:

- *User can try Microsoft Fabric paid features* tenant setting depending on how you want to handle in-product trials.
- *Publish Get Help information* tenant setting if you intend to direct users to a custom URL for licensing requests.

Step 4: Document user licenses

Depending on your internal processes, you might choose to create documentation that augments the information that's available in the portal for your user subscriptions and licenses.

You can build on the information captured in Step 1 by including the following details in your documentation.

- Key decisions, including more context or detail
- Who approved user license purchases, and when
- Timing or pending action items
- Governance requirements related to user licenses
- Auditing requirements related to user licenses
- A snapshot of user license information

> **Tip**
>
> Unless you're a very small organization with very few changes, don't manually document every user license. Instead, use the **Microsoft Graph APIs** to extract information about subscriptions and licenses regularly. Consider storing a snapshot of the user license data every week or every month. That way, you can compare

snapshots to determine what's changed. For more information, see **Audit user licenses**.

> **Important**
>
> Refer to the **Product plans and service plan identifiers** documentation when comparing the Microsoft Graph results to what's displayed in the Microsoft 365 admin center.

Step 5: Manage user licenses

User licenses will need attention on an ongoing basis. The following topics are aspects to consider.

Create a process to accept user license requests

You should create a repeatable, documented process for requesting a user license. It typically involves creating an online form. Information about required prerequisites should be included as well.

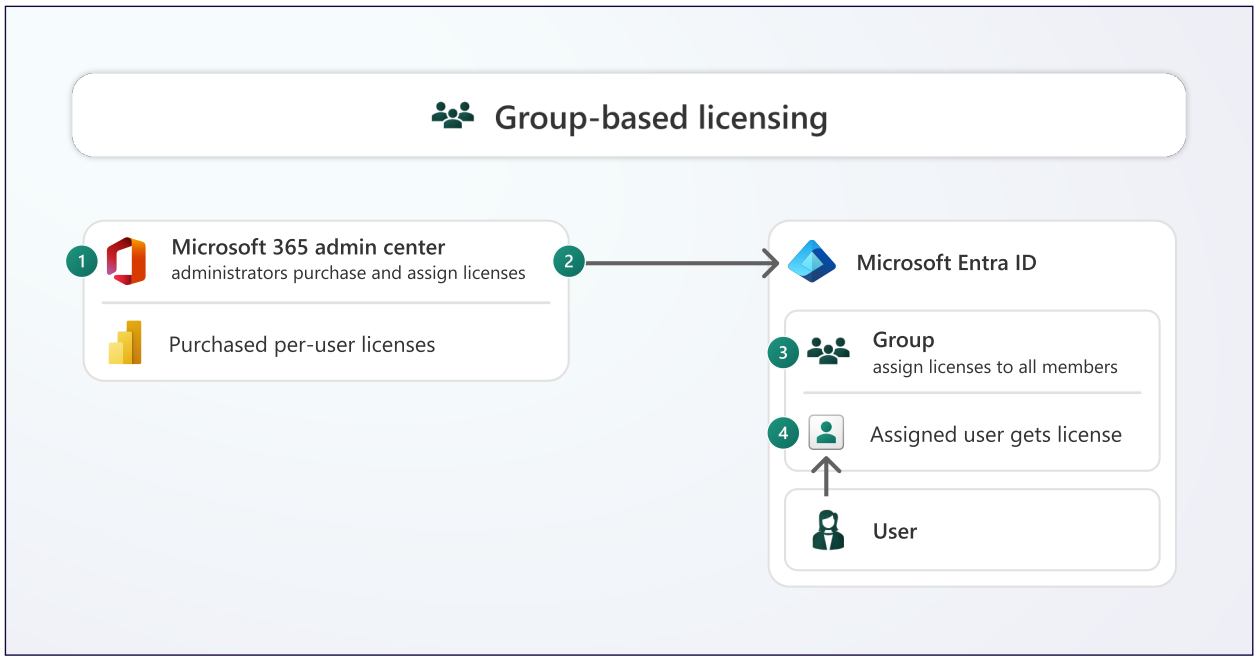
Monitor user trials

Every month you should identify users who have started a trial that will soon expire. It's possible that the user will need to have a license assigned. The goal is to avoid a service interruption for these users. For more information, see User trial auditing.

Automate user license assignments

In large organizations, handling requests for user licenses can involve significant administrative effort. One way to improve efficiency is to use group-based licensing. Group-based licensing allows you to automatically assign a license based on membership of a security group. A group such as *Fabric content authors* works well for this purpose, allowing the assignment of licenses to users in an efficient manner.

The following diagram depicts how group-based licensing works.



The diagram depicts the following processes and steps involved in group-based licensing.

Expand table

Item	Description
1	Billing administrators purchase and assign per-user licenses from the Microsoft 365 admin center.
2	Administrators assign these licenses to groups that they manage in Microsoft Entra ID.
3	The groups are set up to assign licenses to all members.
4	Any user added to a group is automatically granted the Power BI license that's assigned to that group (providing one is available).

Review and optimize user licensing costs

Periodically compare licensed users with the activity log to determine whether users are actively using their license. Look for users who are assigned a license but haven't used it. For example, a user might have a Pro license assigned but they only view content that exists in a capacity. Use consistent criteria whenever possible, such as:

- The license isn't used for a specific period (for example, six months).
- The license is used infrequently or sporadically.
- The license was used once for a single activity.

The activity log will help you identify when activities occurred for a user, and what they are (for example, viewing a report or publishing a report).

Step 6: Audit user licenses

It's important to have a process to regularly audit subscriptions, licenses, and trials for users. A Fabric administrator will need to work together with other administrators to obtain this information (such as a global administrators, billing administrators, and Azure administrators).

User subscription auditing

Here are some actions to look for when you audit subscriptions.

- **List of active subscriptions**: You can view products that have an active subscription in the billing area of the Microsoft 365 admin center. Or, with Microsoft Graph, use the List Subscribed SKUs REST API to extract active subscriptions.
- **New subscription was created**: Recently purchased products display a *New* indicator in the billing area of the Microsoft 365 admin center.

User license auditing

Here are some actions to look for when to audit user licenses.

- **List of user licenses**: View the total number of licenses that are available and assigned in the billing area of the Microsoft 365 admin center. You can also drill down to review users who are assigned a license for each product subscription. Or, with Microsoft Graph, use the List License Details REST API to extract details of each user (to provide the ID parameter, first obtain each user ID from the List Users REST API).
- **User was assigned a license**: Search the audit log in the Microsoft Purview compliance portal. Look for the *Change user license* operation. The *ModifiedProperties* property indicates that a new license was assigned.
- **Users who are assigned a license but haven't used it**: Use the list of user licenses (described previously). Compare those results to the activity log. Look for users that have no activity in the activity log. To make this easier, you can also view activities from the Feature usage and adoption report and its underlying semantic model from the Admin monitoring workspace.
- **Subscription has available licenses not yet assigned**: View the number of available licenses per product subscription in the billing area of the Microsoft 365 admin center. Or, with Microsoft Graph, use the List Subscribed SKUs REST API to extract

license details. The *ConsumedUnits* property indicates how many licenses are assigned, and the *Enabled* property indicates the number of licenses purchased.

User trial auditing

When searching for new user trials in the activity log, look for the *OptInForPPUTrial* or *OptInForProTrial* activities.

> **Important**
>
> The information presented in this step isn't intended to be an all-inclusive list of ways to audit the data. Rather, it's intended to provide you with ideas to get you started with your auditing efforts. For other ideas, we recommend that you consult with your billing administrators.

Checklist - When planning for user licenses, key decisions and actions include:

- **Conduct a review**: Determine the current state by reviewing current user subscriptions, licenses, and trials.
- **Identify existing policies and decisions**: Compile existing internal policies or previous decisions related to user licensing so that they're readily available.
- **Discuss and decide**: Schedule workshops to make key decisions related to user subscriptions, licenses, and trials. Involve all relevant decision makers, stakeholders, and the executive sponsor, when appropriate.
- **Create documentation**: Compile the information gathered, and document key decisions for future reference.
- **Make updates**: Update user subscriptions and licenses based on the decisions made in the workshops.
- **Create a process to manage user requests**: Set up a process for how users can request a license.
- **Set up auditing**: Create auditing processes so you can track activities related to user subscriptions, licenses, and trials.

Review and manage capacity licensing

In addition to user licenses (described earlier in this article), your organization might find significant value in purchasing one or more capacity licenses. For instance, capacity licenses provide access to other features, such as the Fabric experiences that come with a Fabric license. These features can help you to support and scale your Power BI implementation.

Step 1: Review capacity subscriptions

It's important that you first understand the current state of your capacity subscriptions. Your billing and license administrators can help confirm what capacity subscriptions you have. You might also need to speak with current capacity administrators or capacity contributors to gain a clear understanding of the purpose of each existing capacity. For more information, see Manage user permissions.

You can compile the current state of your capacity subscriptions and licenses in different ways.

- View capacity settings in the Fabric admin portal.
- View the billing area of the Microsoft 365 admin center (applicable to Power BI Premium).
- View the Azure portal (applicable to Fabric capacity and Power BI Embedded).
- Programmatically extract the data by using the relevant Microsoft Graph APIs.

As you conduct your review, compile the following information.

- Active subscriptions for capacity licenses:
  - Microsoft Fabric capacity
  - Power BI Premium capacity
  - Power BI Embedded
- Subscription status
- Subscription type (pay-as-you-go or reserved with a start and end date)
- Subscription cost:
  - Pricing for each subscription
  - Organizational pricing incentives from Microsoft (if applicable)
  - Who approved the purchase (if applicable)
  - Cost allocations within the organization (if applicable)

Step 2: Decide on capacity licenses

After you've reviewed the capacity subscriptions, you should decide on capacity licenses.

The use of capacity can play a significant role in your strategy for creating, managing, publishing, and distributing content. Your decisions related to capacity licenses are in addition to the per-user licenses, which were described earlier.

Decide whether you need a capacity license

As you begin to analyze the needs for a capacity license, it's important to have clarity on your architecture needs and user needs that'll inform the decision to use a capacity.

Here are some questions you might initially explore.

- **Data architecture**: What type of data architecture investments are currently in progress? How will they affect the choices you make? Do you have large semantic models that contain large volumes of data?
- **Fabric experiences**: Which experiences are currently in use or are planned for future use? For example, you might be currently using the Power BI experience but you intend to invest in a lakehouse architecture in Fabric, which is part of the Data Engineering experience.
- **Data and BI requirements**: Are there business requirements to address analytical needs that are currently unmet? How do the requirements correlate to architecture (and licensing) decisions?
- **Consumers**: How many view-only consumers do you have?
- **Authors**: How many content authors do you have? Are authors centralized, distributed across various business units, or both?
- **Usage patterns**: What are the current usage patterns for user queries and data refresh? Are usage patterns predictable and consistent from day to day?

> **Note**
>
> As you work through the process of analyzing and planning for capacity licenses, it's possible you'll determine a need to perform a full technical assessment.

Decide which capacity license you need

When you determine that you need a capacity, you'll need to decide which type of capacity license is best suited.

Here's each of the capacity licenses and their uses and suitability.

- **Fabric capacity (F SKUs)**: F SKUs are purchased in Azure (note the pricing is regional). F SKUs have some advantages (that aren't available with P SKUs),

including the ability to:

  - Scale the capacity to resize it up or down at any time. This ability to scale allows you to adjust the size—and cost—as you better understand your workload.
  - Pause the capacity at any time. This functionality is useful for infrequently used capacities.
  - Test the capabilities with a Fabric trial before committing to a purchase.
  - Use a lower tier capacity license for small workloads as a way to reduce the cost.
  - Choose the preferred commitment level:
    - **Pay-as-you-go**: The pay-as-you-go pricing model has no usage commitment. You can resize the capacity up and down as needed, and even pause it. It's suitable when you want flexibility.
    - **Reservation**: The reserved pricing model involves a precommitted size (SKU) for a specific period, which results in a lower cost option than pay-as-you-go. However, a reserved instance can't be paused, so it's suitable when you need to run a capacity 24/7.
  - Use organizational pricing incentives. If you have a monetary agreement in place with Microsoft, programs such as the Microsoft Azure Consumption Commitment (MACC) apply to F SKUs.
  - Use Microsoft Cost Management capabilities to monitor and track costs.
- **Power BI Premium per capacity (P SKUs)**: P SKUs are purchased in the Microsoft 365 admin portal. Power BI Premium uses a reserved pricing model, so it runs 24/7 and can't be scaled or paused. You can't purchase P SKUs after July 1, 2024.
- **Power BI Premium (EM SKUs)**: EM SKUs are a specialized type of Power BI Premium capacity license that are purchased in the Microsoft 365 admin portal or through volume licensing (available through your Microsoft account manager). EM SKUs are targeted at straightforward embedding scenarios, such as embedding a report in an application. The EM SKUs offer is a subset of functionality that's available in the P SKUs. They have less compute power, and no access to the Power BI service. Also, EM SKUs don't support Fabric experiences. For more information, see Capacity and SKUs.
- **Power BI Embedded (A SKUs)**: A SKUs are purchased in Azure (however, this offering is different from the F SKUs previously described). Power BI Embedded is primarily targeted to independent software vendors (ISVs) that want to embed Power BI content in their applications. A SKUs don't support Fabric items. For more information, see the Embed for your customers usage scenario.

> **Tip**
>
> You can also use F SKUs to embed Power BI content as you would with the A and EM SKUs. For more information, see **Power BI Embedded with Microsoft Fabric**.

The remainder of this article focuses on the F SKUs and P SKUs.

Decide to use one or multiple capacities

A key decision is whether to use one larger capacity or multiple smaller capacities. Your choice should involve the following considerations.

- **Level of centralization versus decentralization**: How important is centralized management versus decentralized management for the capacity? When you have a distributed or mesh architecture, it's more likely that multiple capacities will be necessary to allow different teams to manage their own capacities.
- **Data storage location**: Do you have regional, industry-specific, or organizational data residency requirements? The geographic location where data is stored is correlated to the capacity using the Multi-Geo feature.
- **Resource isolation**: What level of resource isolation per capacity is required? For example, you might need to create different capacities for specific business units. Or, you might create a capacity specifically to support workspaces for a domain.
- **Compute resources**: What level of compute resources are required for each capacity? For example, if you choose to provision two F32 capacities instead of one F64, then fewer capacity units are available for both capacities because they're split. The capacity units translate to constraints for each SKU, such as the maximum memory size for a semantic model.
- **Features required**: In addition to the level of compute power, are certain capabilities necessary? For example, an F64 (or P1) or higher capacity allows users with a Free license to view BI content or use Copilot.
- **Cost**: Do you need to track or allocate cost separately for each SKU? That's easier to accomplish when you have separate capacities.

Decide size of capacity

At this point, you're ready to select a specific capacity SKU. Consider what environment(s) you intend to run: development, test, and/or production. It's common for development and test environments to run on a smaller capacity than what's required for a production environment.

To gain confidence about the capacity size that you'll need, consider doing load testing of the capacity. For more information, see Capacity Planning and Assess Your Capacity Load.

Decide needs for scaling up and down

It's important to consider scalability needs during the license planning process because it contributes to cost. For example, you might have needs to occasionally resize (or pause) an F SKU capacity. Alternatively, you can set up autoscale to handle occasional or unexpected bursts in P SKU capacity usage levels. For more information, see Resize capacity.

> **Tip**
>
> You can think about scalability in two ways.
>
> - *Scaling up or down* is when you add or remove resources (for example, scaling up to an F16 from an F8 capacity).
> - *Scaling out* is when you add more capacities (for example, you could purchase two F8s instead of an F16). However, capacities don't use combined resources (like a **data gateway cluster** would when you scale out). Therefore, if you're considering scaling out to multiple capacities, be aware that separate capacities purposefully work in isolation.

Decide whether on-premises or a hybrid license is needed

Power BI Report Server (PBIRS) is a simplified reporting solution. It's targeted at organizations that want to implement a hybrid approach where Power BI content might be published to either the cloud-based Fabric portal, to Power BI Report Server, or both. You can install Report Server on a machine that's running within on-premises infrastructure, or on an Azure virtual machine (with Azure Hybrid Benefit).

You can obtain a Report Server license in one of two ways.

- Power BI Premium (P SKU) subscription
- SQL Server Enterprise Edition with Software Assurance (SA)

> **Note**
>
> An author who publishes content to the Report Server is required to have a Pro license.

> **Tip**
>
> Conduct a technical **proof of concept** to ensure that Power BI Report Server will meet your needs. Be aware that feature parity with the Fabric portal isn't a goal.

Also, when publishing content to Power BI Report Server, the use of **Power BI Desktop for Report Server** (different from regular Power BI Desktop) is recommended.

For more information, see Licensing Power BI Report Server.

> **Important**
>
> We strongly recommend that you refer to your volume license agreement and speak with your Microsoft account representative for specific details. For example, the Report Server licensing agreement includes limits related to the number of cores on the target machine.

Step 3: Update capacity licenses

At this point, your existing capacity subscription information is available, and you've made purposeful decisions. Now you're ready to make updates that are necessary.

The following topics are actions that might be appropriate.

Adjust capacity subscription

Sometimes you might need to make an adjustment to an existing capacity subscription based on what you found during your review (Step 1) and the decisions made (Step 2).

Here are some examples of changes you might make.

- **Change to reserved pricing**: The workloads running on your capacity are consistent and need to run 24/7. Therefore, to save costs, it's prudent to change the Fabric capacity from pay-as-you-go pricing to reserved pricing.
- **Change to pay-as-you-go pricing**: The workloads running on your capacity change regularly and would benefit from the ability to frequently scale up and down. In this case, the most cost effective approach might be to change to pay-as-you-go pricing.

> **Important**
>
> Contact your Microsoft account manager should you have any questions or need clarification about subscription costs and options.

Resize capacity

You might discover there's a need to resize your capacity when a smaller or larger size would better meet your needs.

There are two ways to handle resizing of a capacity.

- **Manual scaling**: You can choose to resize (or pause) an F SKU capacity in the Azure portal. That's helpful when you're troubleshooting performance issues, or you have a known period of time when the load will be higher (such as the final week of every month).
- **Automated scaling**: You can enable autoscale to handle occasional or unexpected bursts in P SKU capacity usage levels without requiring any manual effort. Autoscale can respond to these bursts by elastically resizing the resources to support the increased workload. Automated scaling up reduces the risk of incurring performance or user experience challenges, in exchange for extra cost. If your capacity isn't well managed, autoscale might trigger more often than expected, which could lead you to consider a larger capacity size.

Step 4: Document capacity licenses

Depending on your internal processes, you might choose to create documentation that will augment the information that's available in the portal for your capacity subscriptions.

You can build on the information captured in Step 1 by including the following details in your documentation.

- Key decisions, including more context or detail
- Who approved capacity license purchases, and when
- Timing or pending action items
- Governance requirements related to capacity licenses
- Auditing requirements related to capacity licenses
- A snapshot of capacity license information

Step 5: Manage capacity licenses

Capacity management and subscription management are two separate topics. However, they're highly interrelated. Both need attention on an ongoing basis. The following topics are aspects to consider.

Create a process to accept capacity requests

You should create a repeatable, documented, process for users to request a capacity. It typically involves creating an online form. Request information that you'll need to assess the capacity request, such as:

- Purpose and type of content to be hosted on the capacity.
- Who will administer the capacity.
- Whether the capacity will run 24/7 or not.
- Where the data needs to be stored geographically.
- How to charge or allocate the cost internally.

Monitor and understand capacity usage

Here are some considerations for monitoring and understanding capacity usage.

- Analyze the load to determine whether the current capacity size (SKU) works well for the specific data and BI needs. Be sure to understand how capacity units (CUs) work. Analyze any bursting and smoothing activity to analyze whether you're using your capacity efficiently over time, or whether it's consistently overloaded. You can analyze capacity usage by using the Fabric Capacity Metrics App.
- Resize a capacity when you discover that it's too large or too small to serve your current needs. Changing the size is the same as changing the SKU. The change affects the pricing tier.
- Create a new capacity when you need to:
  - Segregate a workload.
  - Store data in a different region.
  - Assign different capacity administrators (for decentralized capacity administration).
- Create user training or communicate with authors when you find that they can take specific action to improve the efficiency of the capacity.

Set up notifications

If your capacity is regularly overloaded, it indicates that you might need to purchase a larger capacity (scale up) or create other capacities (scale out)—or move content to a different capacity. For these reasons, capacity management and licensing management have a significant impact on each other.

You should set up the following notifications to be kept informed.

- Set the *Enable notifications for service outages or incidents* tenant setting so Fabric notifies you when the capacity becomes overloaded, or when an outage or incident occurs.

- Set up Azure Monitor alerts to be notified when certain capacity metrics exceed a threshold. This feature is available for F SKUs, A SKUs, and autoscale for P SKUs.

Review and optimize capacity costs

You should regularly review and manage your billing. Consider the following options to optimize cost.

- **Differences in purpose**: For example, you might choose to use a smaller capacity size (such as an F16) for test workspaces, and a larger capacity size (such as an F64) for production workspaces.
- **Efficient use of compute resources**: Use the Fabric Capacity Metrics App to determine whether the compute resources are being used efficiently and whether there's any potential to optimize cost.
- **Monitoring compute cost**: Monitor the cost of your capacity, and how frequently capacities are scaled up and down. Consider using cost analysis, spending limits, or budgets with Microsoft Cost Management.
- **Billable storage cost**: Check the billable storage for each workspace in the Fabric Capacity Metrics App. For Fabric items, storage cost is computed separately from the compute cost. Also, check the disaster recovery capacity setting. This setting will have an impact on the billable storage costs.
- **Scale up and down**: Create a process to automatically scale the capacity up or down (or pause it, when applicable) when the workload is intermittent yet predictable.
- **Cost chargebacks**: When you need to distribute costs to other departments, create a chargeback process to allocate subscription costs.

Step 6: Audit capacity licenses

It's important to have a process to regularly audit capacities. A Fabric administrator will need to work together with other administrators to obtain this information (such as a global administrator, billing administrator, or Azure administrator).

> **Tip**
>
> This section focuses on auditing for subscriptions, licenses, and trials. There are many additional auditing and monitoring aspects for capacity, including monitoring of usage and performance (and identifying needs to scale up or down) by using the **Fabric Capacity Metrics App**. You will also want to monitor the **activity log** for situations such as when capacity settings change, capacity administrators are

updated, capacity contributors are added, or when workspaces are assigned to a capacity.

Here are some actions to identify when to audit subscriptions, trials, and cost for capacities.

- **List of active capacities**: The Get Capacities as Admin REST API can provide you with information such as SKU, state, administrators, and region for all of the capacities in your tenant. It's an admin API that returns a snapshot as of a point in time. If you capture this data regularly, you can compare snapshots (for example, this week versus last week) to detect changes that have occurred.
- **A new Fabric trial was started by a user**: Look for the *ChangeCapacityState* activity in the activity log. The *CapacityState* property indicates a new Fabric capacity was provisioned. The *ItemName property* indicates that it's a trial capacity, the SKU, and its ID.
- **A new Fabric capacity was created, or an existing capacity was resized**: Look for the *Update Fabric Capacity Create* operation in the Azure Monitor activity log. You can also view Fabric capacities in the Azure portal.
- **The compute engine for a Fabric capacity was paused or restarted**: In Azure Monitor activity log, look for the *Suspend* operation, or the *Resume* operation. You can also view the state of a Fabric capacity in the Azure portal.
- **A new Premium capacity was created**: In the activity log, look for the *ChangeCapacityState* activity. The *CapacityState* property indicates that it was provisioned as a new capacity. You can also view products that have an active subscription in the billing area of the Microsoft 365 admin center.
- **Monitor cost for a Fabric capacity**: Use Microsoft Cost Management capabilities to analyze costs for Microsoft Fabric capacities and other Azure services.
- **Monitor cost for Premium capacity**: You can view invoices in the billing area of the Microsoft 365 admin center.
- **A workspace was assigned to, or removed from, a capacity**: Look for the *MigrateWorkspaceIntoCapacity* activity or the *RemoveWorkspacesFromCapacity* activity in the activity log.

Tip

When you navigate the Azure Portal, don't be confused by the *Service Fabric* resources. These resources are different services from Microsoft Fabric.

Important

The information presented in this step isn't intended to be an all-inclusive list of ways to audit the data. Rather, it's intended to provide you with ideas to get you started with your auditing efforts. For other ideas, we recommend that you consult with your billing administrators.

Checklist - When planning for capacity licenses, key decisions and actions include:

- **Conduct a review**: Determine the current state by reviewing current capacity subscriptions.
- **Identify existing policies and decisions**: Compile existing internal policies or previous decisions related to capacity subscriptions so that the information is readily available.
- **Discuss and decide**: Schedule workshops to make key decisions related to capacity subscriptions. Involve all relevant decision makers, stakeholders, and the executive sponsor, when appropriate.
- **Create documentation**: Compile the information gathered about capacity subscriptions, and document key decisions for future reference.
- **Make updates**: Update capacity subscriptions based on the decisions made in the workshops.
- **Create a process to manage capacity requests**: Set up a process for how users can request a new capacity.
- **Set up auditing**: Create auditing processes so you can track activities related to capacity subscriptions and trials.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Tenant administration

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces key considerations for administering a Fabric tenant. This article is targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization.
- **IT and system administrators**: Other administrators who collaborate with Fabric administrators to oversee and integrate systems within the organization.
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI and supporting Power BI users in the organization. These teams make key decisions and collaborate with the Fabric administrators.

Managing the Power BI service is one key aspect of *system oversight*. For more information, see System oversight. The routine activities associated with system oversight are commonly known as *system administration*. System administration activities are critical to help ensure that content consumers and content creators consistently have a good experience with Power BI.

As described in the Fabric adoption maturity levels article, organizational adoption refers to the effectiveness of governance and data management practices to support and enable enterprise BI and managed self-service BI. Therefore, administrators who manage analytics and BI platforms can have a sizeable and direct influence on the success of your analytics adoption efforts.

> **Note**
>
> Administering Fabric capacity (or Premium capacity) and managing the Power BI service are different concepts. While most organizations have just one Power BI tenant, an organization can provision multiple capacities for different workloads or business units.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Define scope of responsibilities

There's no single definition of the Fabric administrator role, meaning that the role and routine responsibilities of a Fabric administrator can vary between different organizations. What shouldn't vary is that the role can—and should—evolve over time as organizational priorities and goals change.

From a strategic perspective, a Fabric administrator should focus on:

- **Governance**: Enacting governance guidelines and policies to support enterprise BI and managed self-service BI.
- **User empowerment**: Facilitating and supporting the internal processes and systems that empower the internal user community to the extent possible, while adhering to organizational regulations and requirements.
- **Adoption**: Allowing for broader organizational adoption of Fabric with effective governance and data management practices.

Attempting to balance governance, user empowerment, and adoption goals inherently leads to competing priorities. Ideally, it leads to productive debates about priorities. Clarifying and communicating your expectations for various roles and responsibilities can help to avoid unacceptable levels of friction and conflict.

Consider the following three examples of Fabric administrators.

- **High focus on user enablement**: Riley is a Fabric administrator who works for a large global organization that's made significant investments in managed self-service BI. To enable self-service BI capabilities for users throughout the organization, Riley spends much time coordinating decisions and actions with the Center of Excellence (COE) and other administrators. When needed, Riley steps in to support existing BI solutions.

- **High focus on governance and compliance**: Parker is a Fabric administrator who works for a highly regulated organization. In this organization, most BI development is handled by BI developers within a centralized enterprise BI team. Parker's administrative responsibilities are primarily focused on areas such as auditing, information protection, and security.
- **High involvement in content creation**: Morgan is a Fabric administrator who works for a small organization that's just getting started building its data culture. Presently, the organization has only a few content creators. In addition to system oversight responsibilities, Morgan is a BI developer who regularly creates and publishes content. Sometimes, Morgan gets involved in a co-development project to mentor a colleague, which helps grow BI expertise in the organization.

Checklist - When planning the scope of responsibilities, key decisions and actions include:

- ✓ **Decide on strategic focus**: Determine what the strategic focus for your Fabric administrators should be. Get clarity on goals and priorities to use when decisions (and trade-offs) need to be made.
- ✓ **Identify specific roles and responsibilities**: Determine what the specific expectations are for your Fabric administrators. Clearly document their roles and responsibilities, and update job descriptions with Human Resources when appropriate.

Appoint administrators

The actions of a Fabric administrator have significant impact on the user experience, data culture efforts, governance efforts, who owns and manages content, and organizational adoption efforts. So, it's critical that you appoint the right people to administrator roles.

Here are some key points to consider when you select your administrators.

- **Remain conscious of the highly privileged nature of the role**. An administrator role is a high-privilege role because it's authorized to manage a wide array of tenant settings, workspace access, personal workspace access, and it can view all tenant metadata. For more information, see Power BI administration.
- **Carefully select who's well-suited to be an administrator.** An administrator often needs to collaborate with users and the COE. For this reason, they should

understand business intelligence concepts and what users want to accomplish. Someone who has an overbearing personality or who has the tendency to strictly limit what users are permitted to do probably isn't well-suited to managing a self-service BI platform.

- **Choose 2 to 4 administrators.** Because it's a high-privilege role, appoint only a few administrators. Strike the right balance: Having too many administrators increases the risk of unapproved changes occurring, while having too few administrators increases the risk that the system won't be sufficiently supported.
- **Allow for occasional administrators.** If you have users who *occasionally* need Fabric administrator rights, consider implementing Privileged Identity Management (PIM). PIM lets you assign *just-in-time role permissions* that expire after a few hours. This process provides a useful way to balance risk (of having too many full-time administrators) versus usability (allowing progress to occur). That's especially true in larger, decentralized organizations. When using PIM, delegation of the administrator role is logged and can optionally involve an approval workflow to grant rights.
- **Make Fabric administration a priority.** Often, administration of a BI platform is just one of many responsibilities. Consider how you'll ensure that users are well supported, and your system is sufficiently managed.
- **Regularly review who's assigned to all the related roles.** There are three roles that are permitted to manage the Power BI service: Fabric administrator and Power Platform administrator. It's important that you audit the membership of these roles regularly.

For more information, see About admin roles in the Microsoft 365 admin center.

Checklist – When appointing administrators, key decisions and actions include:

- ✓ **Identify current Fabric administrators**: Review who's currently assigned to the role of Fabric administrator. Also consider the Power Platform administrator and Global administrator roles in this review.
- ✓ **Appoint Fabric administrators**: To reduce risk, appoint 2 to 4 people to the Fabric administrator role. If there are currently more than four people assigned, reduce the number of people assigned to the Fabric administrator role.
- ✓ **Use PIM for occasional administrators**: Identify whether you have people who legitimately, but occasionally, need Fabric administrator rights. Implement PIM to assign just-in-time role permissions that expire after a few hours. Document and communicate how the process works, which could include an approval workflow.

- ✓ **Assign backups and conduct cross-training**: Check the status of cross-training and documentation that's in place for handling Fabric administration responsibilities. Ensure that a backup person is trained so that priority user needs can be met in a timely and consistent way.
- ✓ **Regularly audit administrators**: Verify who's assigned to the Fabric administrator role on a regular basis.

Collaborate with other administrators

Although the Fabric administrator is a high-privilege role, it's limited to managing Fabric. Therefore, at times you'll need to collaborate with other administrators.

Here are some common reasons why a Fabric administrator collaborates with other system administrators.

- **Device setup and installations**: You might need to work with IT, an infrastructure team, or a desktop support team in order to install, update, or manage user devices.
- **Subscriptions and license purchasing**: The Billing admin role in Microsoft 365 is required to manage subscriptions and purchase licenses. A billing administrator might also be responsible for cost analysis and management. For more information about centralized and decentralized (self-service) ways to manage licenses, see User licenses.
- **License assignment and user management**: The License admin role in Microsoft 365 is required to assign licenses (that have been purchased) to specific users. The User admin role is required to manage the properties of a user. It's helpful to work with a user administrator when you're planning to implement automation based on user properties (for example, automatic license assignment or automatic group assignment). For more information, see Commonly used Microsoft 365 admin center roles.
- **Microsoft Entra administrators**: There are various reasons why you need to work with Microsoft Entra administrators. Often, reasons include the need to set up or manage users, groups, and service principals. For more information, see Tenant-level security planning.
- **Source data access**: You might have to work with a system administrator or database administrator in order to get access to data on behalf of content creators. Sometimes, it might also be necessary to request access on behalf of content consumers when semantic models enforce data security based on the identity of content consumers.
- **Data loss prevention and data classification**: You might need to collaborate with your Microsoft Purview administrator for governance and information protection.

- **Teams integration**: When integrating Power BI with Microsoft Teams, you might need to collaborate with a Teams administrator.
- **OneDrive and SharePoint integration**: When integrating Power BI with OneDrive or SharePoint, you might need to collaborate with other administrators.
- **Workspace administration**: You might need to collaborate with a Fabric workspace administrator to plan, organize, or secure content within specific workspaces.
- **Lifecycle management**: When deploying and managing changes to content, you might need to collaborate with a deployment pipeline administrator or an Azure DevOps administrator.
- **Premium capacity management**: You might need to collaborate with a capacity administrator when managing a Premium capacity.
- **Data gateway management**: You might need to collaborate with a gateway administrator to manage and secure an on-premises data gateway. For more information, see Manage gateways.
- **Power Platform administration**: You might need to integrate solutions between Power BI and other Power Platform apps (such as Power Automate or Power Apps).
- **Azure administration**: You might need to work with an Azure administrator to set up, access, and secure other Azure services that you want to integrate with Power BI.
- **Security administration and auditing**: Your organization could have specific compliance, security, or privacy requirements. In this case, you might need to collaborate with your security team to identify and mitigate risks.
- **Networking**: When connecting to different data sources and systems, you might need to work with your network administrators for performance and security reasons.
- **Mobile device administration**: You might need to collaborate with an Intune administrator in order to manage mobile device policies and security.

> ⓘ **Important**
>
> A Fabric administrator shouldn't make decisions or take action (such as changing **tenant settings**) on their own. All key decisions should be discussed, planned, and documented. In addition to collaborating with other administrators, be sure to fully involve the **COE** and your **BI strategy working team**. It might also be appropriate to involve your **executive sponsor** for strategic decisions.

Checklist – When collaborating with other administrators, key decisions and actions include:

- ✓ **Determine who will manage device setup and installations**: Make sure you know who manages devices for your organization. Get familiar with their processes and requirements. Be prepared to work with them as needed.
- ✓ **Identify who will manage subscriptions and licenses**: Make sure you know who manages subscriptions and licenses for your organization. Get familiar with their processes and requirements. Be prepared to work with them as needed.
- ✓ **Identify who will assign licenses and manage users, groups, and service principals**: Make sure you know who manages users, groups, and service principals for your organization. Get familiar with their processes and requirements. Be prepared to work with them as needed.
- ✓ **Determine any other administrators to consult**: As you work through your implementation planning process, identify other relevant administrators. Invite them to pertinent meetings and involve them in relevant decision-making processes. Update documentation and processes as needed.

Govern the Power BI service

Governing and managing the Power BI service is one of the key responsibilities of a Fabric administrator. This section describes how to govern and manage many common settings and features in the Fabric admin portal.

- Tenant settings
- Domains
- Workspaces
- Embed codes
- Organizational visuals
- Azure connections

Govern tenant settings

Tenant settings are the main way to control which Power BI features and capabilities are enabled, and for which users in your organization. Managing tenant settings is one of the most important responsibilities for a Fabric administrator.

Because content creators and content consumers can easily learn about available features and capabilities in Power BI (from documentation), it can result in frustration when they can't do something they expect. It can also lead to user dissatisfaction and less effective organizational adoption, user adoption, and solution adoption.

Here are some common questions that confused and frustrated users ask.

- Why can't I create a workspace?
- Why can't I export data?
- Why doesn't my custom visual work?
- Why can't I certify a semantic model?
- Why can't I assign a sensitivity label?
- Why can't I push an app to specific end users?

> **Important**
>
> Each tenant setting should align with the governance guidelines and policies in your organization. When a Fabric administrator decides on their own to enable or disable settings, it's usually a clear indicator that you need to improve and refine your governance processes.

The remainder of this section describes the following process to manage tenant settings.

1. Review tenant settings
2. Decide on tenant settings
3. Update tenant settings
4. Document tenant settings
5. Manage tenant settings
6. Audit tenant settings

Step 1: Review tenant settings

It's important that you review each tenant setting to clearly understand the current state of your tenant. While you should review all tenant settings, you might prioritize specific critical settings to review first based on a risk assessment.

Here are some factors to consider during the review process.

- **Setting**: Is the specific tenant setting currently enabled or disabled?
- **Permissions**: Is the specific tenant setting applicable to the entire organization? Or is it available, or denied, to specific security groups?

> **Note**
>
> Some tenant settings are enabled by default, while others are disabled by default.

You can compile the current state of your tenant settings in one of two ways.

- View the tenant setting state within the admin portal.
- Programmatically extract the tenant setting state by using the Get Tenant Settings REST API.

The following table presents how you could record the current state of your tenant settings.

Expand table

Date last reviewed	Tenant setting	Current value	Security groups allowed	Security groups not allowed	Tenant setting delegated to other admins
Oct 15, 2023	Create workspaces	Enabled for specific security groups	Power BI workspace creators, Power BI administrators, Power BI COE, Power BI support	N/A	N/A
Oct 15, 2023	Export to Excel	Enabled for entire organization except specific security groups	N/A	Sales Team-Europe	N/A
Nov 1, 2023	Use semantic models across workspaces	Enabled for entire organization	N/A	N/A	N/A
Nov 1, 2023	Certification	Enabled for specific security groups	Power BI certification SMEs	N/A	Domain admins can enable / disable
Nov 5, 2023	Allow specific users to turn on external data sharing	Enabled for specific security groups	Power BI external data sharing	N/A	N/A

For other considerations about security groups, see Planning for Power BI groups.

For more information about allowable tenant setting states, see How to use the tenant settings.

⊗ **Caution**

Sometimes when an administrator monitors the tenant, they discover a situation that isn't ideal. For example, they might see too many data exports in the **user activity data**. In this instance, the administrator might be tempted to disable the related tenant setting. However, before disabling a feature entirely, it's important to first understand why users are relying on certain techniques. That's because prohibiting features can lead to user frustration which can tempt users to create workarounds. Consider instead that a solution might need to be redesigned, or perhaps further user education and training could mitigate concerns.

Step 2: Decide on tenant settings

After you've reviewed your current tenant settings, it's time to examine the decision-making process. For each tenant setting, use workshops to actively discuss and determine which features and capabilities should be allowed or disallowed, and for which users.

Remember that every tenant setting represents a governance decision. Therefore, expect to collaborate with others to arrive at the right decision. Depending on your circumstance, the decision-making process could involve collaborating with the COE, executive sponsor, security team, BI team, or others (such as stakeholders or champions).

> **Tip**
>
> One of the biggest challenges is deciding what to do when you encounter inconsistencies between an existing tenant setting versus your goals and purposeful decisions. Be prepared to resolve these conflicts when you encounter them.

Here are some factors to consider during the decision-making process.

- **What governance decisions already exist?** When possible, refer to existing decisions you've made. Always aim to be consistent and efficient. Also, you need to be aware of internal or external compliance requirements. Where applicable, tenant settings should align with broader governance policies. For example, there might be a current governance policy that specifies when, who, and how data can be shared outside of the organization.
- **Who makes new governance decisions?** When you need to make a decision about a tenant setting, involve all relevant parties in the conversation. Usually, Fabric administrators alone aren't in the best position to make decisions about tenant

settings. For information about assembling a working team and running workshops, see BI strategic planning.

- **Is the decision temporary?** Sometimes, a tenant setting is set for a short period of time only. Usually that's done to allow the COE, BI, and IT teams time to become familiar with a new feature before it's released more broadly throughout the organization. That way, you can verify alignment with governance guidelines, and the user community will be well supported.
- **Will different business units or teams be handled differently?** In larger organizations, a single approach rarely works. To accommodate the needs and skills of different teams, sometimes you might need to set up tenant settings differently for different audiences. Always aim to allow capable teams to operate with as much flexibility as possible, within your governance guidelines.
- **Is there an approval process?** Depending on the specific subject, approval might be required. That's especially true when a tenant setting relates to security or compliance.
- **What's the schedule to re-review each decision?** Schedule a review of each tenant setting decision regularly. Twice a year is a good schedule for this purpose.

The following table presents how you might record your decisions.

Expand table

Date of decision	Decision made	Decision makers involved	Decision approved by	Tenant setting affected	Pending action
Oct 15, 2023	The COE will train approved content creators on best practices for creating and managing workspaces. Approved creators can be from any business unit.	COE and stakeholders	Ellis Turner (executive sponsor) and Taylor Phillips (COE lead)	Create workspaces	Review existing members of the *Power BI workspace creators* group
Oct 15, 2023	Export to Excel will be permitted. The COE will track the action in the activity log and contact users who overuse it. A temporary limitation for the Sales Team-Europe exists due to	COE and security	Ellis Turner (executive sponsor) and Jessie Irwin (Chief Technology Officer)	Export to Excel	Follow up on Europe issue in 60 days

Date of decision	Decision made	Decision makers involved	Decision approved by	Tenant setting affected	Pending action
	an issue currently under investigation.				
Nov 1, 2023	We strongly encourage the use of shared semantic models to encourage data reuse, minimize data silos, and reduce data duplication.	COE	Ellis Turner (executive sponsor)	Use semantic models across workspaces	N/A
Nov 1, 2023	The COE will train approved content creators on the process and requirements for certifying reports and data assets. Approved creators can be from any business unit.	COE and stakeholders	Ellis Turner (executive sponsor) and Taylor Phillips (COE lead)	Certification	Review existing members of the *Power BI certification SMEs* group
Nov 5, 2023	The organization's data sharing policy covers how data can be shared outside of the organization. All users are required to review and acknowledge this policy annually. Training from the COE will help users to comply while working in Power BI.	COE, security, and compliance	Jessie Irwin (Chief Technology Officer)	Allow specific users to turn on external data sharing	Review existing members of the *Power BI external data sharing* group

Consider including other contextual information that explains why a decision was made. Also, include links to the location of existing related documents or policies.

> **Note**
>
> The examples presented in the table intentionally demonstrate how to balance user empowerment, compliance, and internal requirements. For additional considerations, see Governance.

Step 3: Update tenant settings

Now that your existing tenant settings and purposeful decisions are available, you're ready to update your tenant settings.

Considerations during the update process include:

- **Who will handle the update?** If you have several Fabric administrators, it's ideal that one or two Fabric administrators be primarily responsible for updating the tenant settings. The goal is to ensure the process is consistent, well-understood, and well-controlled.
- **What testing process is in place?** Depending on the tenant setting, there might be other effects when it's changed. Conduct testing before making widespread changes. For an example, see Control the use of semantic models across workspaces.
- **Is there a change management process?** Consider how to avoid discrepancies between decisions, documentation, and the resulting updates. Communication between teams is a key success factor in change management. Depending on your auditing requirements, you might choose to maintain an internal change log to track who made what change, when, and why (to record more detail beyond what's captured in the activity log).
- **How will communication with users be handled?** When making a change that affects what users are able to do, be sure to communicate the change. Always aim to avoid user frustration and unnecessary support requests.

Step 4: Document tenant settings

At this point, make sure you have a repeatable method in place for documenting the tenant setting values. For a simplified example, see the table in Step 1 above, which is about reviewing tenant settings.

Here are some aspects to consider for documentation.

- **Extract snapshot data**: When documenting tenant setting values, we recommend that you create a new point-in-time snapshot regularly. For this purpose, creating a snapshot once a week is a good frequency. Use the Get Tenant Settings REST API to automate the process.
- **Provide admin access to snapshot data**: Administrators, the COE, and internal auditors should have access to all the snapshot data. To identify any tenant settings that have changed, compare two snapshots (for example, this week compared with last week). Data from the activity log complements the snapshot

data to provide a complete understanding of what's changed, when, and by whom. For more information, see Step 6 below, which is about auditing tenant settings.

- **Provide a summary of current settings to users**: The tenant setting values are one type of documentation that can be made available to your community of users in your centralized portal. It's a helpful reference for a user who, for example, doesn't understand why a feature isn't available to them. The documentation can also help users know which security group to request to join should they want to use a feature. To reduce the level of manual effort, the latest snapshot results from the REST API can be distributed to users as a Power BI report. Depending on your needs, you might need to merge the snapshot of data with other data that's manually maintained (such as the description of the tenant setting, justification for the setting, links to additional information, or link to a form to request access).

> **Note**
>
> As previously described in Step 2, you'll also have documentation related to the decision-making process. Typically, that type of documentation is available only to the COE and administrators (rather than all Power BI users). For a simplified example, see Step 2.

Step 5: Manage tenant settings

The tenant settings need attention on an ongoing basis. Here are some aspects to consider.

- **New tenant settings**: How will you know when a new tenant setting is available? Since Fabric is a cloud service that continues to evolve, you can expect new tenant settings will be introduced regularly. One way to become aware of new tenant settings is to view the message(s) announced at the top of the tenant settings page in the admin portal. Another way is to compare the data you extract from the current snapshot with the previous snapshot (described previously in Step 4).
- **Changes to existing tenant settings**: How will you know when the behavior of an existing tenant setting has changed? Tenant setting changes are usually announced in the Power BI blog or the Fabric blog. Be sure to follow these blogs to become aware of any notifications.
- **Ongoing user requests**: How will you manage user requests? For example, a user who wants to certify content knows to submit a request to become a member of a specific security group that allows it. That's an effective workflow made possible by publishing documentation of your tenant settings for users to reference (as

described in Step 4 above). You might choose to use a form for gathering these types of requests. Or you could route requests through your help desk.

- **New security groups**: If a tenant setting is to be limited to specific security groups, does a suitable security group already exist? Or does a new security group need to be created? How will you coordinate getting a new security group created when necessary? For more information, see Strategy for using groups.

Step 6: Audit tenant settings

Lastly, it's important to have a process for regularly auditing tenant settings. Here are some actions to identify when you audit the tenant settings.

- **A tenant setting has changed**: Look for changed values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. This activity only indicates that something has been updated (whether it's enabled or disabled, or different security groups have been assigned). For an understanding of what's changed, you'll need to compare the current snapshot with the previous snapshot (described previously in Step 4).
- **A new tenant setting has been introduced**: Look for the message(s) in the admin portal, or compare the data you extract from the current snapshot with the previous snapshot (described previously in Step 4).
- **A group membership has changed**: In some situations, it might not be enough to know which security group is assigned. You might need to determine the security group membership, which can comprise individual users and service principals. You can source group membership data by using Microsoft Graph. For more information, see Users and groups data.

Also, you might want to receive alert notifications when a tenant setting has updated. You can use the audit log alerting capabilities from Microsoft 365 or Microsoft Defender for Cloud Apps to be notified when a tenant setting has changed, and by whom, by using the *UpdatedAdminFeatureSwitch* audit log event. For more information about enabling alerts, see Tenant-level auditing.

Checklist – When planning and managing tenant settings, key decisions and actions include:

- ✓ **Conduct a review of all current tenant settings**: Determine the current state by reviewing all the tenant settings. Identify which security groups are assigned to

each setting.

- ✓ **Identify existing policies and decisions**: Compile existing governance policies or previous decisions so they're readily available.
- ✓ **Discuss and decide**: Schedule workshops to consider and determine what should be allowed or disallowed, and for which users. Ensure all decisions are aligned with data culture objectives, governance guidelines, and internal policies. Be sure to involve all relevant decision makers and stakeholders for each topic. Obtain additional approval when appropriate.
- ✓ **Create a schedule to re-review decisions**: Set up a schedule to re-review decisions and tenant settings on a regular basis (such as twice a year).
- ✓ **Make updates**: Update the tenant settings based on the decisions made in the workshops.
- ✓ **Extract snapshot data with the API**: Use the *Get Tenant Settings* REST API to make the process more efficient, potentially automating the creation of snapshots on a scheduled basis.
- ✓ **Create documentation for users**: Create documentation of your tenant settings for your community of users. Publish the documentation to your centralized portal. Include the security groups to which a user must belong to access a feature or capability.
- ✓ **Create a process to handle user requests**: Set up a process for how users can request access to a feature or capability.
- ✓ **Set up a process for routine management of tenant settings**: Create a schedule so that you can identify new tenant settings as soon as possible.
- ✓ **Set up auditing**: Create an auditing process so you can track when a tenant setting has changed, and by whom.

Govern domains

Domains are used in Fabric to group together two or more workspaces that have similar characteristics. For example, you might create a domain to group all your sales workspaces together, and another domain for your finance workspaces. A domain represents a single management boundary. Usage of domains can also facilitate workspace ownership and management responsibilities that are distributed across the organization.

For more information about planning for domains, see Workspace domains.

> **Tip**

The Fabric domains described in this section are a different concept from **domains in Microsoft 365**.

Step 1: Review domains

The first step is to review the existing domains and tenant environment so you can understand the current state. Here are some factors to consider during the review process.

- **Domains**: Which domains, if any, exist? Does the name and description of each domain clearly indicate its purpose?
- **Permissions**: Who are the domain administrators for each domain? Who are the domain contributors for each domain? Do all assigned administrators and contributors align with the intended purpose of the domain?
- **Workspaces assigned**: Which workspaces are assigned to each domain? Do all assigned workspaces align with the intended purpose of the domain?
- **Delegated tenant settings**: Which tenant settings have been delegated to a domain administrator (or a capacity administrator) so they can override the default setting?

Step 2: Decide on domains

After you've reviewed your domains, it's time to examine the decision-making process.

For existing workspaces, consider how they can be grouped together to form a single management boundary. For more information and ways that you can organize related workspaces, see Workspace domains.

Here are some factors to consider during the decision-making process.

- **What governance decisions already exist?** When possible, refer to existing decisions. Aim for workspace and domain setup that's consistent and efficient.
- **How decentralized is the ownership and management of content?** Consider how content ownership and management currently happens throughout the organization. Sometimes decentralized approaches are referred to as a *distributed*, *federated*, or as a *data mesh* architecture. Factor in that information when analyzing needs for organizing workspaces into domains.
- **What domain groupings will work well?** There are many ways to group workspaces into a single management boundary. For example, you might choose to organize domains by subject area, team, business unit, or project. Keep in mind

that a workspace can only be assigned to one domain. For more information, see Workspace domains.

- **Who should be permitted to administer each domain?** Ideally, domain administrators are users who directly own and manage the content for the domain. Also, domain administrators should be familiar with internal, regional, and governmental regulations for the subject area. They should also be familiar with all internal governance and security requirements.
- **Who will be allowed to assign workspaces to a domain?** The domain contributor(s) role includes users who are permitted to assign a workspace to a domain. Domain contributors must also be a workspace administrator to assign the workspace to the domain. Consider how much control you want to delegate to self-service users in your organization.
- **Will different domains be handled differently?** In larger organizations, some domains might have different policies. Be prepared to adapt your decisions based on the needs and skills of different teams. For more information, see Override tenant-level settings.

Step 3: Update domains

At this point, your existing domain settings and purposeful decisions are available. You're now ready to add, change, or remove domains (if necessary).

Be sure to follow existing change management practices, and to inform all users who will be impacted by any changes.

Step 4: Document domains

Depending on how many domains you have, you might create documentation to augment the information that's available in the admin portal. Your documentation could include:

- More context or detail, such as the purpose for the domain, and why a separate domain is useful.
- Who approved the domain, and when.
- Who the domain owner or steward is—if it's different from the domain administrator(s) set in the admin portal.
- Any other compliance or regulatory requirements related to the domain.

Step 5: Manage domains

Domains should be reviewed in the admin portal regularly. Quarterly, or twice a year, should work well for this purpose.

Also, consider how you'll manage requests from users who want to create a new domain, change an existing domain, or add workspaces to a domain.

Step 6: Audit domains

You should have a process to regularly audit domains and their settings. Here are some actions to identify when you audit domains by using the activity log.

- **A new domain has been created**: Look for the *InsertDataDomainAsAdmin* activity.
- **An existing domain has changed**: Look for the *UpdateDataDomainAsAdmin* activity.
- **A workspace has been assigned to a domain**: Look for the *UpdateDataDomainFoldersRelationsAsAdmin* activity.
- **The domain administrators or domain contributors have changed**: Look for the *UpdateDataDomainAccessAsAdmin* activity.

Checklist - When planning and managing domains, key decisions and actions include:

- ✓ **Review current domains**: Determine the current state by reviewing all domains in the admin portal.
- ✓ **Discuss and decide**: Determine what domain groupings will work best to meet your needs. Involve the relevant decision makers and stakeholders when considering how to manage domains.
- ✓ **Verify whether approval is required**: Determine whether a process should exist to obtain approval from others when creating a new domain.
- ✓ **Create a schedule to re-review**: Set up a schedule to re-review domains on a regular basis.
- ✓ **Make updates**: Update domains when new needs arise.
- ✓ **Create documentation**: If you need to record other information, create documentation for your domains.
- ✓ **Create a process to handle user requests**: Set up a process for how users can request a new domain.
- ✓ **Set up auditing**: Create an auditing process so you can track when new domains are created, or when changes occur.

Govern workspaces

Workspaces are a fundamental concept in Fabric. Workspaces serve as containers for storing and securing content. Primarily, workspaces are designed for content creation, collaboration, and content distribution.

The workspaces page in the admin portal allows Fabric administrators to review and manage all workspaces in the tenant.

Here are several reasons why a Fabric administrator might get involved with managing workspaces:

- **Get access to a standard workspace**: A Fabric administrator might need to assist with a situation (such as a data refresh failure) when the primary owner of the content is, for example, away on vacation. In that case, they'll need to assign themselves to a workspace role.
- **Get temporary access to a personal workspace**: It's possible for a Fabric administrator to get access to the personal workspace of another user, but only for 24 hours. Temporary access to a personal workspace is helpful when the workspace owner has left the organization or is away on vacation.
- **Manage workspace roles**: A Fabric administrator has permission to manage workspace roles for all workspaces in the tenant. That's helpful when a centralized team manages workspace access. It's also helpful when the workspace is in an orphaned state (indicating there's no workspace administrator) which can happen as a result of employment terminations or transfers.
- **Reassign a workspace**: To unlock other features, sometimes the workspace license mode for a workspace needs to be updated. For example, a Fabric administrator can change a workspace from **Pro** or **Premium per user (PPU)** to a capacity.
- **Determine the type of workspace**: A Fabric administrator can review the SKU tier that a workspace is assigned to. For example, the administrator can quickly determine that there are four workspaces in the tenant currently assigned to PPU.
- **Locate and/or recover deleted workspaces**: The workspace state can indicate that a workspace has been deleted. For a brief period, a Fabric administrator can restore a workspace if it was deleted in error. Or, they can restore a deleted personal workspace as a standard workspace. For more information, see Workspace retention.
- **Update workspace name**: A Fabric administrator can rename a workspace, perhaps because its name doesn't conform to the established naming convention.

> ⓘ **Note**

> The level of involvement in managing workspaces depends on the roles and responsibilities assigned to a Fabric administrator. Your strategy for **content ownership and management** can also influence the extent to which a Fabric administrator gets involved with workspace management.

Step 1: Review workspaces

The first step is to review the existing workspaces and tenant environment so you can understand the current state. Here are some factors to consider during the review process.

- **Current workspaces**: Review all workspaces that currently exist. You should also review the role assignments and settings of each workspace to ensure they're appropriate.
- **Current tenant setting**: Review the current setup of the *Create workspaces* tenant setting.

There are two ways you can compile a list of current workspaces.

- View the list of workspaces in the admin portal. The list can be exported to a CSV file.
- Programmatically extract a tenant inventory by using an admin API by using:
  - The *metadata scanning* APIs (also known as the *scanner APIs*). This set of APIs allows you to locate changed workspaces asynchronously. Because it's able to extract incrementally changed workspaces, it's best suited to larger organizations with a large number of workspaces. For more information, see Metadata scanning APIs.
  - The *Get Groups As Admin* REST API or the *Get-PowerBIWorkspace* PowerShell cmdlet. These methods return data for all workspaces in the tenant, so they're best suited to smaller organizations with lower data volumes. For more information, see Groups API or workspaces cmdlet.

Step 2: Decide on workspaces

After you've reviewed your workspaces, it's time to examine the decision-making process.

Here are some factors to consider during the decision-making process.

- **What governance decisions already exist?** When possible, refer to existing workspace governance decisions. Aim for the workspace setup to be consistent and efficient.

- **Who should be allowed to create workspaces?** Consider that workspace creation permissions are a data culture and a governance decision.
- **Should there be a specific process for creating workspaces?** It's important to establish a workspace creation process that's simple and convenient for users to follow.
- **How should workspaces be named?** Determine whether a workspace naming convention would benefit the organization.
- **How are workspaces organized?** It's often useful to organize workspaces by subject and scope. The approach for organizing content into workspaces might vary between departments.
- **How much content is contained in personal workspaces?** Consider what the appropriate use of personal workspaces is within your organization.
- **Who should have access to workspaces?** You should primarily limit workspace access to users who create and manage content. For more information, see Content creator security planning.

For further considerations, see Tenant-level workspace planning and Workspace-level workspace planning.

Step 3: Update workspaces

At this point, your existing workspaces and purposeful decisions are available. If you found workspaces to be renamed or reorganized, you're now ready to make any necessary adjustments.

Considerations during the update process include:

- **Who will handle the update?** If you have several Fabric administrators, determine whether workspaces are managed by one or two specific administrators. Ensure the process is consistent, well understood, and well controlled.
- **Is there a change management process?** Consider how to avoid discrepancies between decisions, documentation, and the resulting updates. Communication between teams is a key success factor in change management. Depending on your auditing requirements, you might choose to maintain an internal change log to track who made what change, when, and why (to record more detail beyond what's captured in the activity log).
- **How will communication with users be handled?** When making a change that affects what users are able to do, be sure to communicate the change. Always aim to avoid user frustration and unnecessary support requests.

Step 4: Document workspaces

Depending on how many workspaces you have, you might create documentation to augment the information that's available from the admin portal or the REST APIs. This type of documentation is a key part of a tenant inventory. Your documentation could include:

- More context or detail, such as the intended purpose for the workspace (if it's not already mentioned in the workspace description).
- Who approved the workspace, and when.
- Who's assigned as the workspace owner, if the owner is different from the workspace administrator(s).
- Compliance or regulatory requirements related to the content stored in the workspace.
- Whether the workspace contains especially sensitive information.
- Governance requirements for the workspace.
- Lifecycle management processes applicable to the workspace.

Step 5: Manage workspaces

Managing workspaces on an ongoing basis primarily involves:

- Creating new workspaces.
- Updating workspace metadata (such as name or description).
- Updating workspace roles.
- Recovering deleted workspaces.
- Reassigning a workspace (for example, from Pro to PPU).
- Managing the *Create workspaces* tenant setting.

Depending on roles and responsibilities, secondary administrator activities might include:

- Publishing content (such as a semantic model or report) to a workspace.
- Troubleshooting issues related to existing workspace content.

> **ⓘ Important**
>
> The Fabric administrator role is a high-privilege role that can perform many high-level functions. Notably, the role doesn't automatically grant access to all data in the tenant. However, because administrators have rights to manage workspaces in the tenant, they can grant access (via workspace roles) to other users including themselves.

Step 6: Audit workspaces

You should have a process to regularly audit workspaces. Here are some actions to identify when you audit workspaces by using the activity log.

- **The *Create workspaces* tenant setting has changed**: Look for changed tenant setting values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. The item name will be *CreateAppWorkspaces*.
- **A Fabric administrator has obtained access to a user's personal workspace**: Look for the *AddAdminPersonalWorkspaceAccess* activity. The workspace name will be in the format of *PersonalWorkspace-NameOfUser*. No activity is logged when the system automatically revokes access, which happens after 24 hours.
- **A new workspace has been created**: Look for the *CreateFolder* activity.
- **An existing workspace has changed**: Look for the *UpdateFolder* activity.
- **Access for a workspace has changed**: Look for the *UpdateWorkspaceAccess* activity or the *UpdateFolderAccess* activity.
- **A workspace has been reassigned**: Look for the *MigrateWorkspaceIntoCapacity* activity.
- **A workspace has been assigned to a domain**: Look for the *UpdateDataDomainFoldersRelationsAsAdmin* activity.

> **Tip**
>
> In addition to using the activity log, we recommend that you regularly create a tenant inventory. It's a snapshot, as of a point in time, which describes all workspaces and their content (such as semantic models and reports). It can also capture details about workspace access. For more information, see **Tenant inventory** and **Access to tenant inventory data**.

Checklist - When planning and managing workspaces, key decisions and actions include:

- ✓ **Review current workspaces**: Determine the current state by reviewing all workspaces and related tenant setting in the admin portal.
- ✓ **Discuss and decide**: Determine how to govern and manage workspaces. Involve the relevant decision makers and stakeholders when deciding who's permitted to manage workspaces.

- ✓ **Verify whether approval is required**: Determine whether a process should exist to obtain approval from others when creating a new workspace.
- ✓ **Create a schedule to re-review**: Set up a schedule to re-review workspaces on a regular basis.
- ✓ **Make updates**: Update workspaces, including role assignments, when new needs arise.
- ✓ **Create documentation**: If you need to track supplementary information, create documentation for your workspaces.
- ✓ **Create a process to handle user requests**: Set up a process for how users can request a new workspace.
- ✓ **Set up auditing**: Create an auditing process so you can track when new workspaces are created, or when changes occur.

Govern embed codes

When you use the Publish to web feature, Power BI generates an *embed code*. An embed code is used to embed a report in a web page that's available on the internet to anyone. This capability is available for specific purposes: It's intended primarily for data journalism or reports containing public data that can be viewed by an anonymous audience, without requiring authentication.

Regularly reviewing and managing embed codes is a key responsibility of the Fabric administrator. It's an especially critical responsibility because it involves verifying reports that have been published publicly on the internet.

> ⊗ **Caution**
>
> Some administrators mistakenly believe that an internal application or intranet site is a safe location to embed a Publish to web report. We strongly discourage using this technique in this way because reports published via *Publish to web*—regardless of where you embed them—can be discovered with a search engine. The appropriate practice to embed Power BI content for internal audiences is to use the **API embedding** functionality, or to use the **no-code embed** technique. For more information, see the **embed for your organization** usage scenario.

Step 1: Review embed codes

The first step is to review the existing embed codes and tenant environment so you can understand the current state. Here are some factors to consider during the review process.

- **Current embed codes**: Review each embed code, across all workspaces, in the admin portal. Make note of each embed code status (such as active or blocked).
- **Current tenant setting**: Review the current setup of the *Publish to Web* tenant setting.

Step 2: Decide on embed codes

After you've reviewed your embed codes, it's time to examine the decision-making process. Involve relevant decision makers and stakeholders in discussions about what content, if any, can be published to the web.

Also determine which users are allowed to publish reports to the web. When a governance policy or security policy exists, refer to it whenever possible.

> **ⓘ Important**
>
> We strongly recommend that you enable the *Publish to Web* tenant setting for a very limited set of users. Due to the high risk of accidentally publishing reports that contain sensitive data, consider disallowing or restricting content creators from publishing to web.

Step 3: Update embed codes

At this point, your existing embed codes and purposeful decisions are available. You're now ready to make temporary or permanent changes to the existing embed codes.

To determine what updates are necessary, you might need to do some further research.

- Review all reports that have active embed codes to confirm that there's no inappropriate information being published to the web. Also verify that underlying semantic models don't contain confidential or proprietary information.
- Contact the user who published the report to clarify its purpose.
- Work with the content owner to relocate the content, if necessary, to a non-personal workspace that's well suited to the purpose. Consider using a workspace that clearly indicates it contains publicly available content. For example, a *Finance Reporting [Public]* workspace name clearly indicates its purpose.
- Review the sensitivity label that's assigned to the content. Verify that the sensitivity label indicates that the target audience is a public audience.

Step 4: Document embed codes

Depending on your circumstances, you might create some documentation to complement the information that's available in the admin portal. Your documentation could include:

- More context and detail, such as purpose, intended audience, and justification.
- Who approved the content to be published publicly, and when.
- Who the content owner is—if it's different from the user who published it.

Step 5: Manage embed codes

Embed codes should be monitored in the admin portal regularly. Also, consider how you'll manage requests from users who want to publish their reports to the web.

> **Note**
>
> Not all reports are supported for use with Publish to web. It's possible that users will have support questions related to using the feature.

Step 6: Audit embed codes

It's important to have a process to audit embed codes regularly. Here are some actions to identify when you audit embed codes by using the activity log.

- **A new embed code has been created**: Look for the *PublishToWebReport* activity.
- **The *Publish to Web* tenant setting has changed**: Look for changed tenant setting values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. The item name will be *PublishToWeb*.

Checklist - When planning and managing embed codes, key decisions and actions include:

- ✓ **Review current embed codes**: Determine the current state by reviewing all embed codes in the admin portal.
- ✓ **Review current tenant setting**: Review the setup of the *Publish to Web* tenant setting.
- ✓ **Discuss and decide**: Determine what content, if any, can be published publicly and by which users. Involve the relevant decision makers and stakeholders when

appropriate. Refer to existing governance policies when possible.

- ✓ **Verify whether approval is required**: Determine whether a process should exist to obtain approval from others when publishing a report to the web.
- ✓ **Create a schedule to re-review**: Set up a schedule to re-review embed codes on a regular basis.
- ✓ **Make updates**: Update the current embed codes as needed. Update the *Publish to Web* tenant setting based on the decisions made (if they differ from what's currently published).
- ✓ **Create documentation**: If you need to track supplementary information, create documentation of your embed codes.
- ✓ **Create a process to handle user requests**: Set up a process for how users can request publishing their report to the web, or having permission to publish their own reports to the web.
- ✓ **Set up auditing**: Create an auditing process so you can track when a new embed code is created, and when the *Publish to Web* tenant setting has changed.

Govern organizational visuals

Power BI report creators can use several types of visuals in their Power BI report design.

- **Core visuals**: The default, out-of-the-box visuals, which are built into Power BI Desktop and the Power BI service.
- **Non-certified custom visuals from AppSource**: Custom visuals that are developed by third-party software vendors or members of the worldwide Power BI community.
- **Certified custom visuals from AppSource**: Custom visuals that are developed by third-party software vendors or members of the worldwide Power BI community, and that have passed a certification process defined by Microsoft.

An organization might choose to limit the use of custom visuals (when the report is published to the Power BI service) by registering which specific visuals (and versions) are allowed. Allowed visuals are known as *organizational visuals*.

Fabric administrators are responsible for registering and managing organizational visuals in the admin center. They can register:

- Both certified and non-certified visuals from AppSource. Registering a specific visual is especially useful when the block uncertified (or all) visuals tenant setting is enabled, yet a specific visual has been validated and approved for use in your organization.
- Proprietary visuals developed internally.
- Custom visuals purchased directly from a vendor.

There are many advantages to registering organizational visuals.

- Some, or all, custom visuals will be automatically available in the visualizations pane for all report creators.
- Content creators don't have to import Power BI visual (.pbiviz) files.
- The version of the custom visuals is consistent for all reports.
- Reports and dashboards will be automatically updated when an organizational visual is updated.
- New and changed custom visuals can be methodically tested and pre-approved before becoming widely available to the organization.
- If an existing custom visual no longer meets the organization's requirements, it can be quickly disabled or deleted.

For more considerations about distributing custom visuals on user devices (for use in Power BI Desktop), see Custom visuals.

The remainder of this section focuses on managing organizational visuals.

Step 1: Review organizational visuals

The first step is to review the existing organizational visuals and tenant environment so you can understand the current state.

- **Current organizational visuals**: Review what custom visuals have been added to the organizational visuals repository.
- **Current tenant setting**: Review the setup of *Power BI visuals* tenant setting.

Step 2: Decide on organizational visuals

After you've reviewed your organizational visuals, it's time to examine the decision-making process. You should be prepared to make carefully considered decisions regarding how to govern custom visuals.

Here are some questions to consider during the decision-making process.

- **Are custom visuals permitted by the organization?** There are several reasons why an organization might choose to be intentional when relying on custom visuals.
  - The expectations for quality and stability will vary based on who developed the custom visual.
  - Freely available custom visuals might not have technical support.
  - For organizations with significant data privacy concerns—or that are extremely sensitive to data leakage concerns—the use of custom visuals might not be compatible with their risk profile. That's because custom visuals have access to

data queried from semantic models. Also, it's possible that the custom visual could transmit data back to a web service (which is often for legitimate purposes, such as calling an API or running an AI algorithm).

- **How is a custom visual validated and approved for use?** All custom visuals should be tested and pre-approved for use in the organization. This validation process reduces the risk that untrustworthy visuals are used. It also allows the administrator to specify which version has been tested and approved for use.
- **Who is permitted to use custom visuals?** The *Allow visuals created using the Power BI SDK* tenant setting controls who's permitted to add, share, or interact with a custom visual. If the organization has made decisions to restrict or limit this functionality (from AppSource or imported .pbiviz files), then you might depend on organizational visuals as the way to allow specific custom visuals.
- **Are certified visuals required?** If AppSource is permitted, some organizations prefer to restrict it to certified visuals only. That's done by setting up the *Add and use certified visuals only* tenant setting. In this situation, you can use organizational visuals to distribute an uncertified visual that's been approved for use by the organization.
- **Should custom visuals be centrally managed?** When visuals are downloaded from AppSource by individual report creators, there can be issues with mismatched versions. By using the organizational visual repository to centrally manage custom visuals, it makes the process simpler for report creators because it allows all Power BI report creators within the tenant to use the same approved version. However, it does require the Fabric administrator to get involved, which could lead to delays.
- **What sources are allowed?** Organizational visuals can come from either AppSource or a .pbiviz file. AppSource is typically the best source, especially when you want to use a certified visual. A .pbiviz file is appropriate when the visual has been obtained from a vendor privately, or when it has been developed internally.
- **When should the custom visual appear in the visualizations pane?** In many cases, it's appropriate to allow the custom visual to appear in the visualizations pane so that it's automatically available to all report creators.

Step 3: Update organizational visuals

At this point, your existing organizational visuals and purposeful decisions are available. You're now ready to make temporary or permanent changes to the existing organizational visuals.

You might also need to modify the tenant settings related to custom visuals (if report creators are permitted to download and install custom visuals that aren't in the organizational visuals repository).

> **Note**
>
> The tenant settings related to custom visuals only apply to published reports, not to reports in Power BI Desktop. To ensure that report creators have consistent custom visual options in both the Power BI service and Power BI Desktop, you'll need to manage local machine customs visuals (for Power BI Desktop) with **Group Policy**. For more information, see **User tools and devices**.

Step 4: Document organizational visuals

Depending on your situation, you might create some documentation to complement the information that's available in the admin portal. Your documentation could include:

- More context and detail, such as what the custom visual accomplishes.
- Who created the custom visual (such as an internal developer or a vendor), or who to contact for more information.
- Tests that were performed to validate the visual, so that they can be repeated should the visual be updated.
- Who approved the use of the custom visual, and when.

Step 5: Manage organizational visuals

Organizational visuals should be monitored in the admin portal regularly. Also, consider how you'll manage requests from users who want to use a new custom visual they find online.

Occasionally, you should also review when each custom visual was last updated. Investigate whether a newer version is available. When a newer version is available, you can update the organizational visual, provided it passes testing.

Step 6: Audit organizational visuals

It's important to have a process to audit organizational visuals regularly. Here are some actions to identify when you're auditing organizational visuals by using the activity log.

- **A new organizational visual has been added**: Look for the *InsertOrganizationalGalleryItem* activity.
- **An existing organizational visual has been updated**: Look for the *UpdateOrganizationalGalleryItem* activity.

- **The *Allow visuals created using the Power BI SDK* tenant setting has changed**: Look for changed tenant setting values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. The item name will be *CustomVisualsTenant*.
- **The *Add and use certified visuals only (block uncertified)* tenant setting has changed**: Look for changed tenant setting values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. The item name will be *CertifiedCustomVisualsTenant*.

Checklist – When planning and managing organizational visuals, key decisions and actions include:

- ✓ **Review current organizational visuals**: Determine the current state by reviewing all organizational visuals in the admin portal.
- ✓ **Review current tenant settings**: Review each of the *Power BI visuals* tenant settings. Determine how they may influence your reliance on organizational visuals.
- ✓ **Discuss and decide**: Determine how custom visuals should be used in the organization, and by whom. Involve the relevant decision makers and stakeholders when considering how to use organizational visuals, AppSource, and .pbiviz files.
- ✓ **Create a schedule to re-review**: Set up a schedule to re-review organizational visuals on a regular basis.
- ✓ **Make updates**: Update the current organizational visuals as needed. Update the *Power BI visuals* tenant settings based on decisions made (if they differ from what's currently published).
- ✓ **Manage user machines**: Set up Group Policy to ensure that custom visuals are managed the same way in Power BI Desktop as they are in the Power BI service.
- ✓ **Create documentation**: If you need to track supplementary information, create documentation of your organizational visuals.
- ✓ **Create a process to handle user requests**: Set up a process for how users can request the use of custom visuals (in general) or to request access to a specific custom visual.
- ✓ **Set up auditing**: Create an auditing process so you can track when a new custom visual is registered as an organizational visual, and when any of the *Power BI visuals* tenant settings have changed.

Govern Azure connections

Power BI can integrate with Azure services to extend features and provide other functionality. There are three main reasons to use Azure connections.

- **Data storage for dataflows (Gen1).** You can access the data for Power BI dataflows (Gen1) directly in Azure. A workspace can be connected to a storage account that's defined at the tenant level, or to a storage account that's specific to the workspace. This technique is sometimes referred to as *bring-your-own-lake (BYOL)*. The flexibility of a BYOL strategy is helpful when you want to reuse dataflows data beyond Power BI by allowing other processes, or other users, to view or access the data. For more information, see Configure dataflow storage to use Azure Data Lake Storage Gen2 and the self-service data preparation usage scenario.

- **Semantic model backup and restore.** You might need to back up and restore a semantic model for disaster recovery purposes, to meet data retention requirements, or to migrate a data model. For more information, see Backup and restore semantic models with Power BI Premium.

- **Azure Log Analytics integration.** You can analyze semantic model activity, performance, and trends. Log Analytics integration allows you to review diagnostic data that's generated by the Analysis Services engine (which hosts Power BI semantic models). For more information, see Semantic model event logs.

  > ⓘ **Note**
  >
  > The dataset name change has been rolled out in the Power BI service and in documentation, though there might be some instances—like with event log operation names—where the change hasn't occurred yet.

If your main use case for using Azure connections is for data storage (BYOL described in the first point above), we recommend that you consider using dataflows Gen2 and OneLake instead. Although both make use of ADLS Gen2 for data storage, they offer different capabilities, have slightly different purposes, and use different storage options (depending on how the data is written). For example: OneLake stores tabular data and dataflows Gen2 data in the open Delta Parquet format whereas the output from Power BI dataflows (Gen1) (with Azure connections) stores data in the common data model format. For more information, see Getting from dataflow generation 1 to generation 2.

The remainder of this section focuses on governing Azure connections in the admin portal.

Step 1: Review Azure connections

The first step is to review the existing Azure connections and tenant environment so you can understand the current state. There are two areas to review.

First, review the existing settings in the admin portal.

- **Current tenant-level storage setting**: Review how tenant-level storage is currently set up. It provides a default Azure connection that workspace administrators can choose to connect to within their workspace settings.
- **Current workspace-level storage permissions**: Review whether workspace-level storage permissions is enabled. When enabled, workspace administrators are permitted to connect the workspace to their own ADLS Gen2 account.

Second, review the setup of the *Azure Log Analytics connections for workspace administrators* tenant setting. When enabled, it allows workspace administrators to connect a workspace to an ADLS Gen2 account for the purpose of sending diagnostic data for semantic models.

Step 2: Decide on Azure connections

After you've reviewed your Azure connections, it's time to examine the decision-making process.

Here are some questions to consider during the decision-making process.

- **Does the usage of Azure connections fit with your data strategy and user needs?** Consider whether Azure connections would be useful for the storage of dataflows (Gen1). Determine whether you have requirements to use the semantic model backup and restore functionality. Consider whether there are needs for Azure Log Analytics integration.
- **What data storage is centralized versus decentralized?** Understand the needs of your decentralized teams, and whether individuals or departments currently maintain their own Azure Storage accounts. Determine whether workspace administrators will be permitted to connect their own ADLS Gen2 account, or whether you prefer to use one ADLS Gen2 account for all workspaces (tenant-level storage).
- **How will OneLake be used versus Azure connections?** With the introduction of OneLake, consider whether you might choose to gradually move to using OneLake for data storage (BYOL).

For more information, see Workspace integration with ADLS Gen2.

For more information, see Workspace integration with Azure Log Analytics.

Step 3: Update Azure connections

At this point, your existing Azure connections are available, and you've made purposeful decisions about whether you intend to integrate a data lake with Power BI. You're now ready to adjust the settings, if necessary, based on your findings.

Step 4: Document Azure connections

Depending on your situation, you should create some documentation to complement the information that's available in the admin portal. Your documentation might include:

- The tenant-level data lake location that's approved for usage. Include who owns and manages the data lake, and who to contact for more information.
- Whether workspace-level data lakes can be integrated. Other information, such as key decisions made and reasons why, should be documented for future reference.

Step 5: Manage Azure connections

Azure connections should be monitored in the admin portal from time to time.

Consider how you'll support multiple ADLS Gen2 accounts in the organization (if workspace-level Azure connections are permitted).

Also, consider how you'll manage requests from users who want to connect a workspace to Azure Log Analytics.

Step 6: Audit Azure connections

It's important to have a process to audit Azure connections regularly. There are several actions to identify when you're auditing Azure connections by using the activity log.

- **A workspace has been connected to ADLS Gen2**: Look for the *AddLinkToExternalResource* activity. The *ResourceType* will indicate whether it's a storage account or Log Analytics.
- **A workspace has been disconnected from ADLS Gen2**: Look for the *DeleteLinkToExternalResource* activity. The *ResourceType* will indicate whether it's a storage account or Log Analytics.
- **Tenant-level storage is enabled or disabled**: Look for changed values in the activity log by using the *AddExternalResource* or the *DeleteLinkToExternalResource* activity.
- **Workspace-level storage is enabled or disabled**: Look for changed values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. The item name will

be *storageAccountAttachForWorkspaceAdminsEnabled*. The *SwitchState* will be either true or false.

- **The *Azure Log Analytics connections for workspace administrators* tenant setting has changed**: This tenant setting allows some or all workspace administrators to integrate their own ADLS Gen2 account. Look for changed tenant setting values in the activity log by using the *UpdatedAdminFeatureSwitch* activity. The item name will be *LogAnalyticsAttachForWorkspaceAdmins*.

Checklist - When planning and managing Azure connections, key decisions and actions include:

- ✓ **Review current Azure connections**: Determine the current state by reviewing the tenant-level and workspace-level settings for Azure connections in the admin portal. Also review the setup of *Azure Log Analytics connections for workspace administrators* tenant setting.
- ✓ **Discuss and decide**: Determine whether you intend to integrate Azure connections with Power BI. Going forward, decide what your optimal use is for OneLake versus Azure connections for data storage (BYOL).
- ✓ **Verify whether approval is required**: Determine whether a process should exist to obtain approvals for using workspace-level storage accounts.
- ✓ **Create a schedule to re-review**: Set up a schedule to re-review Azure connections on a regular basis.
- ✓ **Make updates**: Update the current Azure connections as needed to modify tenant-level and workspace-level storage permissions. Also update the *Azure Log Analytics connections for workspace administrators* tenant setting based on the decisions made (if they differ from what's currently set).
- ✓ **Create documentation**: If you need to track supplementary information, create documentation of your Azure connections.
- ✓ **Create a process to handle user requests**: Set up a process for how users can request to be able to use Azure connections.
- ✓ **Set up auditing**: Create an auditing process so you can monitor when workspaces have set an Azure connection, or when settings have changed.

Audit usage of Power BI

Tenant-level auditing data allows you to analyze adoption efforts, understand usage patterns, educate users, support users, mitigate risk, improve compliance, manage

license costs, and monitor performance. It's critical that you extract and store Power BI auditing data as early as possible, even if today you're not yet ready to analyze the data.

For information about auditing users, activities, and solutions in Power BI, see Tenant-level auditing.

Monitor the Power BI service

Monitoring refers to ongoing activities that inform you about what's going on. Monitoring is usually a passive activity that involves alerting and automation, though sometimes it's done actively.

For information about monitoring Power BI, see Tenant-level monitoring.

Related content

For more considerations, actions, decision-making criteria, and guidance to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Workspaces

05/30/2025

This overview describes a series of articles that help you plan your implementation of Microsoft Fabric workspaces. The articles focus on the Power BI experience inside Fabric.

> **Note**
>
> This article is part of the *Power BI implementation planning* series of articles. The series focuses on planning to implement a Power BI experience inside Microsoft Fabric. See the series introduction.

As described in each article, the article audience might include:

- **Fabric administrators**: Administrators who are responsible for overseeing the Fabric implementation in the organization.
- **Center of Excellence (CoE), IT, and business intelligence (BI) teams**: Teams that are responsible for overseeing the use of data and BI in the organization, and for supporting self-service users throughout the organization.
- **Content creators and owners**: Self-service users who create, publish, and manage content in workspaces.

Comprehensive workspace planning is an integral part of a successful Fabric workspace implementation. Gaps in workspace planning can reduce user flexibility and lead to user workarounds in organizing and securing content.

A *workspace* is a container in the Fabric portal where users store and secure content. Workspaces are primarily designed for content creation and collaboration.

> **Note**
>
> The concept of a workspace originated in Power BI. In Fabric, the purpose of a workspace broadens. A Fabric workspace can contain items from more than one Fabric experience (also called a *workload*). Although the scope of content in a Fabric workspace is broader than in Power BI, you can apply most of the implementation planning in these articles to planning your Fabric workspaces.

Articles

The workspace planning content is organized in the following articles:

- **Tenant-level workspace planning**: This article describes strategic decisions and actions you take that affect all workspaces in the tenant.

  Strategic decisions cover goals and objectives that you have for your organization's implementation.

- **Workspace-level planning**: This article describes tactical decisions and actions you take for each workspace.

  Tactical decisions include specific implementation plans that support your organization's long-term implementation strategy.

Next step

Power BI implementation planning: Workspaces at the tenant level

Power BI implementation planning: Workspaces at the tenant level

05/30/2025

This article describes strategic implementation planning that you do at the *tenant level* for Microsoft Fabric workspaces, with an emphasis on the Power BI experience inside Fabric.

> **Note**
>
> This article is part of the *Power BI implementation planning* series of articles. The series focuses on planning to implement a Power BI experience inside Microsoft Fabric. See the series introduction.

The article primarily is for:

- **Fabric administrators**: Administrators who are responsible for overseeing the Fabric implementation in the organization.
- **Center of Excellence (CoE), IT, and business intelligence (BI) teams**: Teams that are responsible for overseeing the use of data and BI in the organization, and for supporting self-service users throughout the organization.

This article might be helpful for self-service creators who create, publish, and manage content in workspaces.

Because workspaces can be used in different ways, most tactical implementation decisions are made at the workspace level. You make some strategic planning decisions at the tenant level.

We recommend that you make tenant-level workspace decisions as early as possible because they affect everything else. Also, it's easier to make individual workspace decisions when you're clear about your overall workspace goals and objectives.

> **Note**
>
> The concept of a workspace originated in Power BI. In Fabric, the purpose of a workspace broadens. A Fabric workspace can contain items from more than one Fabric experience (also called a *workload*). Although the content scope is broader than in Power BI, you can apply most of the workspace planning activities described in these articles to planning your Fabric workspaces.

Workspace creation permissions

The decision of who is allowed to create workspaces in the Power BI service is a data culture and governance decision.

You have two options:

- **All (or most) users can create new workspaces**: This approach usually aligns with existing decisions for other applications. For example, when users are permitted to create their own SharePoint sites or Teams channels, it makes sense that Fabric adopts the same policy.
- **Workspace creation is limited to a select set of users**: This approach usually indicates a governance plan is in place or is planned. Managing this process can be fully centralized (for example, if only IT is permitted to create a workspace). A more flexible and practical approach is when it's a combination of centralized and decentralized individuals. In this case, specific satellite members of the CoE, champions, or trusted users are trained to create and manage workspaces on behalf of their business units.

You should set up the Create workspaces tenant setting in the Fabric admin portal according to your decision on who is allowed to create workspaces. For more information, see Govern workspaces.

Checklist of key decisions and actions when you plan who can create workspaces:

- ✓ **Determine and validate user needs**: Schedule collaborative discussions with relevant stakeholders and interested parties to learn how users currently work. The goal is to ensure that you have a clear understanding of user needs.
- ✓ **Decide who is allowed to create workspaces**: Determine whether all users, only a centralized team, or certain centralized and decentralized users can create a new workspace. Ensure that the decision purposefully aligns with your goals for data culture in the organization. Be sure to obtain approval from your executive sponsor.
- ✓ **Create a security group for who is permitted to create workspaces**: If a subset of users can create workspaces, create a security group for these users. Name the group clearly, like **Fabric Workspace Creators**. Add members who are permitted to create workspaces to this security group.
- ✓ **Update the tenant setting**: Add the new security group to the **Create workspaces** tenant setting in the admin portal. In addition to the Fabric Workspace Creators group, other

groups that might be assigned this tenant setting are CoE, support, and Fabric administrators.

Workspace naming conventions

Workspace naming conventions are an agreed-upon pattern for how workspaces are named. Usually, naming conventions are a requirement and not a suggestion.

It can be difficult to strictly enforce naming conventions when many users possess the permission to create workspaces. You can mitigate this concern with user education and training. You can also conduct an auditing process to find workspaces that don't conform to the naming conventions.

The workspace name can convey additional information about the workspace, including:

- **Purpose**: A workspace name should always include a description of its content. For example, *Sales Quarterly Bonus Tracking*.
- **Item types**: A workspace name can include a reference to the types of items it contains. For example, use *Sales Data* to indicate the workspace stores items like a lakehouse or semantic models. *Sales Analytics* could indicate that the workspace stores analytical reports and dashboards.
- **Stage (environment)**: A workspace name might include its stage. For example, it's common to have separate workspaces (development, test, and production) for lifecycle management.
- **Ownership and responsibility**: A workspace name might include an indication of who's responsible for managing the content. For example, use of an *SLS* prefix or suffix can indicate that the sales team owns and manages the content.

> **Tip**
>
> To keep workspace names short, you can include more detail in the workspace description. However, make sure that the most relevant information is included in the workspace name, particularly if you anticipate that users will search for workspaces. You can also use a workspace image to augment the workspace name. These considerations are described further in the **workspace settings** section in the next article.

Having consistent workspace names helps everyone. The user experience is improved because users can find content more easily. Also, administrators can oversee the content more easily when you use predictable naming conventions.

We recommend that you include the workspace naming conventions in your centralized portal and training materials.

The following list describes more considerations related to workspace naming.

- **Use short yet descriptive names**: The workspace name should accurately reflect its contents, with the most important part at the beginning of the name. In the Fabric portal, long workspace names can become truncated in user interfaces, requiring the user to hover the cursor over the workspace name to reveal the full name in a tooltip. Here's an example of a short yet descriptive name: *Quarterly Financials*.

- **Use a standard prefix**: A standard prefix can arrange similar workspaces together that are sorted. For example: *FIN-Quarterly Financials*.

- **Use a standard suffix**: You can add a suffix for additional information, such as when you use different workspaces for development, test, and production. We recommend appending *[Dev]* or *[Test]* suffixes but leaving production as a user-friendly name without a suffix. For example: *FIN-Quarterly Financials [Dev]*.

- **Be consistent with the Power BI app name**: The workspace name and its Power BI app can be different, particularly if it improves usability or understandability for app consumers. We recommend keeping the names similar to avoid confusion.

- **Omit unnecessary words**: The following words might be redundant, so avoid them in your workspace names:
  - The word *workspace*.
  - The words *Fabric* or *Power BI*. Many Fabric workspaces contain items from various workloads. However, you might create a workspace intended to target only a specific workload (such as Power BI, Data Factory, or Synapse Data Engineering). In that case, you might choose a short suffix so that the workspace purpose is made clear.
  - The name of the organization. However, when the primary audience is external users, including the organization's name can be helpful.

> **Note**
>
> We recommend that you notify users when a workspace name is scheduled to change. In most scenarios, it's safe to rename a workspace in the Fabric portal. The **GroupID** value, the unique identifier of a workspace, doesn't change and is part of the workspace URL. However, XMLA connections are affected because they connect by using the workspace name instead of the **GroupID** value.

Checklist of key decisions and actions when you plan workspace naming conventions:

- ✓ **Determine requirements or preferences for workspace names**: Consider how you want to name workspaces. Decide whether to create strict naming convention requirements or allow more flexible requirements guided by suggestions and examples.
- ✓ **Review existing workspace names**: Update existing workspace names as appropriate so that they're good examples for users to follow. When users see an existing workspace renamed, they interpret that as an implied standard to adopt.
- ✓ **Create documentation for workspace naming conventions**: Provide reference documentation about workspace naming convention requirements and guidelines. Be sure to include examples that show the correct use of acronyms, prefixes, and suffixes. Make the information available in your centralized portal and in training materials.

Workspace domains

Essential to the planning and implementation process is being clear about how you want content owned and managed. Clarity is especially critical when responsibilities for creating and managing data assets are decentralized among many departments or business units. Sometimes this approach is called a *distributed*, *federated*, or *data mesh* architecture.

One way to support workspace ownership and management in Fabric is by using domains. A *domain* provides a way to logically group multiple workspaces that have similar characteristics. For example, you might create a domain to group all your sales workspaces together, and create another domain to hold your finance workspaces.

Here are the key advantages of using domains:

- Domains group similar workspaces into a single management boundary.
- Domains permit certain tenant settings to be managed at the domain level. For more information, see Override tenant-level settings.
- Domains help users find relevant data. For example, they can use filters in the OneLake catalog.

The following table lists different ways you can organize related workspaces:

Expand table

Method for organizing workspaces	Example domain
By subject area, domain, or content type	The *Finance* domain includes all workspaces related to finance content.
By the team or department that owns and manages the content	The *Enterprise BI* domain includes all workspaces that the team is directly responsible for managing.
By organizational business unit or segment	The *European division* domain includes all workspaces that are directly related to operations in Europe.
By project	The *Subsidiary acquisition* domain includes all workspaces that support a highly sensitive project.

Here are some questions to consider when you plan for Fabric domains in your tenant:

- **How does each workspace map to a domain?** Each workspace can be assigned to only one domain, so be prepared to do some detailed planning. Consider creating a matrix diagram that lists workspaces in the rows and domains in the columns to help you plan how they're assigned. If you discover that you need to reorganize workspaces, you can reassign the domain in the workspace settings or in the admin portal.
- **Who is authorized to manage a domain?** Users assigned the Domain Administrator role are authorized to manage an existing domain. When possible, assign domain administrators who directly own and manage the content for the domain. Domain administrators should be experts who are familiar with internal, regional, and governmental regulations for the subject area. They should also be familiar with all internal governance and security requirements. For more information, see Domain roles.
- **Who is allowed to assign workspaces to a domain?** Users assigned the Domain Contributor role define which workspace administrators can assign a workspace to a domain. If you allow more users to assign workspaces to a domain, you should frequently audit the accuracy of the assigned groupings. If you allow only specific groups of users or Fabric admins and domain admins, you have more control over how domains are assigned. For more information, see Domain roles.
- **Do you have specific compliance needs or restrictions, such as geographic area?** Keep in mind that the geographic area for data storage is set for each capacity rather than for the domain. Consider how assigning a workspace to a domain and to a capacity affects your planning process.

For more information, see Govern domains.

Checklist of key decisions and actions when you plan for workspace domains:

When you plan for workspace domains, key decisions and actions include:

- ✓ **Validate how content ownership works**: Ensure that you deeply understand how content ownership and management is happening throughout the organization. Factor that information into your plans to organize workspaces into domains.
- ✓ **Plan workspace domains**: Have discussions to plan how to best organize workspaces into domains. Confirm all key decisions with the CoE and your executive sponsor.
- ✓ **Educate Fabric administrators**: Ensure that your tenant administrators are familiar with how to create a domain, and how to assign and manage domain administrators.
- ✓ **Educate domain administrators**: Ensure that your domain administrators understand the expectations for this role in managing a domain.
- ✓ **Decide how to handle domain contributors**: Consider which users should have permission to assign workspaces to a domain.
- ✓ **Create an auditing process**: On a regular basis, validate that the assigned domain groupings are correct.

Workspace creation process

If you decide to limit who can create workspaces, the broader user population needs to know what the process is to request a new workspace. In this case, it's important to establish a request process that's easy and convenient for users to find and follow.

It's also critical to respond quickly to a request for a new workspace. A service-level agreement (SLA) of 2 to 4 hours is ideal. If the process to request a new workspace is too slow or burdensome, people will use the workspaces they have, so they can keep moving. If they elect to skip creating a new workspace, what they use instead could be suboptimal. They might choose to reuse an existing workspace that isn't well-suited to the new content, or they might share content from their personal workspace.

> **Tip**
>
> When you create a new process, the goal is to make it easy for people to comply with the process. Like with all data governance decisions, the point is to make it easy for users to follow documented processes.

The following table lists the information to collect in a request for a new workspace:

Expand table

Information needed	Example	Validation required
Workspace name	SLS-Field Sales Analytics	• Does the name follow naming conventions? • Does another workspace with the same name exist?
Stages needed	SLS-Field Sales Analytics [Dev], SLS-Field Sales Analytics [Test], and SLS-Field Sales Analytics	• Are multiple workspaces necessary to properly support the content? • If so, should a deployment pipeline be created too?
Description	Customer sales and order history for monthly, quarterly, and yearly analysis.	• Is there an expectation that sensitive data or regulated data will be stored? • If so, will that affect how the workspace is governed?
Target audience	Global field sales organization	• How broad is the content delivery scope? • How will that scope affect how the workspace is governed?
License mode assigned to the workspace	A Fabric capacity for the sales team is needed because a large number of the salespeople are viewers only and they have a free license	• What level of Fabric capacity is required?
Data storage requirements	Data residency in Canada	• Do data residency needs require Multi-Geo? • What quantity of data do you expect to process and store?
Workspace administrators	FabricContentAdmins-FieldSalesAnalytics	• Is the administrator (preferably) a group? • Are at least two administrators assigned?
Person submitting the request	requestor@contoso.com	• Does the person submitting the request work in a role or line of business related to the information provided?

This table includes the minimum amount of information required to set up a workspace. However, it doesn't include all possible configurations. In most cases, a workspace administrator takes responsibility for completing the setup after the workspace is created. For more information, see Workspace-level settings.

You can choose from many technology options to create an online form for the workspace creation request. Consider using Microsoft Power Apps, a low-code software option that's ideal for building simple web-based forms and applications. The technology you choose to use for creating a web-based form depends on who is responsible for creating and maintaining the form.

> **Tip**
>
> To improve efficiency and accuracy, consider automating the process by using the **Power BI REST API** to programmatically create or update a workspace. In this case, we recommend including review and approval processes rather than automatically processing each request.

Checklist of key decisions and actions when you plan the process to request a new workspace:

- ✓ **Establish a process for requesting a new workspace**: Decide what the specific process is for requesting a new workspace. Consider the information you'll need, how to capture the information, and who processes the request.
- ✓ **Create a standard form for requesting a new workspace**: Decide what information to include on the form for a new workspace. Consider building a Power Apps app to collect the information from the user. Ensure links to the form are broadly available and easy to find in your centralized portal and other common locations. Include a link to the form in ongoing communications too.
- ✓ **Decide who will respond to submitted requests and how quickly**: Determine who'll process requests. Consider what the expected response time is for handling a request for a new workspace. Verify that you can handle requests quickly so that self-service users don't experience delays.
- ✓ **Conduct a knowledge transfer session**: If another team will support the workspace request process, conduct a knowledge transfer session with them so they have all the information they need.
- ✓ **Create documentation for how to approve or deny a request**: Create documentation about how to approve a request, targeted at those who will review or process requests.

Also include reasons why a request might be denied, and what action should be taken.

✓ **Create documentation for how to request a workspace**: Create documentation about how to request a new workspace, targeted at users who can't create their own workspaces. Include what information is required, and expectations for a response. Ensure that the information is available in your centralized portal and in training materials.

Workspace governance level

Not all workspaces need the same level of oversight. Certain workspaces might be considered *governed*. A governed workspace results in more requirements and expectations for its content. Some organizations use the term *managed* instead of governed.

Four key decision criteria determine the level of governance required:

- Who owns and manages the BI content?
- What is the scope for delivery of the BI content?
- What is the data subject area?
- Is the data or the BI solution considered critical?

> ⓘ **Note**
>
> For more information about the four key decision criteria, see the governance article in the Fabric adoption roadmap.

You might start out with two levels of workspaces: governed and ungoverned. We recommend that you keep the governance levels as simple as possible. However, depending on your specific circumstances, you might need to subdivide the governed classification. For example, critical content that the enterprise Power BI team manages might have a unique set of governance requirements. Critical content that business units own and manage might be subject to a slightly different set of requirements. In some cases, decisions are tailored to individual business units.

The following table lists some of the most common requirements when a workspace is considered governed.

Expand table

Category	Potential governance requirement
Ownership and support	• Ownership is assigned, with clear responsibilities for the technical content owner or subject matter expert. • User support team/person is assigned, and users understand how to request

Category	Potential governance requirement
	help or submit issues. • A mechanism is in place for user feedback, questions, and enhancement requests. • A communication plan exists to announce important changes to content in the workspace.
Workspace setup	• The workspace is well-organized with a well-defined purpose. • A specific naming convention is used. • The workspace is assigned to a specific domain. • Workspace description, image, and contacts are required.
Accuracy	• All content is certified. • Data validations are automated so that content owners become aware of data quality issues on a timely basis.
Distribution	• A Power BI app is used for distributing reports and dashboards.
Security and data protection	• Security groups are used (instead of individual accounts) for managing workspace roles. • Sensitivity labels are used for information protection. • Only sanctioned (or approved) data sources are permitted. • All source files reside in a secure, backed-up location.
Change management	• Separate development, test, and production workspaces are used. • Source control (such as Git integration) is used for all Power BI Desktop files and items in the Fabric portal. • Versioning or source control is used for all data source files. • Lifecycle management and change management processes, including deployment pipelines or DevOps processes, are followed.
Capacity	• The workspace is assigned to an appropriate Fabric capacity level. • The Fabric capacity is managed and monitored.
Gateway	• A data gateway in standard mode (nonpersonal) is used. • All gateway data source credentials use approved credentials.

Category	Potential governance requirement
Auditing and monitoring	• Active auditing and monitoring processes are in place for tracking adoption, usage patterns, and performance.

> **Tip**
>
> Governance requirements usually aren't optional. For this reason, timely auditing is important, and enforcement becomes necessary in certain situations. For example, if governed workspaces require all files to be in a secure location and an unapproved file location is detected during an audit, a required action is to update the file location.

Checklist of key decisions and actions when you plan workspace governance level:

- ✓ **Decide on the workspace governance levels**: Determine the levels of governance that you'll need. Keep it as simple as possible.
- ✓ **Decide on the criteria for how to classify a workspace**: Determine what the decision criteria to use to classify workspaces to a specific governance level.
- ✓ **Decide what the workspace governance requirements are**: For each governance level, determine what the specific requirements.
- ✓ **Decide how to designate the workspace governance level**: Find the simplest way to identify the governance level for a workspace. You could record it as part of its name, part of its description, or stored elsewhere (for example, a SharePoint list that contains more information about each workspace).
- ✓ **Create documentation for workspace governance requirements**: Create useful documentation targeted at content creators that includes what their responsibilities are for managing content in a governed workspace. Make the information available in your centralized portal and in training materials.
- ✓ **Create workspace auditing processes**: For workspaces that are considered governed, create an auditing process to identify areas of noncompliance with the most important requirements. Ensure that someone is responsible for contacting content owners to address compliance issues.

Next step

Power BI implementation planning: Workspaces at the workspace level

Power BI implementation planning: Workspaces at the workspace level

05/30/2025

This article describes tactical implementation planning that you do at the *workspace level* for Microsoft Fabric workspaces, with an emphasis on the Power BI experience inside Fabric.

> **Note**
>
> This article is part of the *Power BI implementation planning* series of articles. The series focuses on planning to implement a Power BI experience inside Microsoft Fabric. See the series introduction.

The article primarily is for:

- **Fabric administrators**: Administrators who are responsible for overseeing the Fabric implementation in the organization.
- **Center of Excellence (CoE), IT, and business intelligence (BI) teams**: Teams that are responsible for overseeing the use of data and BI in the organization, and for supporting self-service users throughout the organization.
- **Content creators and owners**: Self-service users who create, publish, and manage content in workspaces.

To use workspaces effectively, you make many tactical decisions. Whenever possible, individual workspace-level decisions should align with your tenant-level decisions.

> **Note**
>
> The concept of a workspace originated in Power BI. In Fabric, the purpose of a workspace broadens. A Fabric workspace can contain items from more than one Fabric experience (also called a *workload*). Although the content scope is broader than in Power BI, you can apply most of the workspace planning activities described in these articles to planning your Fabric workspaces.

Workspace purpose

When you plan a workspace, it's important to consider not only the type of content it will store but also the activities the workspace is intended to support.

Consider the following two examples of finance-related workspaces. Although they're both dedicated to the same team, each workspace serves a different purpose:

- **Financial month-end workspace**: The *Financial month-end workspace* contains reconciliation and month-end closing reports. This workspace is considered an informal workspace to support collaborative efforts. A Power BI app isn't necessary for content viewers because the primary use of this workspace is collaboration by a small group of people who work closely together. Most team members have permissions that let them edit content in this workspace.
- **Financial reporting workspace**: The *Financial reporting workspace* contains the finalized, presentation-level reports. This workspace contains content that is broadly distributed throughout the organization to many viewers, including to executives, via a Power BI app. The workspace is closely governed.

With these two examples in mind, consider two specific aspects of workspace purpose: *intent for collaboration* and *intent for viewing*.

Intent for collaboration

The primary objective of a workspace in the Fabric portal is to facilitate collaboration among multiple contributors.

Collaboration in a workspace occurs in multiple ways:

- **Team-based development**: Users can work together to build, test, and publish content. One user might work on the design of a lakehouse. Another user might work on the design of the semantic model, and other users might focus on building reports.
- **Testing and validations**: Users might need to perform data validations for new content. Subject matter experts from the business unit might need to perform user acceptance testing (UAT). A data quality team might need to validate the accuracy of the semantic model.
- **Enhancements**: Content stakeholders and consumers might suggest enhancements to content throughout the lifecycle of the content.
- **Ownership transfer**: Another person or team might assume responsibility for content that someone else created.

One of the key areas of the Fabric adoption roadmap is content ownership and management. The type of collaboration that occurs in a workspace differs based on the approach you choose for content ownership and management:

- **Business-led self-service BI**: Content that the content creators in a business unit or department own or manage. In this scenario, most collaboration in the workspace occurs among users in that business unit.

- **Managed self-service BI**: Data that a centralized team owns or manages, but various content creators from business units take responsibility for reports and dashboards. In this scenario, it's highly likely that multiple workspaces are needed to securely facilitate collaboration by multiple teams of people.
- **Enterprise BI**: Content that a centralized team, such as IT, enterprise BI, or the CoE owns or manages. In this scenario, collaboration efforts in the workspace occur among users in the centralized team.

Checklist of key decisions and actions when you plan for collaboration in a workspace:

- ✓ **Consider expectations for collaboration**: Determine how workspace collaboration needs to occur and who's involved in a single team or across organizational boundaries.
- ✓ **Consider expectations for content ownership and management**: Think about how the different content ownership and management approaches (business-led self-service BI, managed self-service BI, and enterprise BI) will influence how you design and use workspaces.

> **Tip**
>
> When your needs aren't met by taking a single approach, be prepared to be flexible and use a different **content ownership and management** strategy for different workspaces. The strategy can be based on the scenario and on the team members that are involved.

Intent for content viewing

The secondary objective for a workspace is to distribute content to consumers who need to view the content. For content viewers, the primary Fabric workload is Power BI.

You can approach content distribution in the Power BI service in multiple ways:

- **Reports can be viewed by using a Power BI app**: Content stored in a nonpersonal workspace can be published to a Power BI app. A Power BI app is a more user-friendly experience than viewing reports directly in a workspace. For this reason, using a Power BI app is often the best choice for distributing content to consumers. Audiences for a Power BI app are flexible. However, sometimes your goals for how to distribute content with an app are a factor in determining how to organize content in or across workspaces. For more information about securing Power BI apps, see Report consumer security planning.

- **Reports can be viewed directly in the workspace**: This approach is often appropriate for informal, collaborative workspaces. Workspace roles define who can view or edit the content contained in a workspace. For more information about workspace roles, see Content creator security planning.
- **Reports can be shared**: Use of per-item permissions (links or direct access) is useful when you need to provide read-only access to a single item in a workspace. We recommend that you use app permissions and workspace roles more frequently than sharing because they're easier to maintain. For more information, see Report consumer security planning.
- **Reports can be embedded in another application and viewed in the application**: Sometimes the intention is for consumers to view Power BI content embedded in another application. Embedding content is useful when it makes sense for the user to remain in the application to increase efficiency and stay within its workflow.

Another key area of the Fabric adoption roadmap is content delivery scope. The ways that a workspace support content distribution differs based on the content delivery scope:

- **Personal BI**: Content is intended for the content creator to use. Because sharing content with other users isn't an objective, personal BI is done in a personal workspace as described in the next section.
- **Team BI**: Content is shared with a relatively small number of colleagues who work closely together. In this scenario, most workspaces are informal, collaborative workspaces.
- **Departmental BI**: Content is distributed to many consumers who belong to a large department or business unit. In this scenario, the workspace is primarily for collaboration efforts. In departmental BI scenarios, content is commonly viewed in a Power BI app instead of directly viewed in the workspace.
- **Enterprise BI**: Content is delivered broadly across organizational boundaries to the largest number of target consumers. In this scenario, the workspace is primarily for collaboration efforts. For enterprise BI scenarios, content is commonly viewed in a Power BI app instead of directly viewed in the workspace.

> **Tip**
>
> When you plan your workspaces, consider the needs of the audience when determining the workspace **license mode**. The type of license assigned to the workspace affects the features that are available, including who can view or manage workspace content.

Checklist of key decisions and actions when you consider expectations for how workspace content is viewed:

- ✓ **Consider expectations for viewing content**: Determine how you expect consumers to view content that's been published to the workspace. Consider whether viewing will happen directly in the workspace directly or by using a different method.
- ✓ **Determine who the content will be delivered to**: Consider who the target audience is. Also consider the workspace license mode, especially when you expect a significant number of content viewers.
- ✓ **Evaluate needs for a Power BI app**: Consider what the workspace purpose is as it relates to the content distribution requirements. A requirement for a Power BI app can influence decisions about creating a workspace.
- ✓ **Consider expectations for content delivery scope**: Consider how the different content delivery scopes (personal BI, team BI, departmental BI, and enterprise BI) will influence how you design and use workspaces.

> **Tip**
>
> Be prepared to be flexible. You can use a different content viewing strategy for workspaces based on the scenario and on the team members that are involved. Also, don't be afraid to use different **content delivery scope** approaches for workspaces when it can be justified.

Appropriate use of personal workspaces

You can choose from two types of workspaces:

- **Personal workspaces**: Every user has a personal workspace. A personal workspace can be used for publishing certain types of content to the Fabric portal. Its primary purpose is to support personal BI usage scenarios.
- **Workspaces**: The primary purpose of a workspace is to support collaboration among multiple users. Secondarily, a workspace can also be used for viewing content.

A personal workspace used for anything other than learning personal BI, temporary content, or testing increases an organization's risk. In a personal workspace, only the workspace owner creates and manages the content. A personal workspace also doesn't support collaboration with others.

To allow a user to create any type of Fabric item, like a lakehouse or a warehouse, a workspace must be added to a Fabric capacity. The process applies to both standard workspaces and

personal workspaces. You can govern who can create certain types of items in a personal workspace by managing its capacity assignment.

A personal workspace is limited in its options to share content with others. You can't publish a Power BI app from a personal workspace. Also, Power BI apps are an important mechanism for distributing content to the organization. Per-item permissions (links or direct access) are the only way to share personal workspace content with others. Extensive use of per-item permissions involves more effort, and it increases the risk of error. For more information, see Report consumer security planning.

Checklist of key decisions and actions when you consider expectations for how personal workspaces are used:

- ✓ **Understand the current use of personal workspaces**: Have conversations with your users and review the activity activity data to ensure you understand what users are doing with their personal workspaces.
- ✓ **Decide how personal workspaces should be used**: Decide how personal workspaces should (and should not) be used in your organization. Focus on balancing risk and ease of use with needs for content collaboration and viewing.
- ✓ **Relocate personal workspace content when it's appropriate**: For critical content, move content from personal workspaces to standard workspaces when it's appropriate.
- ✓ **Create and publish documentation about personal workspaces**: Create useful documentation or FAQs for your users about how to effectively use personal workspaces. Make the information available in your centralized portal and in training materials.

For more information, see the Fabric adoption roadmap:

- Fabric centralized portal
- Fabric training
- Fabric documentation

Workspace ownership

One of the most important things to consider when you plan workspaces is determining the ownership and stewardship roles and responsibilities. The goal is to have clarity on exactly who is accountable for creating, maintaining, publishing, securing, and supporting the content in each workspace.

Clarity on ownership is especially relevant when responsibilities for creating and managing data are decentralized or distributed among departments and business units. This concept is also sometimes referred to as a *data mesh* architecture. For more information about data mesh, see What is data mesh?.

In Fabric, decentralized or distributed ownership is enabled through workspaces. Different areas of the organization can work independently but still contribute to the same underlying data structure in OneLake. Each workspace can have its own administrator, access control, and capacity assignment (for billing, geographic data location, and performance monitoring).

> **Tip**
>
> Another way to support workspace ownership in Fabric is with **domains**, which are described later in this article.

When the intent for collaboration involves decentralization and multiple teams beyond a single business unit, it can add complexity for managing workspaces. Often, it's helpful to create separate workspaces to clearly delineate which team is responsible for which content. Use of multiple workspaces allows you to be specific as to ownership and management responsibilities, and it can help you to set security according to the principle of least privilege. For more security considerations, see Content creator security planning.

> **Tip**
>
> Your decisions related to accountability and responsibility should correlate directly with your actions related to defining **workspace access**, which is described later in this article.

Checklist of key decisions and actions when you consider workspace ownership responsibilities:

- ✓ **Gain a full understanding of how content ownership works**: Ensure that you wholly understand how content ownership and management happens throughout the organization. Recognize that you likely won't have a one-size-fits-all approach to apply uniformly across the entire organization. Understand decentralized or distributed ownership needs.
- ✓ **Define and document roles and responsibilities**: Ensure that you define and document clear roles and responsibilities for people who collaborate in workspaces. Make this

information available in onboarding activities, in training materials, and in your centralized portal.

- ✓ **Create a responsibility matrix**: Map out who is expected to handle each function to create, maintain, publish, secur, and support content. Have this information ready when you start planning for workspace access roles.
- ✓ **Consider co-ownership or multi-team ownership scenarios**: Identify a scenario when it would be helpful to create separate workspaces so that responsibilities are clear.
- ✓ **Create workspace management documentation**: Educate workspace administrators and members about how to manage workspace settings and access. Include responsibilities for workspace administrators, members, and contributors. Make the information available in your centralized portal and in training materials.

Workspace organization

How to organize workspaces is one of the most important aspects of workspace planning.

Different business units and departments might use workspaces differently depending on their collaboration requirements. When you need a new workspace, we recommend that you consider the factors described in this section.

Workspace subject and scope

The following options present some suggestions about how you can organize workspaces by subject and scope.

In some cases, you might already have some useful groups established in Microsoft Entra ID. You can use the groups to manage access to resources for the defined subject area and scope. However, you might need to create new groups to fully implement this approach. For more information, see Workspace access.

Option 1: One workspace per subject area or project

If you create one workspace for each subject area or project, you can focus on the workspace purpose. This approach can be more balanced in the distribution of content among workspaces.

Examples: *Quarterly Financials* or *Product Launch Analysis*

The advantages of option 1 include:

- It's more straightforward to manage user access for who can edit or view content because access is scoped per subject area.

- When users across organizational boundaries access content, structuring workspaces by subject area is more flexible and easier to manage, which is compared to option 2, discussed next.
- One scope per subject area is a good compromise between workspaces that contain too many items and workspaces that contain too few items.

A disadvantage of option 1 is that, depending on how narrow or wide workspaces are defined, you still risk users creating an excessive number of workspaces. Finding content can be challenging for users when content is spread across many workspaces.

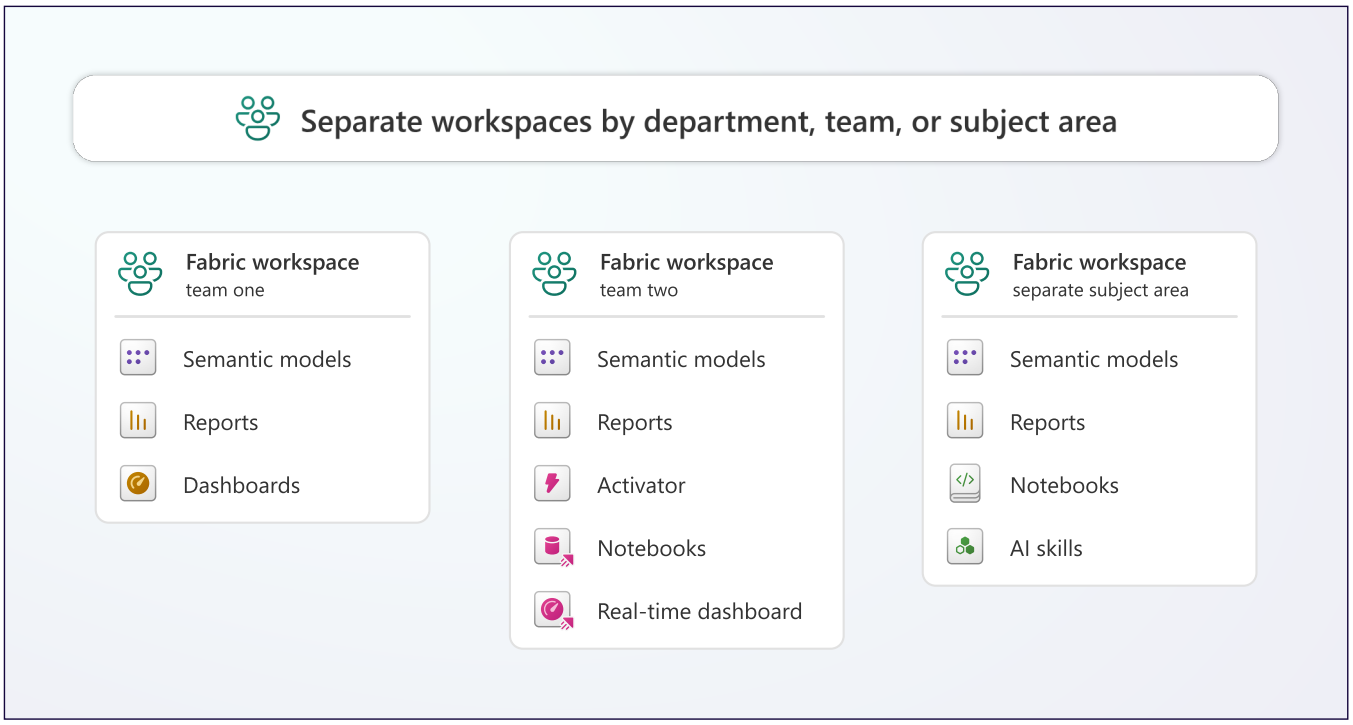
> **Tip**
>
> When they're well-planned and managed, creating one workspace per subject area or project usually results in a manageable number of workspaces.

Option 2: One workspace per department or team

A common approach is to create one workspace per department, team, or business unit. In fact, alignment with the organizational chart is the most common way people begin planning workspaces. But this approach isn't ideal for all scenarios.

Examples: *Finance Department* or *Sales Team Analytics*

The following diagram depicts a generalized example of how you might separate workspaces by department, team, or subject area. Option 1 and option 2 look the same. The items to include in each workspace depends on the nature of the data each department, team, or subject area focuses on, and how they intend to use the data.



The advantages of option 2 include:

- Getting started with planning is simple. All content that department users need is in one workspace.
- It's easy for users to know which workspace to use because all their content is published in the workspace associated with their department or team.
- Managing security roles can be straightforward, especially when you use the best practice of assigning Microsoft Entra groups to workspace roles.

The disadvantages of option 2 include:

- The result is often a broad-scoped workspace that contains many items. A broadly defined workspace scope can make it challenging for users to locate specific items.
- Because a one-to-one relationship exists between a workspace and a Power BI app, a broadly defined workspace can result in apps for users that contain lots of content. You can mitigate the issue by excluding certain workspace items from the app and via design of app navigation.
- When users from other departments need to view specific workspace items, managing permissions can be more complex. A risk is that people assume that everything in the department workspace is for their eyes only. Another risk is that sharing individual items instead of roles is used to accomplish granular viewing permissions.
- If some content creators need permissions to edit some but not all items, it's not possible to set those permissions in a single workspace. Workspace roles, which determine edit or view permissions, are defined at the workspace level.
- When you have a large number of workspace items, you often need to use strict naming conventions for items so that users can find what they need.

- Broad workspaces with many items might run into a technical limitation on the number of items that can be stored in a workspace.

> **Tip**
>
> When you create workspaces that align with your organizational chart, you often end up with fewer workspaces. However, you might end up with workspaces that contain disproportionately large amounts of content. We recommend that you don't align workspaces per department or team when you expect to have a significant number of items or many users.

Option 3: Workspace for a specific report or app

Creating a workspace for each report or type of analysis isn't recommended except in specific circumstances.

Examples: *Daily Sales Summary* or *Executive Bonuses*

Advantages of option 3 include:

- The purpose of a narrowly defined workspace is clear.
- Ultra-sensitive content can, and often should, be segregated into its own workspace so that it can be managed and governed explicitly.
- Fine-grained workspace permissions are applicable to a few items. This setup is useful when, for example, a user is permitted to edit one report but not another.

Disadvantages of option 3 include:

- If overused, narrowly defined workspaces can lead to a large number of workspaces.
- A large number of workspaces requires more effort for users. Although users can rely on search, finding the right content in the right workspace can be frustrating.
- A larger number of workspaces increases the auditing and managing workload.

> **Tip**
>
> You should create a workspace with a narrow scope, such as an individual report, only for specific reasons. It should be the exception rather than the rule. Occasionally, **separating scorecards** into their own workspace is a useful technique. For example, using a separate workspace is helpful when a scorecard presents goals that span multiple subject areas. It's also helpful to set up specific permissions for managing and viewing the scorecard.

Checklist of key decisions and actions when you consider the subject area and scope of workspace content:

- ✓ **Assess how workspaces currently are set up**: Review how people currently use workspaces. Identify what works well and what doesn't work well. Plan for potential changes and user education opportunities.
- ✓ **Consider the best workspace scope**: Identify how you want people to use workspaces based on purpose, subject area, scope, and who's responsible for managing the content.
- ✓ **Identify the location of highly sensitive content**: Determine when you can justify creating a specific workspace to store only highly sensitive content.
- ✓ **Create and publish documentation about using workspaces**: Create useful documentation or FAQs for your users about how they are expected to organize and use workspaces. Make this information available in training materials and in your centralized portal.

Workspace item types

A common practice to decouple data assets from analytical assets is to separate data workspaces from reporting workspaces.

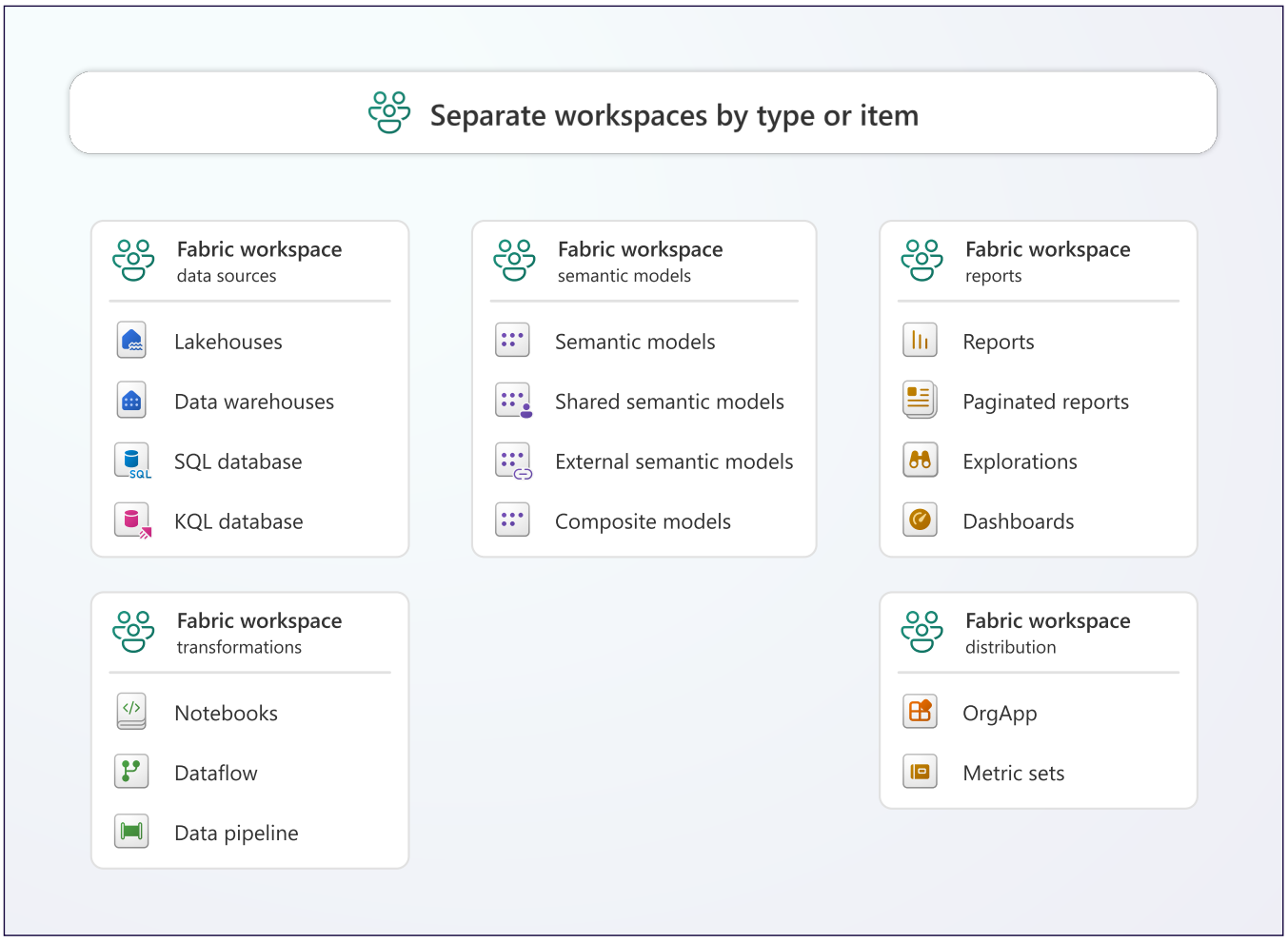
- A *data workspace* is dedicated to storing and securing data items such as a lakehouse, warehouse, data pipeline, dataflow, or semantic model.
- A *reporting workspace* is focused more on downstream analytical activities. A reporting workspace stores and secures only items like reports, dashboards, and metrics. Reporting workspaces typically include Power BI content, but it's not required.

In Fabric, you might extend this separation to have distinct workspaces for other item types.

Some examples include:

- *Data source workspaces* for data warehouses, lakehouses, and SQL databases that store data
- *Data transformation workspaces* for data pipelines, notebooks, and dataflows that transform data
- *Distribution workspaces* for scorecards, metric sets, and organization applications that distribute data to users

The following diagram depicts an example of how you might separate workspaces by item type.



> **Tip**
>
> Each Fabric experience allows you to create various types of items. These items don't always neatly fit the concept of data versus reporting or analytical content. One example is a Fabric notebook that can be used in many different ways. Users might use a Fabric notebook to load and transform data in a lakehouse, to submit Spark SQL queries, or to analyze and visualize data by using PySpark. When the workspace contains mixed workloads, we recommend that you focus primarily on the workspace purpose and ownership of the content as described in this article.

The advantages for separating data workspaces from reporting workspaces include:

- Critical organizational data, such as an endorsed lakehouse or semantic model, can reside in a specific workspace designed to make reusable data available at enterprise scale. Common examples include:
  - Report creators can locate and reuse trustworthy shared semantic models more easily. For more information, see the managed self-service BI usage scenario.
  - Semantic model creators can locate trustworthy dataflows or lakehouse tables more easily. For more information, see the self-service data preparation usage scenario and the advanced self-service data preparation usage scenario.

- Access management can be centralized for critical organizational data. Managing access separately for the data workspace compared with reporting workspaces is useful when different people are responsible for data and reports. With managed self-service BI, it's common to have many report creators and fewer data creators.
- Limiting who can edit and manage semantic models minimizes the risk of unintentional changes, especially to critical data items that are reused for many purposes or by many users. Physical separation reduces the chances of inadvertent, or unapproved, changes. This extra layer of protection is helpful for certified semantic models, which the organization relies on for their quality and trustworthiness.
- Co-ownership scenarios are clarified. When shared semantic models are delivered from a centralized BI or IT team, self-service content creators in business units publish reports. A good practice is to segregate the semantic models in a separate workspace. This approach avoids the ambiguity of co-ownership scenarios because ownership and responsibility per workspace is more clearly defined.
- Row-level security (RLS) is enforced. When you encourage creators to work in different workspaces, they don't have unnecessary edit permissions to the original semantic model. The advantage is that RLS and object-level security (OLS) are enforced for content creators and content viewers.

Disadvantages for separating data workspaces from reporting workspaces include:

- A workspace naming convention is required to differentiate a data workspace from a reporting workspace.
- Extra user education is required to ensure that content authors and consumers know where to publish and find content.
- Sometimes it's challenging to clearly delineate the item types that should be contained in a workspace. Over time, a workspace can end up containing more types of content than was originally intended.
- Use of separate workspaces results in a larger number of workspaces that you need to manage and audit. As you plan for purpose, scope, and other considerations (such as the separation of development, test, and production content) the approach to workspace design can become more complicated.
- Extra change management processes could be required to track and prioritize requested changes to centralized data items, particularly when report creators have requirements beyond what composite models and report-level measures can handle.

Workspace development stage

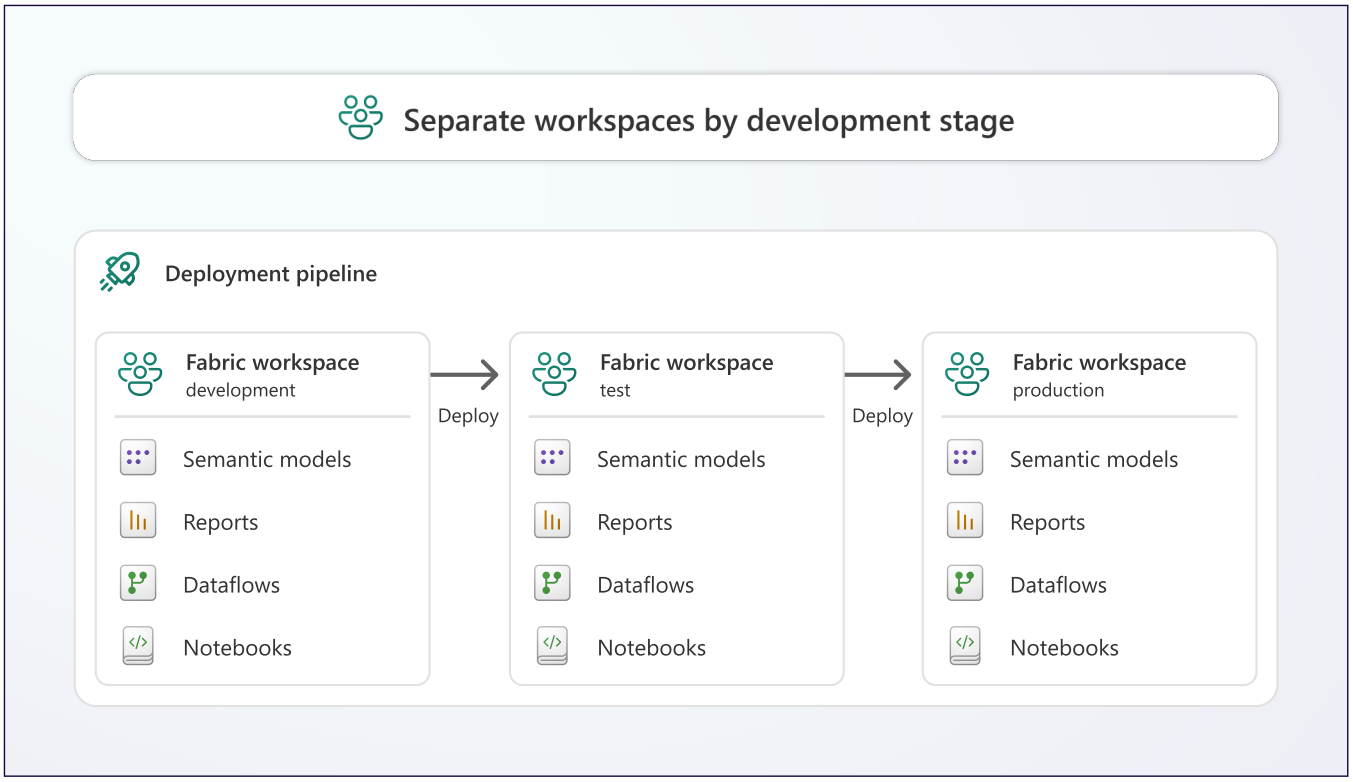
A common practice is to use separate workspaces for different stages of content development. Typically, this practice involves the following stages:

- *Development workspaces* for untested changes

- *Test workspaces* for dedicated internal and user testing
- *Production workspaces* for releasing content for consumers

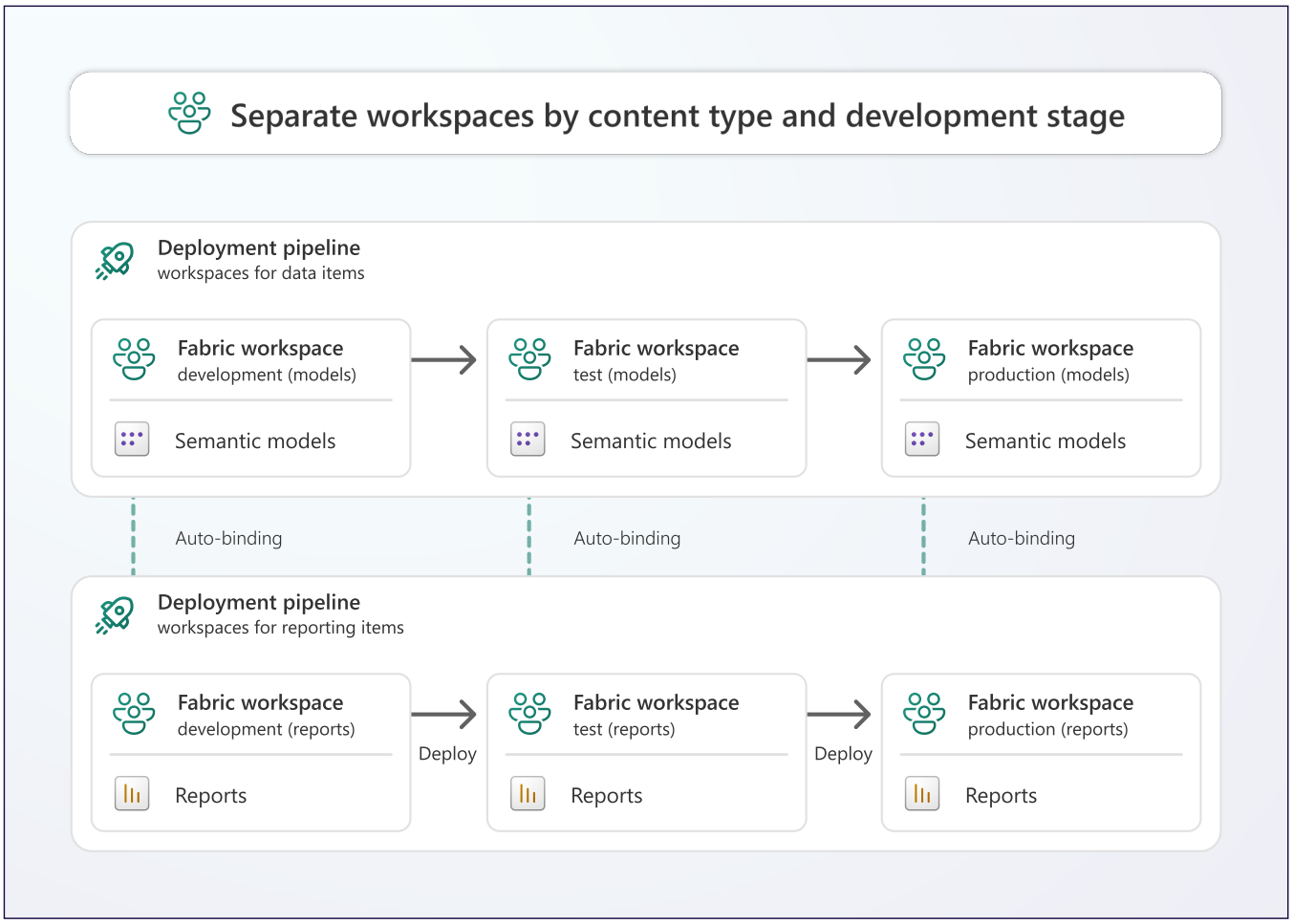
In Fabric, you can add workspaces for each stage to a deployment pipeline. A deployment pipeline helps with content lifecycle management, by letting pipeline administrators compare and deploy changes between stages. Typically, you first publish content to the earliest stage (like *development*) and then deploy it to the next stage (like deploying content from *development* to *test* workspaces, and then from *test* to *production* workspaces).

Here's an example of this setup:



You can also combine separating workspaces by both development stage and item type. If you use deployment pipelines, you can use *autobinding* to ensure that stages are linked. Autobinding ensures that, for example, reports in the *report development workspace* point to the correct semantic model in the *model development workspace*.

Here's an example:



Optionally, you might have other workspaces, such as the following types of workspaces:

- *Private workspaces* for creators to work in isolation. Using this setup is a common practice for collaborating on content by using Git integration because each content creator works on their own separate copy (branch) of the content to avoid disrupting each other's work. Then, creators can open a pull request to merge their changes into another branch that syncs to the development workspace, which creators can view, but not modify or publish content to.
- *Preproduction workspaces* for creators to perform specific tests before releasing content. These tests might include performance testing or testing supporting resources like data gateways and apps.
- *Sandbox workspaces* for creators to freely experiment and conduct *improvised* explorations. Sandbox workspaces are typically emptied at a regular cadence (often automatically, such as by using APIs or notebooks). Content that creators want to retain from a sandbox workspace can be copied to a personal or private workspace for more development.

> **Tip**
>
> We recommend that you organize workspaces by a minimum of two development stages. Taking this approach ensures separation between development or testing by creators, and

consumption by business users. If you use a single workspace stage, you frequently will struggle to avoid disrupting existing content that is consumed from that workspace or from changes from other creators.

Also, you can organize workspaces using multiple approaches. For example, you can use separate data and reporting workspaces that have multiple development stages.

The advantages for separating workspaces by development stage include:

- You can support more sophisticated and structured development processes, including:
  - Deployment Pipelines to copy content between stages
  - Git integration and branching strategies for deployment and release
  - Notebooks to orchestrate or automate certain tasks in the content lifecycle
- You can avoid disrupting production content that consumers use.
- You have better access control for content.

The disadvantages for separating workspaces by development stage include:

- You must manage and govern more workspaces, which create more overhead.
- You must plan for new deployment and post-deployment activities.
- You might have to account for other, non-supported tools or features that deploy or promote content through the development stages.
- More sophisticated workspace access policies are required to prevent creators from publishing to and using the wrong stage.

Intra-workspace organization

In addition to organizing your workspace structure, you also have to organize content *in* a single workspace.

To better organize content in a workspace, consider the following guidelines:

- Use clear naming conventions to easily identify different content. Consider using numerical prefixes (like *01 - Daily Sales* to order content alphabetically, if needed).
- Use *task flows* to group content by its purpose in your workflow into tasks. Using task flows helps you quickly identify and select similar content (by selecting the task in the task flow). It's important if you decide to include many item types with different purposes in the same workspace.
- Use workspace folders to organize similar content into groups. Workspace folders can be used instead of or in addition to tasks in task flows.

- Use endorsement and sensitivity labels to label content appropriately based on their endorsement and sensitivity status.

Checklist of key decisions and actions when you consider item types that a user can store in a workspace:

- ✓ **Determine your objectives for data reuse**: Decide how to achieve data reuse as part of a managed self-service BI strategy.
- ✓ **Update the tenant setting for who can use semantic models across workspaces**: Determine whether this capability can be granted to all users. If you decide to limit who can use semantic models across workspaces, consider using a group such as *Fabric approved report creators*.

Workspace access

Because the primary purpose of a workspace is collaboration, workspace access applies primarily to users who create and manage workspace content. Access can also be relevant when the workspace is used for viewing content. This purpose is a secondary purpose for workspaces, as described earlier in the article.

When you begin to plan for workspace roles, it's helpful to consider the following questions:

- How does collaboration occur in the workspace?
- Will consumers directly view content in the workspace?
- Who is responsible for managing the content in the workspace?
- Who can view content stored in the workspace?
- Do you intend to assign individual users or groups to workspace roles?

A best practice is to use groups to assign workspace roles. Security groups, mail-enabled security groups, distribution groups, and Microsoft 365 groups are all supported for workspace roles. For more information about using groups, see Tenant-level security planning.

When you plan to use groups, you might consider creating one group per role per workspace. For example, to support the *Quarterly Financials* workspace, you could create the following groups:

- *Fabric workspace **admins** – Quarterly Financials*
- *Fabric workspace **members** – Quarterly Financials*
- *Fabric workspace **contributors** – Quarterly Financials*

- *Fabric workspace **viewers** – Quarterly Financials*
- *Power BI **app viewers** – Quarterly Financials*

> **Tip**
>
> You gain flexibility when you create groups per role and per workspace. However, the trade-off is more groups to create and manage. Also, managing a large number of groups can be challenging when only IT creates and maintains groups. You can mitigate the challenge by enabling **self-service group management** to certain satellite members. These members can include the CoE, champions, or trusted users who are trained in how to manage role memberships for their business units. For more information, see **Tenant-level security planning**.

When data workspaces are separated from reporting workspaces, as described earlier in this article, it results in an even larger number of groups. Consider how the number of groups doubles from five to 10 when you separate data and reporting workspaces:

- *Fabric **data** workspace **admins** – Quarterly Financials*
- *Fabric **reporting** workspace **admins** – Quarterly Financials*
- *Fabric **data** workspace **members** – Quarterly Financials*
- *Fabric **reporting** workspace **members** – Quarterly Financials*
- *Fabric **data** workspace **contributors** – Quarterly Financials*
- *Fabric **reporting** workspace **contributors** – Quarterly Financials*
- *Fabric **data** workspace **viewers** – Quarterly Financials*
- *Fabric **reporting** workspace **viewers** – Quarterly Financials*
- *Power BI **app viewers** – Quarterly Financials*

When multiple workspaces exist for development, test, and production, it results in an even larger number of groups. Potentially, the number of groups might triple. For example, for just the data workspace admins, you would create these three groups:

- *Fabric data workspace admins – Quarterly Financials **[Dev]***
- *Fabric data workspace admins – Quarterly Financials **[Test]***
- *Fabric data workspace admins – Quarterly Financials*

The preceding examples are intended to convey that the use of groups that map to workspace roles can quickly become unmanageable.

> **Tip**
>
> In some scenarios, fewer groups are needed, especially in the area of development. For example, you might not need to specify a workspace viewers group in development. That

group is needed only for testing and production. Or you might be able to use the same workspace admins group for development, test, and production. For more information about development, test, and production, see **Workspace lifecycle management**.

The effective use of groups for workspace roles can require considerable planning. Be prepared to encounter scenarios when existing groups (that might be aligned with the organizational chart) don't meet all your needs for managing Fabric content. In this case, we recommend that you create groups specifically for this purpose. The words *Fabric* or *Power BI* are included in the group name in the preceding examples for this purpose. If you have multiple business intelligence tools, you can choose to use only *BI* as the prefix instead. That way, you can use the same groups across multiple tools.

Lastly, the examples show one workspace - *Quarterly Financials* - but often it's possible to manage a collection of workspaces with one set of groups. For example, multiple workspaces owned and managed by the finance team might be able to use the same groups.

> **Note**
>
> Often, you plan security more broadly, taking into consideration semantic model Read and Build permissions and row-level security (RLS) requirements. For more information about what to consider to support report consumers and content creators, see the security implementation planning articles. This article focuses only on workspace roles as part of the workspace planning process.

Checklist of key decisions and actions when you consider workspace access:

- ✓ **Refer to roles and responsibilities**: Use the roles and responsibilities information prepared earlier to plan for workspace roles.
- ✓ **Identify who'll own and manage the content**: Verify that all the items you expect to store in a single workspace align with the people who are responsible for owning and managing the content. If mismatches occur, consider how the workspaces could be better organized.
- ✓ **Identify who'll view content in the workspace**: Determine whether people will view content directly from the workspace.
- ✓ **Plan for the workspace roles**: Determine which people are suited to the Admin, Member, Contributor, and Viewer roles for each workspace.
- ✓ **Decide on group or individual role assignments**: Determine whether you intend to assign individual users or groups to workspace roles. Check whether you can use existing groups

for workspace role assignments.

- ✓ **Determine whether new groups need to be created**: Consider carefully whether you need to create a new group per workspace role. Bear in mind that it can result in creating and maintaining many groups. Determine what the process is when a new workspace is created and how related groups will be created.
- ✓ **Configure and test the workspace role assignments**: Verify that users have the appropriate security settings they need to be productive when they create, edit, and view content.

Workspace domain

As described earlier in this article, it's critical to have clarity on workspace ownership. One way to further support workspace ownership in Fabric is by using domains. A *domain* is a logical grouping of multiple workspaces that have similar characteristics.

For more information about planning for domains in your tenant, see Workspace domains.

Workspace settings

You can set up several settings for each individual workspace. These settings can significantly influence how collaboration occurs, who is allowed to access the workspace, and the level of data reusability across Fabric workloads.

Workspace license mode

Each workspace has a license mode setting. You can set a workspace license mode to **Pro**, **Premium per user**, **Premium capacity**, **Embedded**, **Fabric capacity**, or **Trial**.

> ⓘ **Important**
>
> This article refers to Power BI Premium or its capacity subscriptions (P SKUs). Currently, Microsoft is consolidating purchase options and retiring the Power BI Premium per-capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

The type of license is important for workspace planning because it determines:

- **Features**: Different features are supported. PPU includes more features (such as deployment pipelines) that aren't available in Pro. Many more Fabric features (such as lakehouses) become available for workspaces assigned to a Fabric capacity.

- **Content access**: The license type determines who can access content in the workspace:
  - Only users who have a PPU license (in addition to being assigned a workspace role) can access a PPU workspace.
  - If you expect to deliver content to content viewers who have a free license, you need a license of F64 or higher.

- **Data storage location**: When you need to store data in a specific geographic region (outside of your home region), that becomes possible with a workspace assigned to a capacity (and, accordingly, the capacity is created in that region). For more information about data storage location, see Tenant setup.

Checklist of key decisions and actions when you consider workspace license mode:

- ✓ **Consider which features are required for each workspace**: Determine the feature requirements of each workspace. Consider differences in workload and which users you intend to access the workspace.
- ✓ **Set the workspace license mode**: Review and update each workspace license mode according to which features are needed by each workspace.

Workspace lifecycle management

When content creators collaborate to deliver analytical solutions that are important to the organization, you must make various *lifecycle management* decisions. The lifecycle management processes are also called *continuous integration/continuous delivery (CI/CD)*, and they're one aspect of DevOps.

Lifecycle management considerations include how to:

- Ensure timely, reliable, and consistent delivery of content.
- Communicate and coordinate activities between multiple content creators who are working on the same project.
- Resolve conflicts when multiple content creators edit the same item in the same project.
- Structure a straightforward and reliable deployment process.
- Roll back deployed content to a previous stable, working version.
- Balance fast releases of new features and bug fixes as you safeguard production content.

Fabric has two main lifecycle management components:

- **Version control of content**: Git integration allows content owners and creators to create versions of their work. It can be used with web-based development in a workspace, or when teams develop in a client tool, such as Power BI Desktop. Version control (also called *source control*) is achieved by tracking all revisions to a project by using *branches* associated with local and remote *repositories* in Azure DevOps. Changes are *committed* at regular intervals to branches in the remote repository. When a content creator completes revisions that are tested and approved, their branch is *merged* with the latest version of the solution in the main remote repository (after they resolve any merge conflicts). Git integration can be specified for each workspace in the Fabric portal if the feature is enabled in the tenant settings.
- **Promoting content**: Deployment pipelines are primarily focused on release management in order to maintain a stable environment for users. You can assign a workspace to a stage (development, test, or production) in a deployment pipeline. Then, you can easily and systematically *promote*, or *deploy*, your content to the next stage.

When you combine the lifecycle management features, during your planning process, cosider using best practices. For example, you might choose to use Git integration for your development workspace and deployment pipelines to publish to your test and production workspaces. Those types of decisions require using the agreed-upon practice consistently. We recommend that you do a proof of concept to fully test your setup, processes, and permissions model.

Checklist of key decisions and actions when you plan for lifecycle management of your workspaces:

- ✓ **Determine how users need to use version control**: Analyze how your self-service and advanced content creators work to determine whether file versioning with OneDrive for Business or SharePoint is appropriate. Introduce Git integration for advanced users who need more capabilities. Prepare to support both types of users.
- ✓ **Determine how users need to promote content**: Analyze how your self-service and advanced content creators work to determine whether deployment pipelines are a good fit for promoting content.
- ✓ **Decide whether Git integration should be enabled**: Consider whether Git integration with workspaces is a good fit for how your content creators work. Set the *Users can synchronize workspace items with their Git repositories* tenant setting to align with this decision. Review

each of the Git integration tenant settings and set them according to your governance guidelines.

- ✓ **Do a proof of concept**: Conduct a technical proof of concept to clarify how you intend for Git workspaces and deployment pipelines to work together.
- ✓ **Decide which workspaces should have Git integration**: Consider how your content creators work, and which workspaces should be assigned to a development, test, or production (release) branch.
- ✓ **Verify licenses**: Confirm that you have a capacity license available to use Git integration. Ensure that each workspace is assigned to a Fabric capacity or Power BI Premium capacity.
- ✓ **Set up Azure DevOps**: Work with your administrator to set up the Azure DevOps projects, repositories, and branches that you need for each workspace. Assign appropriate access to each repository.
- ✓ **Connect workspaces**: Connect each workspace to the appropriate Azure DevOps repository.
- ✓ **Consider who should deploy to production**: Make decisions about who can update production content and how. Ensure that these decisions align with how workspace ownership is handled in your organization.
- ✓ **Educate content creators**: Ensure that all your content creators understand when to use lifecycle management features and practices. Educate them on the workflow and how different workspaces impact lifecycle management processes.

Workspace integration with Data Lake Storage Gen2

It's possible to connect a workspace to an Azure Data Lake Storage Gen2 account. You might take this approach for two reasons:

- **Storage of Power BI dataflows data**: If you choose to bring-your-own-data-lake, the data for Power BI dataflows (Gen1) could be accessed directly in Azure. Direct access to dataflow storage in Data Lake Storage Gen2 is helpful when you want other users or processes to view or access the data. It's especially helpful when your goal is to reuse dataflows data beyond Power BI. You have two choices for assigning storage:
  - Tenant-level storage, which is helpful when you want to centralize all data for Power BI dataflows into one Data Lake Storage Gen2 account.
  - Workspace-level storage, which is helpful when business units manage their own data lake or have certain data residency requirements.

- **Backup and restore for Power BI semantic models**: The Power BI semantic model backup and restore feature is supported for workspaces that are assigned to capacity or PPU. This feature uses the same Data Lake Storage Gen2 account that's used for storing Power BI dataflows data (described in the previous bullet point). Semantic model backups are helpful for:

- Complying with data retention requirements
- Storing routine backups as part of a disaster recovery strategy
- Storing backups in another region
- Migrating a data model

> **ⓘ Important**
>
> Setting Azure connections in the Fabric admin portal doesn't mean that all dataflows for the entire tenant are stored by default to a Data Lake Storage Gen2 account. To use an explicit storage account (instead of internal storage), each workspace must be explicitly connected. It's critical that you set the workspace Azure connections *before you create any Power BI dataflows* in the workspace.

Checklist of key decisions and actions when you plan for workspace integration with Data Lake Storage Gen2:

- ✓ **Decide whether the workspace will be used in ways that require Azure Storage**: Consider whether a bring-your-own-data-lake scenario would be useful for the storage of dataflows and whether you have requirements to use the semantic model backup and restore functionality.
- ✓ **Determine which Azure Storage account to use**: Select an Azure Storage account that has the hierarchical namespace enabled (Data Lake Storage Gen2) for tenant-level (centralized) storage of dataflows data or semantic model backups. Ensure you have the Azure Storage account information readily available.
- ✓ **Configure the tenant-level storage account**: In the Fabric admin portal, set the tenant-level Data Lake Storage Gen2 storage account.
- ✓ **Decide whether workspace administrators can connect a storage account**: Have discussions to understand the needs of decentralized teams, and whether individual teams are currently maintaining their own Azure Storage accounts. Decide whether this capability should be enabled.
- ✓ **Configure the admin setting for workspace-level storage**: In the Fabric admin portal, enable the option that allows workspace administrators to connect their own storage account.
- ✓ **Set the workspace-level Azure Storage connections**: Specify the Azure Storage account for each individual workspace. You must set the storage account prior to creating any Power BI dataflows in the workspace. If you intend to use semantic model backups, ensure the workspace license mode is set to capacity or PPU.

- ✓ **Update your workspace management documentation**: Ensure that your workspace management documentation includes information about how to assign Data Lake Storage Gen2 storage accounts correctly. Make the information available in your centralized portal and in training materials.

Workspace integration with Log Analytics

Log Analytics is a feature of Azure Monitor. You can use Log Analytics to review diagnostic data generated by the Analysis Services engine, which hosts Power BI semantic models. Workspace-level logs are useful for analyzing performance and trends, performing data refresh analysis, analyzing XMLA endpoint operations, and more. Log Analytics is available only for workspaces assigned to capacity or PPU.

> ⓘ **Note**
>
> Although the names are similar, the data sent to **Log Analytics** is different from the data captured by the **Power BI activity log**. The data sent to Log Analytics is concerned with **events** generated by the Analysis Services engine (for example, *Query begin* and *Query end* events). Conversely, the activity log is concerned with tracking **user activities** (for example, *View report* or *Edit report* events).

For more information about semantic model event logs, see Data-level auditing.

For more information about how to set up Log Analytics for use with Power BI, see Configure Log Analytics for Power BI. Be sure to understand prerequisites you must have in place to implement the integration.

Checklist of key decisions and actions when you plan for workspace integration with Log Analytics:

- ✓ **Decide whether workspace administrators can connect to Log Analytics**: Determine whether all or some workspace administrators are permitted to use Log Analytics to analyze workspace-level logs. If access is restricted to only certain people, decide which group to use.
- ✓ **Set up the tenant setting for Log Analytics connections**: In the Fabric admin portal, set the tenant setting according to the decision for which workspace administrators set connections.

- ✓ **Set the Log Analytics workspace for each workspace**: In the workspace settings, specify the Log Analytics information for each workspace. To capture workspace-level logs, ensure that the workspace license mode is set to capacity or PPU.
- ✓ **Update your workspace management documentation**: Ensure that your workspace management documentation includes information about how to assign a workspace to Log Analytics.

Other workspace properties

Several other workspace properties can provide helpful information. For governed workspaces, we recommend that you set these properties.

Here are some suggestions for how to set the key settings to improve the experience for your users:

- **Workspace description**: A good workspace description includes a brief but specific explanation of what type of content is in the workspace. You can use up to 4,000 characters to describe:
  - The purpose for the workspace
  - The target audience
  - The type of content published to the workspace
  - Whether the workspace is considered governed
  - Whether the workspace includes development, test, or production data
  - Who to contact for questions or support

- **Workspace contacts**: The workspace contact list includes the workspace administrators by default. If you have technical content owners that are different from the subject matter experts, you might find it helpful to specify other contacts. Other contacts could be groups or individuals who can answer questions about the workspace content.

- **Workspace image**: Consistent use of workspace images can be helpful for users when they scan a list of workspaces.

  Consider using an image to help users identify:
  - The domain or subject area
  - The business unit or team that owns and manages the content
  - A data workspace dedicated to storing reusable items, such as a lakehouse, warehouse, data pipeline, dataflow, or semantic model
  - A reporting workspace dedicated to storing analytical items, such as reports, dashboards, or metrics

- **Data model settings**: Allows workspace members, administrators, and users with Build permissions on the semantic models to edit Power BI data models by using the web

interface. This setting is used with the *Users can edit data models in the Power BI service* tenant setting. This setting should align with your decisions and processes for how to create, manage, and deploy content. Also, consider your method for version control as described earlier.

Checklist of key decisions and actions when you consider other workspace properties:

- ✓ **Specify the workspace description**: Ensure to include a helpful and thorough description in the workspace description.
- ✓ **Use a helpful image for the workspace**: Set a consistent image for the workspace that visually helps users understand its subject area, who owns and manages content in the workspace, and the type of content stored in the workspace.
- ✓ **Identify contacts for the workspace**: Verify whether workspace administrators should be the workspace contacts, or whether to set specific users or groups for contact.
- ✓ **Specify data model settings**: Consider which workspaces can permit web-based data model editing. Set the *Users can edit data models in the Power BI service* tenant setting according to your preferences for who can edit and manage content.

Other technical factors

Other technical factors might influence your workspace setup:

- Integrating content with other tools and services might have licensing implications. For example, if you embed a Power Apps visual in a Power BI report, you must have relevant Power Apps licenses.
- Per-workspace storage limits apply to the amount of data you can store in a Pro workspace. If using capacity or PPU isn't an option, consider how to work within the storage limits during the workspace planning process.
- When you install a template app from Microsoft AppSource, the app creates a new workspace that has a narrow subject and scope.

Checklist of key decisions and actions when you consider other technical factors:

- ✓ **Pay attention to technical factors**: As you work through the planning process, determine whether a technical consideration or limitation (such as per-workspace storage limits) influences your decision-making process.
- ✓ **Reorganize workspace content**: If storage limits might become a problem, create separate workspaces now, and then republish content to these new workspaces.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see:

- Introduction to Power BI implementation planning

Power BI implementation planning: Content lifecycle management

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces the content lifecycle management series of articles, which helps you to plan how to manage the lifecycle of your Power BI content. The series of articles is primarily targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators might need to collaborate with other administrators, like those who oversee Microsoft 365 or Azure DevOps.
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content. These teams might also include roles like release managers who handle the lifecycle of content releases, or engineers who create and manage the components needed to effectively use and support lifecycle management.
- **Content creators and content owners**: Users who create Power BI content that they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the content that they create.

To ensure that you deliver reliable and useful Power BI content to consumers, it's important to follow effective content *lifecycle management* practices. Lifecycle management refers to how you handle content from creation to publication (or *deployment*), including retiring content when users no longer need it. Planning your lifecycle management strategy for Power BI content is an essential step to scale and grow for successful enterprise and self-service analytics.

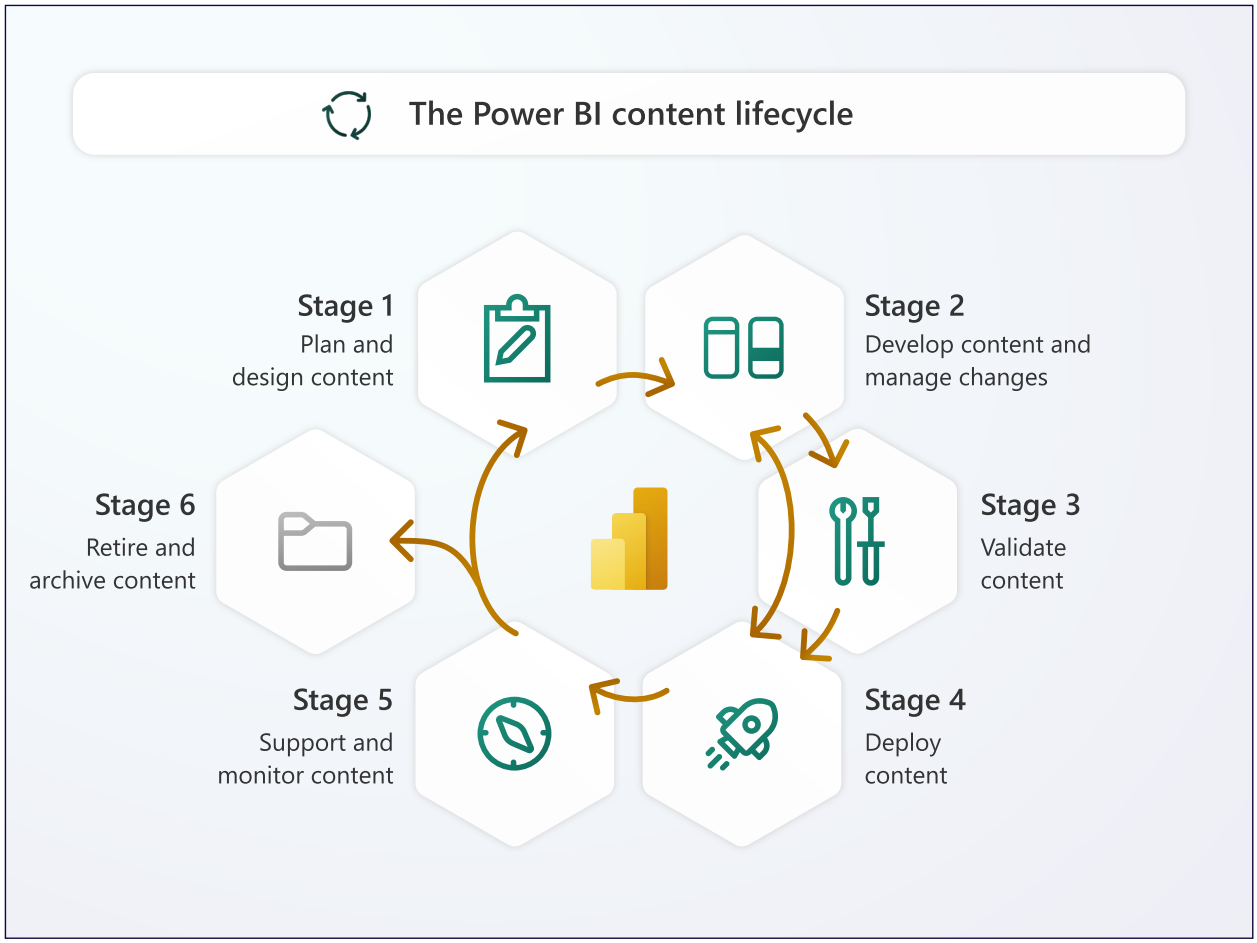
> **Note**
>
> This series provides an overview of Power BI content lifecycle management. It focuses on key considerations and guidance to plan your content lifecycle management strategy. These articles describe different approaches to lifecycle

management from smaller, self-service environments, to more sophisticated enterprise scenarios.

This series focuses primarily on the Power BI workload within Microsoft Fabric. However, the underlying concepts can also be relevant for the other Fabric workloads. Some lifecycle management features discussed in this series can also be used for other items in Fabric.

The Power BI content lifecycle

The following diagram depicts a typical Power BI content lifecycle.



> **Tip**
>
> For guidance about how to plan and create Power BI content, see BI solution planning.

The six stages shown in the diagram include:

1. Plan and design content: You typically collect requirements, design the solution, and make key lifecycle management decisions to start Power BI content creation.

2. **Develop content and manage changes**: You begin to create content and use *version control* (or *source control*) to track and manage content changes during development.
3. **Validate content**: You routinely test content to ensure quality results and to prevent changes in existing content from causing new issues.
4. **Deploy content**: When ready, you deploy content to a workspace, or promote it between different environments (like promoting content from a test to a production workspace).
5. **Support and monitor content**: After it's published, you support content by addressing issues or requests from users. Supporting and monitoring content often leads to planning and creating more content. You also monitor the deployed content to ensure it's reliable.
6. **Retire and archive content**: When consumers no longer need or use content, you should retire it. When you retire content, you typically remove and archive it for later reference, if needed.

Each of these stages is described in detail in the articles in this series on content lifecycle management. The guidance in these articles helps you to determine your approach to manage your Power BI content.

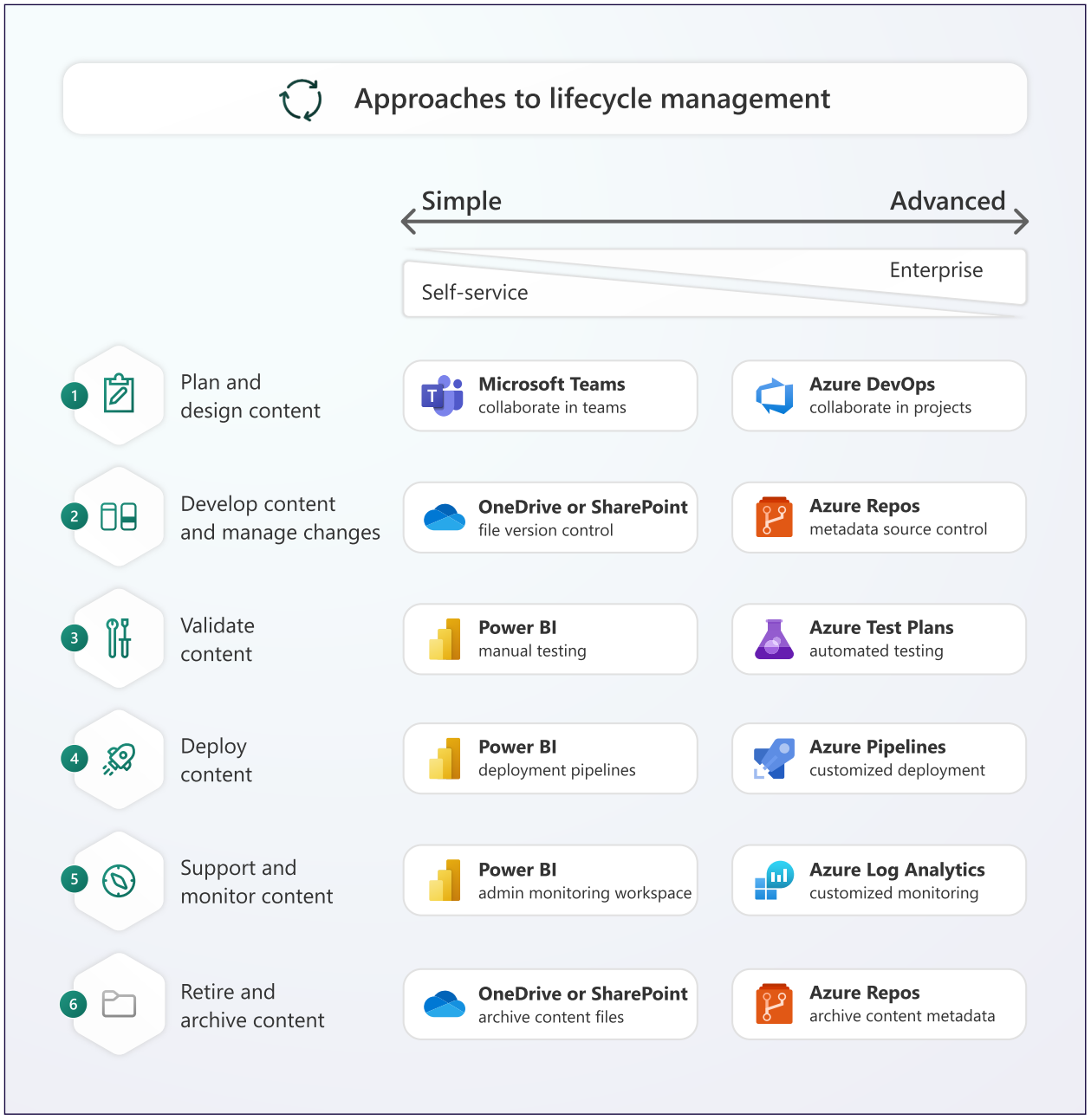
Approaches to manage the content lifecycle

You can manage the lifecycle of Power BI content by using different approaches. These approaches vary in their complexity and robustness, depending on the components and processes you use.

Typically, different approaches range from simple to advanced.

- *Simpler* approaches are typically useful for self-service developers. These approaches involve fewer steps and components to manage, but don't provide the detail and level of control that a departmental or enterprise usage scenario might need. Self-service content publishing is an example of a simpler approach to lifecycle management.
- *Advanced* approaches are typically preferred by larger teams that collaborate on larger or more complex solutions. Because their solutions are more complex, these teams often have a greater need for more automation, customization, and sustainability to manage content. As such, these approaches typically involve more robust and sophisticated processes. However, these processes could be too intricate for smaller deployments, such as personal or team usage scenarios. Enterprise content publishing is an example of an advanced approach to lifecycle management.

For each stage of the content lifecycle, the following diagram depicts some examples of components you might use for simpler or more advanced lifecycle management approaches.



The diagram depicts the following components for each stage in the content lifecycle. The components are examples. The first example is how you might achieve a simple lifecycle management approach; the second example is how you might achieve an advanced lifecycle management approach.

Expand table

Item	Description
1	You can plan and design content by using Microsoft Teams to collaborate in teams, or by using Azure DevOps to collaborate in projects.

Item	Description
2	You can develop content and manage changes by using OneDrive for Business (also known as OneDrive for work or school) to conduct file version control, or by using Azure Repos in Azure DevOps for metadata source control.
3	You can validate content by using Power BI to conduct manual testing, or Azure Test Plans in Azure DevOps to conduct automated testing.
4	You can deploy content by using Power BI deployment pipelines, or by using Azure Pipelines in Azure DevOps to orchestrate CI/CD (continuous integration/continuous deployment).
5	You can support and monitor content by using the default Power BI monitoring reports, like those found in the admin monitoring workspace, or by creating your own customized monitoring report with Azure Log Analytics integration.
6	You can retire and archive content by using OneDrive for Business (also known as OneDrive for work or school) to archive and store files, or by using Azure Repos in Azure DevOps to archive metadata.

Choosing how you approach content lifecycle management depends on your needs and other factors. Here are some key factors to consider as you go through the content in this series.

- **Who will create the content**: Content creators have varying needs, skills, and workflows. Each of these factors can influence the success of different lifecycle management approaches. What works for a larger central team that manages enterprise environments might not be effective for a small, decentralized team that delivers content to a small audience.
- **Whether there's collaborative content creation**: When content creators work together on the same content, there's a higher likelihood of inconsistency and disruption. For example, one creator could overwrite the changes of another. Effective lifecycle management is essential in collaboration to avoid wasted time and enhance productivity.
- **What the type and scope of content is**: Different content may require different approaches. Business-critical content with a high number of consumers should have a more robust lifecycle management approach. Conversely, a small prototype may only require a minimal approach, like removing and archiving it once it's served its purpose.
- **What licensing is in place**: Different lifecycle management options are available depending on the Fabric or Power BI licenses you have. For instance, Premium features like deployment pipelines are available only for Fabric or Premium capacities or users with Premium Per User (PPU) licenses. However, Fabric Git integration is only available with Fabric capacity.

- **How content creators deliver content**: Content creators who use different content delivery scopes might need different approaches to manage that content. A team that distributes content to an internal audience by using Power BI apps will likely require a different approach than a team that distributes embedded content for external customers.
- **How mature the Power BI or Fabric implementation is**: When your implementation reaches a certain scale, more decisions and actions depend on Power BI content. To avoid disruption, content lifecycle management becomes more important as you scale and grow.

> **Important**
>
> There are many other valid approaches to manage the lifecycle of your Power BI content. Select and plan an approach that best fits your **content ownership** and **content delivery scope** strategies, and which helps you meet your needs and achieve your **BI objectives**.

Related content

In the next article in this series, learn how to plan and design content as part of managing the content lifecycle.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Plan and design content

Article • 03/19/2025

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

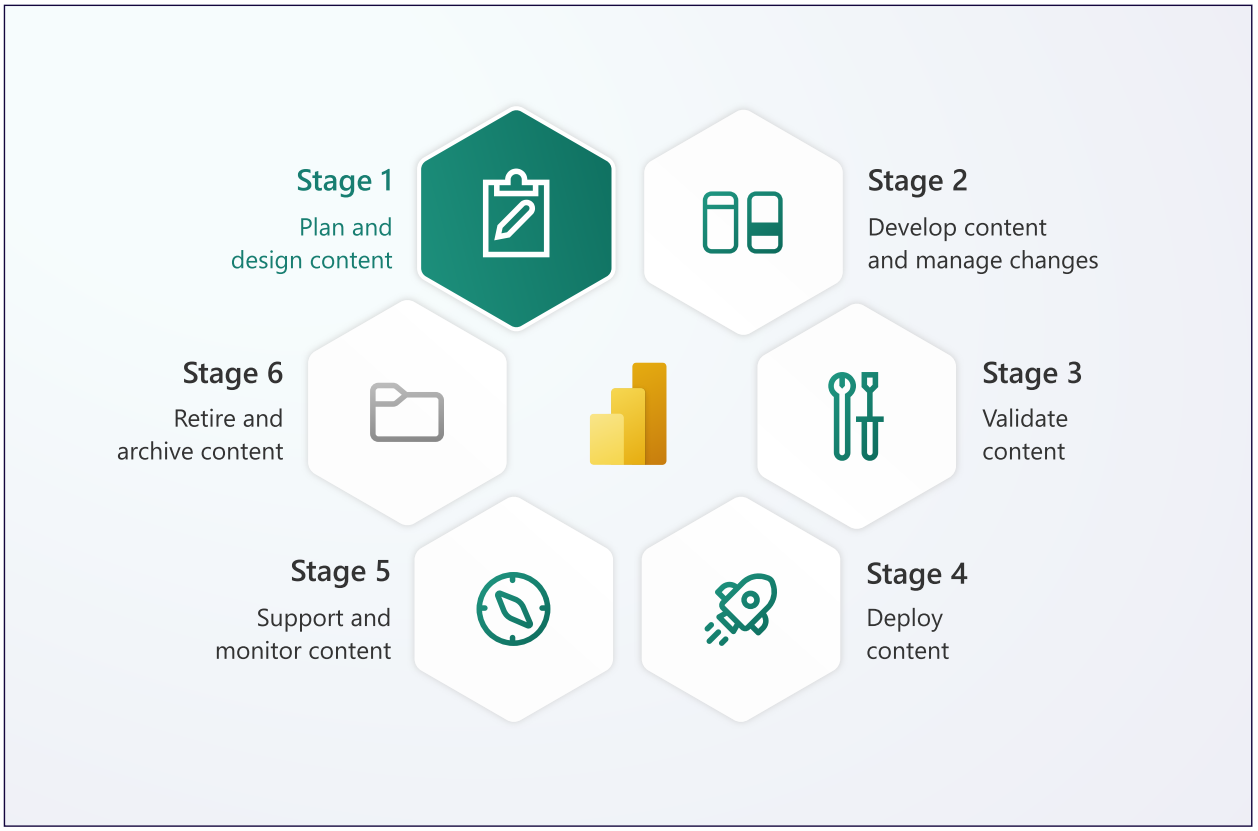
This article helps you to plan and design content as part of managing the content lifecycle. It's primarily targeted at:

- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content.
- **Content creators and content owners**: Users who create content that they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the Power BI content that they create.

Lifecycle management consists of the processes and practices that you use to handle content from its creation to its eventual retirement. As described in the first article in this series, managing the Power BI content lifecycle is important to ensure reliable and consistent delivery of content to business users.

The first stage of the content lifecycle is to plan and design content. You typically start the content lifecycle by performing BI solution planning. You gather requirements to understand and define the problem that your solution should address, and arrive at a solution design. During this planning and design stage, you make key decisions to prepare for the later stages.

The following image depicts the lifecycle of Power BI content, highlighting stage one, where you plan and design content.



> **Note**
>
> For an overview of content lifecycle management, see the first article in this series.

> **Tip**
>
> This article focuses on key considerations and decisions for content planning and design as they pertain to lifecycle management.
>
> - For more information about *how to effectively plan and design a Fabric or Power BI solution*, we recommend that you read the solution planning article.
> - For more information about how to *effectively plan a Power BI migration*, we recommend that you read the Power BI migration series.

When gathering requirements, you should clearly describe aspects about the content that affect your approach to lifecycle management. You should document these aspects as part of your solution planning and design.

The following sections in this article describe the key aspects and considerations of a solution that will motivate your approach to lifecycle management as you plan and design your content.

Identify and describe the content

When you design your solution, you should describe what the content is, who will create it, who will support it, and how critical this content is for the organization. You should address these factors during, or after you finish, gathering requirements as part of your solution design.

> **Note**
>
> Like your requirements, the answers to these questions may change as you develop the solution, or later in its lifecycle. After you answer these questions, be prepared to periodically re-evaluate them when you make changes to content, or as it scales in the number of users that it serves.

Answer the following questions about your content to help you make later lifecycle management decisions.

What is the format of the content?

The type, scope, and complexity of content motivates key decisions about how you'll manage it. For instance, a single report for a limited audience requires a different lifecycle management approach compared to a semantic model that will be used by the entire organization and by multiple different downstream workloads.

Answer questions like the following to help you determine the type of content you'll create.

- Which item types do you expect to create, and how many of each? For instance, will you create data items like dataflows or semantic models, reporting items like reports or dashboards, or a combination of both?
- How is the content delivered to content consumers? For instance, will consumers use data items to build their own content, will they only view centralized reports, or a combination of both?
- How complex is the content? For instance, is it a small prototype, or a large semantic model that encompasses multiple business processes?
- Do you expect the scale, scope, and complexity of the content to grow over time? For instance, will the content encompass other regions or business areas in the future?
- How long do you expect the business to need this content? For instance, will this content support a key initiative of the business that has a finite timeline?

> **Tip**
>
> Consider making an architectural diagram to describe the format of the content. You can include different data sources, item types, and content consumers, and the relationships between these discrete components. An architectural diagram can help to concisely depict the content and its complexity, and it helps you to plan its lifecycle management. You can use the **Fabric icons** and **Azure icons** to create these diagrams in external software. Alternatively, you can use **Azure Diagrams**, which comes with icons and drawing tools to make these diagrams.
>
> For an example of such diagrams, see the Power BI implementation planning **usage scenario diagrams**.

Who will create and support the content?

Content creators have varying needs, skills, and workflows. These factors will influence the success of different lifecycle management approaches. Larger, central teams with collaboration often require more sophisticated content lifecycle management than smaller teams of self-service creators.

Answer questions like the following to help you determine who will create or support the content.

- How many different individuals do you expect to create this content? Will multiple content creators collaborate, or is a single individual responsible for creating the content?
- Are content creators familiar with lifecycle management and related concepts, such as version control? Do content creators understand the benefits of lifecycle management?
- Will the content creators who develop the solution be the same individuals who support it after deployment?
- Do content creators or their teams have existing lifecycle management practices in place to support existing solutions?
- Do content creators currently use lifecycle management tools like Azure DevOps?

> **Important**
>
> Ensure that you clearly document who is responsible for creating content, and who will support it once it's deployed to production. Involve all of these individuals in your content lifecycle management planning.

What is the importance of the content?

Depending upon how important the content is for the business, you'll make different decisions about how to manage it. Business-critical content requires more robust content lifecycle management approaches to safeguard quality and mitigate possible disruptions.

Answer questions like the following to help you determine whether the content is critical.

- How critical is this content to the business? How urgent is the request to develop it?
- Will business-critical decisions or actions be made from information provided by this content?
- How broadly do you expect to distribute this content (from organization-wide to a limited local team)?
- Will executives or other strategic decision makers rely upon this content for their work?
- What is the impact of this content? For instance, if the content is suddenly not available, what business impacts would occur, such as lost revenue or interrupted business processes?

When you've sufficiently identified and described the content you'll create, you should next decide how content creators should collaborate.

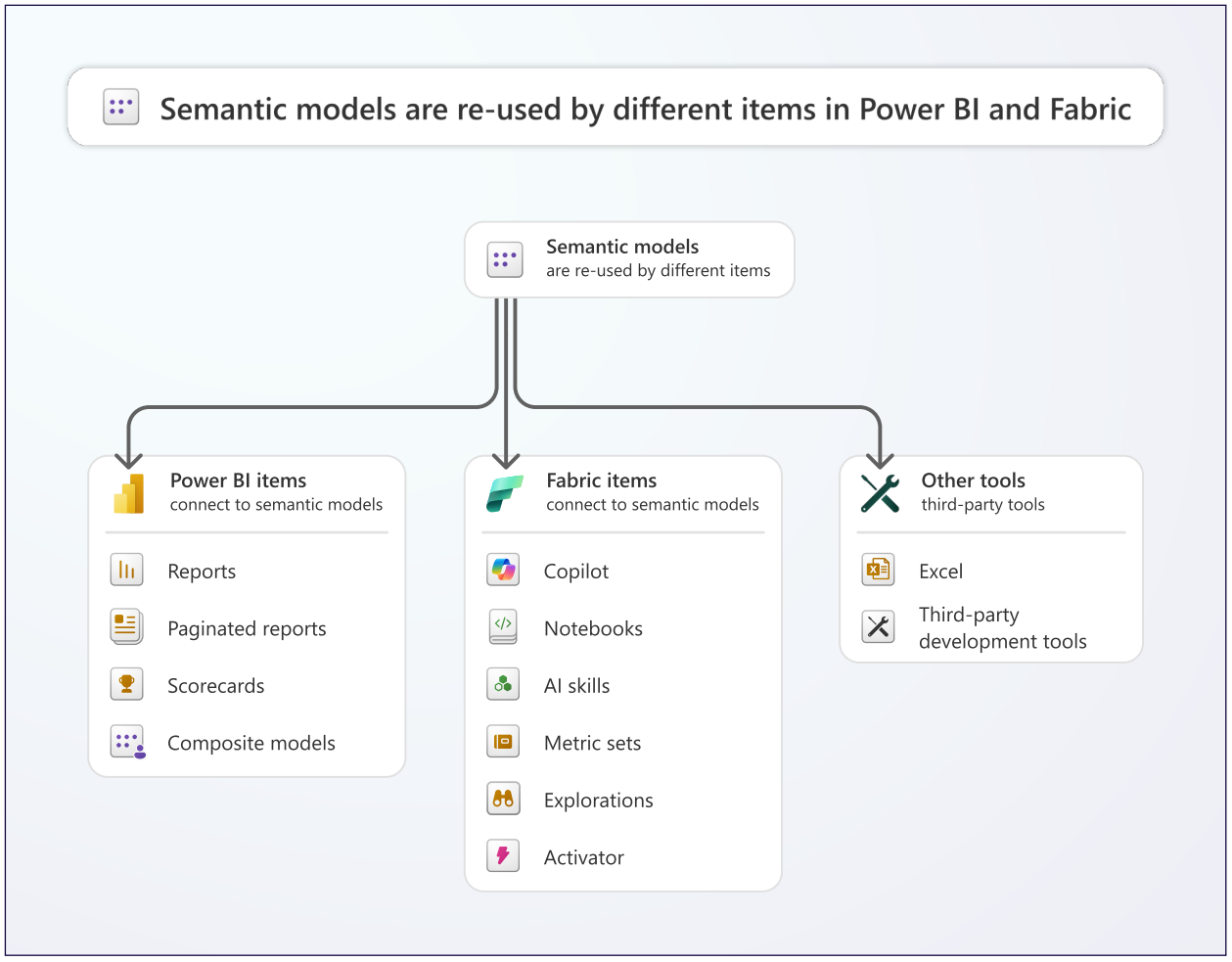
Determine how users will consume the content

In Power BI, there are many different ways in which users can consume content, including semantic models and reports. Depending on how people will access and use this content, you will need to make different decisions about its design and how you will develop it. Some of these decisions can drastically alter both how you make the content, and how much time and effort it takes. Typically, the more different ways that people will use your models and reports, the more considerations you keep in mind, and the higher the resulting development cost.

Use and re-use of semantic models

Semantic models are debatably the most important item types in Power BI and Fabric, since users can consume them by using many different downstream approaches and items. If you will only distribute reports, then you will make very different design

decisions as when users will connect various items to the model to do their own analyses or accomplish their own goals.



You can re-use a shared semantic model by using the following downstream item types:

- Power BI
  - Reports
  - Paginated reports
  - Scorecards
  - Excel (analyze in Excel pivot tables)
  - Semantic models (composite models)
- Microsoft Fabric
  - Activator (formerly Reflex)
  - AI skills
  - Copilot
  - Explorations
  - Metric sets
  - Notebooks (via the Semantic Link and semantic-link-labs libraries)

The following sections give an overview of important considerations when you use semantic models with some of these items.

> [! NOTE] Many of these options are only available if you enable the relevant tenant settings. Ensure that you familiarize yourself with the tenant settings to know what consumption methods are possible in the workspace where you'll publish your semantic model. If these options are limited to specific security groups, then ensure you identify whether your users belong (or should belong) to these security groups, or not.

Reports

There are several different ways in which users can engage with semantic models via reports:

- **Viewing reports.** The standard scenario, in which a user views data in a report that you distribute or share with them.
- **Connecting to the semantic model and creating a new report.** With build permissions, users can create a new report in Power BI Desktop or in the Power BI service. This report has a live connection to the shared semantic model. Users can also convert the live connected report to a new composite semantic model that queries the original by using DirectQuery.
- **Creating an exploration from an existing report visual.** With build permissions, users can also select a supported visual to create an exploration of its data. This creates a new exploration item which allows the user to add fields or change formatting. Users can save and share the resulting exploration if they meet the required criteria for licensing, workspace membership, and item permissions.
- **Personalizing report visuals, in which users can change field and formatting.** Personalize visuals works similar to an exploration, but it only requires read permissions, and doesn't create a new item. Personalize visuals also use any perspectives that a user applies to a report page, which limits the available fields that a user can see and use.

These various scenarios create a number of considerations that you must keep in mind for your semantic models, such as:

- Which permissions you'll grant to users, and how you'll manage these permissions. We recommend that users complete a training before getting build permissions.
- Whether you'll add perspectives to your model, or not. You can only add perspectives by using the TMDL view in Power BI Desktop, or by using other, third-party tools. We recommend that you add perspectives when users will use personalize visuals or composite models.
- Which fields you should hide in your model, and whether you need to enable the *IsPrivate* TOM property of tables to avoid that they or their child objects appear as suggestions. Note that the *Private* TOM property can only be set by using external

tools on a model metadata file (like .bim or .tmdl) or a remote model that is published in the Power BI service.
- How you'll document your model, and what that documentation should entail. We recommend that you make documentation tailored to specific use-cases and that you include relevant and useful information, and not exports of model metadata like DAX measure expressions, which are generally not useful for end-users.
- How you'll advise users to choose one approach over another.

> [! CAUTION] Once you grant read or build permissions to a semantic model, then users might be able to query any table or column in your model by using the approaches described in this section. This is true even if you don't expose that table, measure, or column in a report. Ensure that you always protect sensitive data by using data security, or exclude it from your model. That way, you can avoid unintentionally exposing sensitive information to the wrong people.

Excel (analyze in Excel pivot tables)

If users have build permissions to a model, then they can also connect to a semantic model from Excel and query it by using MDX from an Excel pivot table. This can be useful when users prefer to work in Excel to explore or analyze data, themselves.

When users will use analyze in Excel to query your semantic model, you might need to consider things like the following:

- Certain features like field parameters or measure dynamic format strings only work in Power BI, and not in Excel. To achieve a similar result, you need to use other approaches.
- Analyze in Excel requires that the column property *IsAvailableinMDX* is enabled. If users won't use Excel, then disabling this property might result in smaller and more performant models.
- Users can't view hidden columns or measures in the semantic model, like they can from Power BI Desktop (by right-clicking the data pane, and selecting "view hidden").
- Users can't create their own measures or visual calculations in Excel, like they can when using a live connection from Power BI Desktop. However, they can reference the pivot table fields in other Excel cells and worksheets for custom calculations.
- Users of analyze in Excel often require additional training about how to use it. This is particularly true if they expect an export-like experience or they express an intent to save offline copies of the data. Consider training users about things such as:
  - How to refresh the data.
  - Filter the data *before* adding fields to the pivot table.

    - Avoiding large and detailed queries without filters.
    - Power BI Desktop will be more performant than Analyze in Excel, because analyze in Excel queries the model by using MDX, but Power BI Desktop queries the model with DAX.
    - How to avoid creating governance or security risks while still leveraging Excel the way they prefer.

Copilot and AI skills

If users have read permissions to a model, then can use natural language to explore and ask questions about their data by using Copilot. Alternatively, they can also ask data questions about their data in an AI skill, but unlike Copilot, you must first create and share this item with them.

When users want to interact with your semantic model by using natural language, there are considerable consequences for the design and implementation of your model:

- **Linguistic schema:** You'll have to add synonyms and relationships to your model so that the appropriate English words and terms are associated with the correct model objects.
- **Naming conventions:** You'll have to use AI-friendly, English naming conventions. This means that you should avoid duplicate or overly-similar field names, acronyms, symbols, and abbreviations, even if they are standard in your organization.
- **Properties:** You'll have to set properties like column or measure descriptions, data categories, and others so that AI tools can better recognize and use these fields.
- **Hiding fields:** You'll have to hide fields that you don't want to expose to Copilot or use in its responses.

You also need to spend additional effort training users:

- What Copilot or AI skills can and can't do.
- When to use Copilot or AI skills to explore data over another (non-AI) approach.
- How to write effective prompts.
- How to validate outputs.
- How to troubleshoot unexpected outputs.

> [! TIP] See the following articles for additional, detailed tips and considerations about optimizing models to work well with Copilot:
>
> - Optimization for report authors and model consumers
> - Update your data model to work well with Copilot for Power BI
> - Write DAX queries with Copilot

Copilot and other generative AI technology have important limitations and considerations that you should also understand in the planning and design stage of your content:

- It is non-deterministic, which means that the same input on the same context might produce a different output.
- You can get low-quality or inaccurate outputs, such as when the tool hallucinates, or overstates the importance of certain facts. Copilot might also be incorrect or inaccurate by omission, where it leaves out essential information.
- The tools are limited by their training data, so questions about more novel information are less likely to produce useful outputs.

> [! CAUTION] You should mitigate the risks of these limitations and considerations. Copilot, AI skills, LLMs, and generative AI are nascent technology, so you should not use them for autonomous, high-risk, or business-critical processes and decision-making. For more information, see also Security guidance for Large Language Models.

Paginated reports

You can write a DAX query for a semantic model to create a paginated report with the query result. This is relevant if users require paginated reports and you intend to use your semantic model as the data source.

When you plan to create paginated reports on a semantic model, you might need to consider the following points:

- How you will write and maintain the DAX queries, such as by using the DAX query view of Power BI Desktop, or other third-party tools.
- Whether you need to make design decisions to ensure your semantic model is performant when queried, such as materializing certain data in a model column or table.
- Whether you will embed paginated reports into your interactive Power BI reports or not by using the paginated report visual.
- How you will ensure that there is not redundancy between your paginated reports and other reports.
- In which format you should distribute the paginated reports, and whether you require sensitivity labels or data loss prevention to protect those outputs from becoming a governance or security risk.

Others

There are also other ways to consume semantic models. Some examples are below:

- **Activator (formerly Reflex):** You can use a semantic model to automate data alerts and trigger downstream flows, such as those you create by using Power Automate.
- **Metric sets:** You can create a metric set, which includes measures and recommended dimensions from multiple semantic models in one place. Metric sets can improve data discovery for users.
- **Explorations:** Aside from creating explorations from reports and Copilot outputs, users can also create explorations from a semantic model.

Distribution and sharing of Reports

Depending on how you will distribute and share your Power BI reports, you will make different design choices. For instance, cross-report drillthrough is a feature that lets users navigate between reports in the same workspace or app. However, you can't use cross-report drillthrough if you embed a report or share it publicly via Publish-to-Web.

Other items

Depending on how people will use your content and the underlying data, you might need to turn to other item types in Power BI or Fabric. For instance, if users want to add their own data and calculations, they could use composite models with your semantic model. Alternatively, you might make other centralized data items for them to consume in new models, such as a dataflow.

Once you determine how users will consume the content you make, you should next decide how content creators should collaborate during development.

Decide how content creators should collaborate

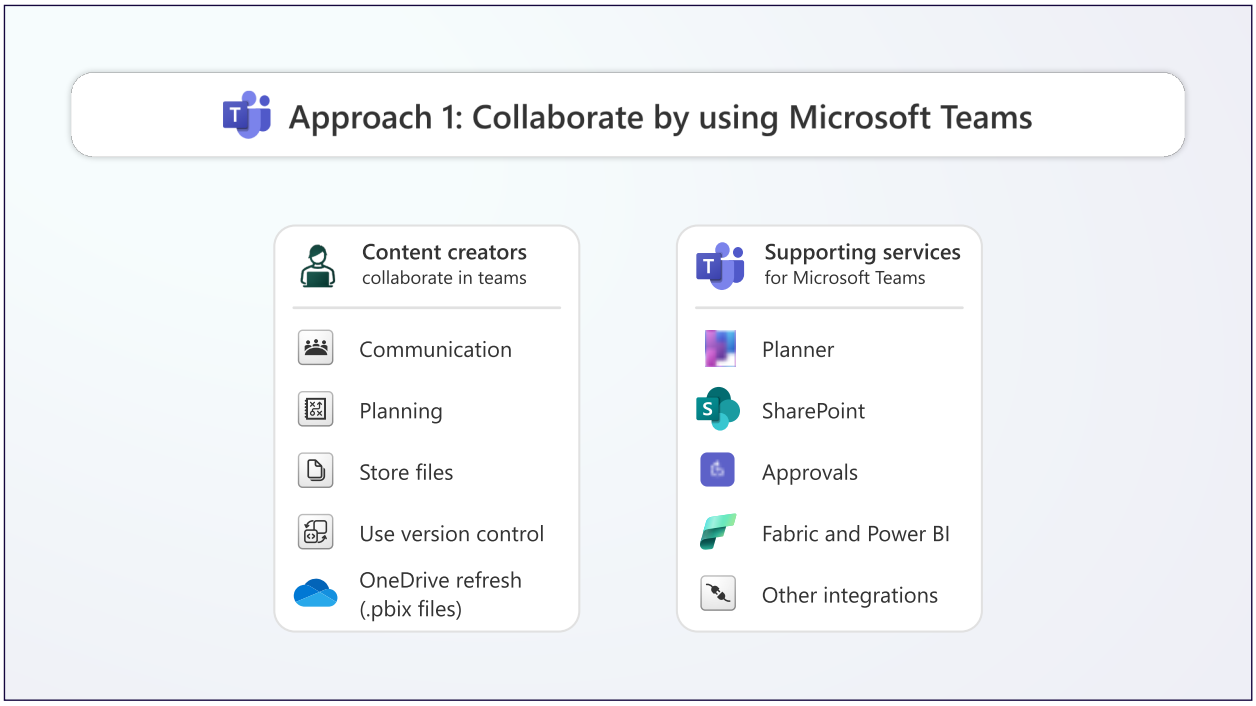
As a solution increases in scope and complexity, it might become necessary for multiple content creators and owners to work in collaboration. When creating complex solutions, we recommend that you use effective tools that help structure, manage, and support collaboration. There are many ways to collaborate when producing Power BI content, such as by using Microsoft Teams, Azure DevOps, or GitHub.

> **Tip**

Even when content creators work independently, they can still benefit from planning and structuring their work by using tools such as Microsoft Teams, Azure DevOps, or GitHub. This is especially important when you plan how to deploy content, such as by using **OneDrive Refresh** from a Microsoft Teams documents library or **Git integration** from an Azure DevOps or GitHub repo.

Microsoft Teams

For smaller or simpler projects, content creators can collaborate by using Microsoft Teams.



By using Microsoft Teams, content creators structure their communication, planning, and work in teams and channels. Microsoft Teams is often a good choice for simpler collaboration scenarios. For instance, decentralized teams that produce content for a limited audience can use document libraries for storing files and version control. They can also make use of other integrated tools and services.

Tip

We recommend that you use Microsoft Teams to facilitate effective content lifecycle management in **self-service scenarios** with decentralized content delivery.

To collaborate and communicate in Microsoft Teams, you use supporting services throughout the lifecycle of your Power BI content.

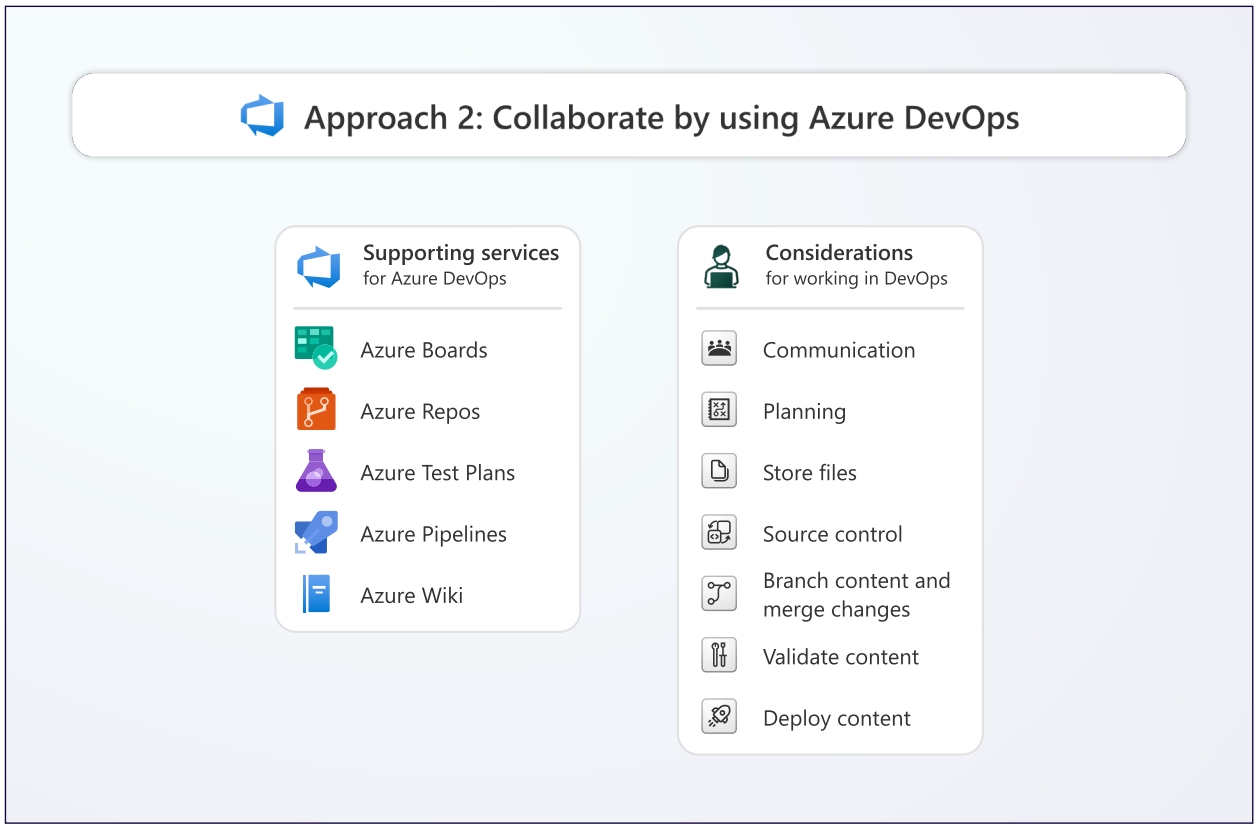
- **Planner**: Content owners can use Planner to create plans, which they use to track tasks and scope content work. Tasks can describe issues, bugs, or features in the solution, and the corresponding stakeholders.
- **SharePoint**: Content creators can store and manage files in a Microsoft Teams document library or connected site for each channel. Content files stored in SharePoint can use version control to help track and manage content changes. For more information about tracking and managing changes by using SharePoint, see Stage 2: Develop content and manage changes.
- **Approvals**: Content creators and owners can set up and use workflows to approve content changes or releases after review.
- **Fabric and Power BI**: Content creators and owners can access the Fabric portal from within Microsoft Teams. From there, they can manage or discuss content, and add helpful reports to tabs in Teams channels.
- **Other integrations**: Content creators can make use of other Microsoft or third-party services that integrate with Microsoft Teams to best fit their preferred workflow and needs.

We recommend that you define a structured process for how content creators should use Microsoft Teams to collaborate. Ensure that you determine:

- How to manage access to teams and channels.
- Who's responsible for managing teams and channels.
- How work is scoped and organized into distinct teams, channels, and plans.
- How content creators should use a document library to organize files and track and manage changes. For instance, how to organize the document library and whether content creators should check in and check out files.
- Whether content creators should use OneDrive Refresh to automatically publish Power BI Desktop (.pbix) files.
- How file sync conflicts are resolved.
- When to archive and remove files from a document library that are no longer relevant.

Azure DevOps or GitHub Enterprise

Content creators and owners can also communicate and collaborate in a central, organized hub by using Azure DevOps or GitHub Enterprise.



> **Note**
>
> Azure DevOps is a suite of services that integrate with Power BI and Fabric to help you plan and orchestrate content lifecycle management. When you use Azure DevOps this way, you typically leverage the following services:
>
> - Azure Repos: Allows you to create and use a remote Git repository, which is a remote storage location you use to track and manage content changes.
> - Azure Pipelines: Allows you to create and use a set of automated tasks to handle, test, and deploy content from a remote repository to a workspace.
> - Azure Test Plans: Allows you to design tests to validate the solution and automate quality control together with Azure Pipelines.
> - Azure Boards: Allows you to use boards to track tasks and plans as work items, and link or refer to work items from other Azure DevOps services.
> - Azure Wiki: Allows you to share information with their team to understand and contribute to content.

By using Azure DevOps or GitHub Enterprise, content creators use projects to structure their communication, planning, and work. Additionally, content creators can orchestrate content lifecycle management from within Azure DevOps by performing *source control*, validation, and deployment. Source control is the process of managing more granular changes to content code and metadata.

Azure DevOps is often a good choice for more advanced collaboration scenarios, because there are supporting services and options to orchestrate content creation and deployment.

> **Tip**
>
> We recommend that you use Azure DevOps to aid in effective content lifecycle management in **enterprise scenarios** with centralized content delivery. Collaborating by using Azure DevOps or similar tools is preferred in larger or more complex scenarios over collaborating by using Microsoft Teams or SharePoint. That's because there are more tools and options available to facilitate more robust collaboration and automation.

We recommend that you define a structured process for how content creators should use Azure DevOps to collaborate. Ensure that you determine:

- How work is scoped and how content branches are created, named, and used.
- How authors group and commit changes and describe them with commit messages.
- Who's responsible for reviewing and approving changes by using pull requests.
- How to pull request merge conflicts are resolved and who resolves them.
- How changes made in different branches should be merged into a single branch.
- How content is tested and who performs testing before content is deployed.
- How and when changes are deployed to development, test, and production workspaces.
- How and when deployed changes or versions of the solution can be rolled back.

GitHub

Content creators and owners can also communicate and collaborate by using GitHub (cloud versions only) and GitHub Enterprise. Similar to Azure DevOps, GitHub provides a platform with services that you can use to help orchestrate and manage aspects of your Power BI or Fabric environment.

The main difference between Azure DevOps and GitHub is that while Azure DevOps provides a suite of tools for managing the software development lifecycle, GitHub focuses mainly on hosting Git repositories, source control, and collaboration on code. You mainly use GitHub when you plan to deploy content by using Git integration. Additionally, you can use GitHub to sync content from a public, open-source repository to a workspace.

To use Git integration with GitHub, you have to enable the admin tenant setting *Users can sync workspace items with GitHub repositories*.

> **Note**
>
> You can also use Microsoft Teams together with Azure DevOps or GitHub because there are different ways to integrate these services. For instance, you can view and manage Azure Boards and monitor events in Azure Pipelines from within Microsoft Teams.
>
> You can also collaborate and plan content by using other tools that aren't mentioned, here. What's most important is that you use tools and services that facilitate collaboration for you, and that best fit your team's needs and the way they work.

When you've decided if and how content creators should collaborate, you should next decide where you'll store your files. Many of these files will be stored where you choose to collaborate.

Decide where to store files

When creating content, you typically produce different types of files. It's important to decide where to store these files so that you can effectively manage them.

> **Tip**
>
> Store files where they can be accessed by multiple team members, and where changes can be easily tracked (known as *version control*). This approach ensures that the departure of a team member or loss of a file doesn't lead to disruption.

The types of files you'll need to store often include:

- **Content files**: Files that contain the content data or metadata. Content files with data like .pbix and Power BI Project (.pbip) files contain sensitive information. Store content files in a secure location accessible only by those who need access to them. Also, you should store content files in a location that supports version control, such as a document library in Microsoft Teams or a Git repository in Azure DevOps. Examples of content files include:
  - Power BI Desktop (.pbix) files
  - Power BI Project (.pbip) files
  - Power BI paginated report (.rdl) files

    - Model metadata (.bim or .tmdl) files
    - Dataflow metadata (.json) files
- **Data source files**: Files that are consumed by data items like semantic models or dataflows. Content is directly dependent on data source files, so it's important to carefully consider where they're stored because removing them will result in data refresh failure. Additionally, these files might contain sensitive information. So, store data source files in a secure, reliable, and trustworthy environment that has limited access by other individuals. Examples of data source files might include:
    - Structured data sources, like Excel workbooks, Parquet, or CSV files.
    - Semi-structured data sources, like JSON or XML files.
    - Unstructured data sources, like images that you import into reports.
- **Supporting files**: Files that support content creation or management, but aren't required for it to function. Supporting files should be stored in a location that supports version control, and where other tools and content creators can access them. Examples of supporting files might include:
    - Power BI theme (.json) files.
    - Image (.png, .jpg, or .gif) files in Power BI reports.
    - Best Practice Analyzer Rules (.json) files.
    - Source code files for content and queries.
    - Custom visualization (.pbiviz) files.
- **Templates and documentation**: Files that aid in the creation of self-service content or describe existing content. Templates and documentation should be easily accessible by the people who need to use them. Examples of templates and documentation might include:
    - Power BI template (.pbit) files.
    - Visualization templates and example reports.
    - Solution designs and documentation.
    - Solution planning and roadmaps.
    - User requests and solution issues.

> **⊗ Caution**
>
> Some content files like .pbix and .pbip files can contain sensitive data imported from data sources. Additionally, metadata files can also contain sensitive information, or in some cases, data points. An example of this is report metadata and .pbit templates, which can contain column values in certain circumstances. Ensure that you take the necessary precautions to store these files in secure locations and that you practice effective data loss prevention.

You have different options to store files. Ensure that you select the appropriate location, depending on the type of file, its content, and how it will be used.

SharePoint Online or OneDrive

A common solution for storing files is by using SharePoint sites. SharePoint is widely accessible for most users and highly integrated with both Power BI and other Microsoft 365 applications, like Microsoft Teams. Additionally, it has built-in version control, making it convenient for storing most file types. Version control allows you to view and manage different saved versions of a file.

When you store files in SharePoint, consider the following points.

- **Organization**: Ensure that you maintain a consistent and logical structure so it's straightforward to find specific files. Use good naming conventions, organize files in folders, and archive files that are no longer relevant for ongoing projects.
- **OneDrive refresh**: You can link a published semantic model or report to a .pbix file that's stored in a SharePoint or OneDrive for Business (also known as OneDrive for work or school) site. With this approach, you no longer have to publish the semantic model to bring changes into effect. Instead, your changes are visible after an automatic OneDrive refresh, which occurs hourly. While convenient, be aware that this approach comes with some caveats and challenges. When things go, it can't be easily reversed.
- **Preview reports**: In SharePoint, it's possible to view Power BI reports without having to install Power BI Desktop or download the .pbix file locally. When you open reports in this way, they're displayed in the browser. This capability can be a convenient alternative to viewing reports from the Fabric portal. It's enabled by default in the Fabric tenant settings.

> **Tip**
>
> When you collaborate by using Microsoft Teams, consider storing files in the channel document library. This approach helps centralize files and facilitates collaboration.

Consider storing the following file types in SharePoint.

- **Templates and documentation**: Store templates and documentation in SharePoint when you don't have an existing storage solution. SharePoint is ideal for these files because you can grant access to others and manage files without complex setup or processes.

- **Supporting files**: Store supporting files in SharePoint when you don't have an existing storage solution. However, some supporting files (like Power BI theme .json files for reports) might be better stored in a version control system that allows viewing and managing saved changes.
- **Content files**: Store content in SharePoint when it isn't critical to the business, or when you don't have access to a remote repository like Azure Repos.
- **Data sources**: Store data sources in SharePoint only when they're small in size and complexity. Exercise discipline when using SharePoint to store data source files. Consider other possible alternatives, like OneLake.

> ⊗ **Caution**
>
> Don't use SharePoint as an alternative to a proper data architecture. While storing data source files in SharePoint can be convenient in some limited scenarios, this approach doesn't scale when you have larger, more complex data sources, or when you need lower data latency.

> ⚠ **Warning**
>
> Don't use personal file systems or personal OneDrive accounts to store files. If the owner leaves the organization, these files will no longer be available.

OneLake

If you have a Fabric capacity, OneLake can be a good choice to store data source files. You can upload or synchronize files to OneLake by using OneLake File Explorer, where they can be transformed to tables for use in downstream workloads like Power BI. For larger or regularly updated data sources, you can load files to OneLake automatically by using Fabric Data Factory or other applications that use the Azure Data Lake Storage (ADLS) Gen2 API or the Azure Storage Python SDK.

> ⊗ **Caution**
>
> Actions like uploading or downloading files from OneLake **consume Fabric capacity units**. You should **monitor capacity metrics** and take steps to avoid capacity strain caused by unnecessary movement of large files.

Additionally, files accessed by users with OneLake File Explorer are vulnerable to accidental changes or loss. We recommend that you avoid using OneLake File Explorer for business-critical solutions.

> ⚠ **Warning**
>
> OneLake File Explorer has several important limitations and considerations. For instance, OneLake doesn't support version control for files, like SharePoint or OneDrive. Take into account these considerations and limitations when you decide where to store files.

> **Tip**
>
> When storing data in OneLake, consider enabling Business Continuity and Disaster Recovery (BCDR) to mitigate risk of data loss. With BCDR enabled, your data is duplicated and stored in two different geographic regions, according to Azure's standard region pairings.

Remote repository

Content creators can commit and save work from their local machine to a *remote repository*—such as an Azure Repos or GitHub Git repository—at regular intervals during development. A remote repository contains the latest version of content or supporting files, and it's accessible by the entire development team. Typically, a remote repository facilitates more advanced lifecycle management approaches than using Teams, SharePoint, or OneDrive. That's because by using a remote repository, content creators can benefit from more sophisticated options to collaborate on files, or track and manage file changes. For instance, content creators can work on their own branch of the remote repository to make changes, and request to merge those changes into the main branch when ready.

Consider storing the following file types in a remote repository.

- **Templates and documentation**: Store templates and documentation in a remote repository when you manage the project with related services like Azure DevOps.
- **Supporting files**: Store supporting files in a remote repository when one is available to easily track and manage changes.
- **Content files**: Store content in a remote repository when it's critical to the business, or you intend to collaborate with other developers on the same content.

A remote repository is ideal for tracking content changes and facilitating collaboration.

> **Tip**
>
> When you use a remote repository, we strongly recommend that you store Power BI reports and semantic models as **Power BI Desktop projects (.pbip) files** instead of .pbix files. That's because saved changes can't be identified in a .pbix file.
>
> Also, consider using the **Power BI enhanced report format (PBIR)** for reports. The PBIR format ensures that report metadata is easier to read, which makes tracking and managing changes in source control easier. Furthermore, reports that use this format can be more easily managed by programmatic tools such as **semantic-link-labs**, a Python library from Microsoft that you can use in Fabric notebooks.

No files: Content created in the Fabric portal

Content creators can author content directly in the Fabric portal. In this scenario, they don't typically work directly with content files. You should typically author content in the Fabric portal only when the item types can't be created elsewhere (like dataflows, dashboards, or scorecards). You can also author reports and semantic models in the Fabric portal when don't have access to a Windows machine, and therefore can't use Power BI Desktop. For more information, see User tools and devices.

> **Caution**
>
> You can't download as a file some content that's created in the Fabric portal. For instance, reports created in the Fabric portal can't be downloaded as .pbix files.
>
> When authoring content in the Fabric portal, you should instead use the **Fabric APIs** or **Git integration** to back up **content definitions**. When you back up content definitions, you mitigate disruption if that content gets accidentally deleted or unintentionally changed. If content is accidentally deleted or changed, you can replace it by using the backup.

Checklist - When planning and designing content, key decisions and actions include:

- ✓ **Conduct solution planning**: Gather business requirements and technical requirements to sufficiently understand the problem your content will address, and to design how this content will address the problem.
- ✓ **Identify who will create the content**: Depending on the workflow, skills, and needs of the individual content creator, different approaches to lifecycle management might be needed.
- ✓ **Identify whether multiple content creators need to collaborate**: Ensure that collaborating content creators are using file types that support version control, like .pbip files.
- ✓ **Decide how content creators will collaborate**: Decide how sophisticated the collaboration will be. Additionally, decide how you'll facilitate this collaboration, such as by using Microsoft Teams, Azure DevOps, or GitHub.
- ✓ **Decide the format of content**: Decide whether Power BI semantic models and reports will consist of .pbix or .pbip files (or other formats like .bim or .tmdl for models), and whether reports will use the PBIR format or not.
- ✓ **Determine how content will be consumed**: Evaluate how users will engage with data. Decide which item types you need to create to support consumption, and whether users require read permissions only, or also build permissions to underlying semantic models or other data items. If users do require build permissions, determine early-on how they will consume semantic models, and how you will train them to do this, effectively.
- ✓ **Set up collaboration tools**: Ensure that you perform the necessary first-time setup for the solution or project. Make key decisions about how you'll manage collaboration by using these tools.
- ✓ **Store data source files in SharePoint or OneLake**: Store small, simple data source files in SharePoint. Otherwise, use OneLake or ADLSGen2 (if they're available) instead.
- ✓ **Store content and supporting files in SharePoint, Microsoft Teams, or a Git repository**: For simpler, smaller projects, use SharePoint for most files if it's organized and you practice good access management. For larger environments or when parallel collaboration is required, consider using a Git repository in GitHub or Azure DevOps, which will provide detailed visibility of content changes.
- ✓ **Store templates and documentation in Microsoft Teams or SharePoint**: These templates and documentation are intended for the user community. Ensure that templates and documentation are easy for others to find, use, and understand.
- ✓ **Plan for development and deployment**: To conclude this first stage, perform specific planning to address key areas and conduct initial setup. For instance, establish tools and test data source connections.

Related content

In the next article in this series, learn how to develop content and manage changes as part of managing the content lifecycle.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Develop content and manage changes

Article • 03/03/2025

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

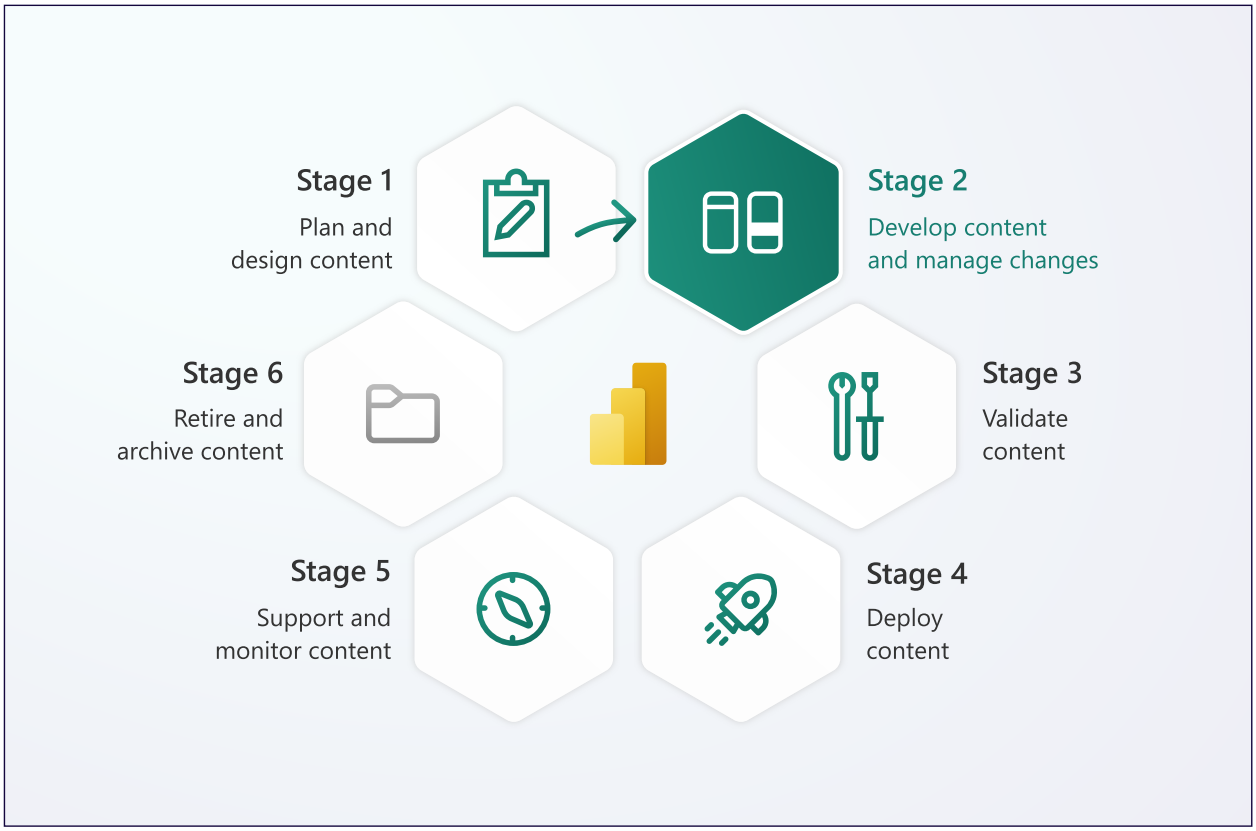
This article helps you to develop content and manage changes as part of managing the content lifecycle. It's primarily targeted at:

- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content. These teams can also include roles like release managers, who handle the lifecycle of content releases, or engineers who create and manage the components needed to effectively use and support lifecycle management.
- **Content creators and content owners**: Users who create content, which they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the Power BI content that they create.

Lifecycle management is the processes and practices you use to handle content from its creation to its eventual retirement. In the first stage of lifecycle management, you plan and design content, which involves solution planning and making key decisions that affect your approach to lifecycle management. In the second stage, you develop content and manage changes.

Managing content changes during its lifecycle is important to ensure effective collaboration among content creators and stable and consistent delivery of content to consumers.

The following image depicts the lifecycle of Power BI content, highlighting stage two, where you develop content and manage changes.



> **Note**
>
> For an overview of content lifecycle management, see the first article in this series.

> **Tip**
>
> This article focuses on key decisions and considerations to help you develop content and manage changes throughout its lifecycle. For more guidance about how to develop content and manage changes, see:
>
> - What is lifecycle management in Microsoft Fabric?: This article provides a technical introduction and tutorial to Fabric Git integration and deployment pipelines.
> - Lifecycle management best practices: This article contains practical tips and guidance for using the lifecycle management features of Fabric and Power BI to develop content and manage changes.
> - Power BI Desktop OneDrive and SharePoint integration: This article contains an overview of the options to use and store files saved to OneDrive for Work and School or SharePoint when you perform version control with .pbix files.

- Get started with Git in Azure Repos: This series of articles contain practical tips, tutorials, and guidance for performing source control by using a Git repository in Azure Repos.

Content creators and owners should manage content changes by using *version control*. Version control is the practice of managing changes to files or code in a central repository. This practice facilitates more effective collaboration and content management.

Other benefits of version control include the ability to:

- Merge changes from multiple content creators, and handle merge conflicts.
- Identify which content changed, and what changed in that content.
- Link content changes to specific work items.
- Group changes into distinct releases with version history.
- Roll back changes or entire versions of content.

Tip

We recommend that you use version control for all content that you create, where possible. Using version control has significant benefits both for content creators and consumers and reduces the risk of disruption due to file loss or the inability to roll back changes.

The first step to set up version control is to decide how you'll develop content.

Decide how to develop content

Depending on how you author content, you'll make different decisions about how to manage it. For instance, for Power BI reports and semantic models, a Power BI Desktop (.pbix) file has fewer options for version control compared to the Power BI Desktop project (.pbip) format. That's because a .pbix file is a binary format, whereas the .pbip format contains text-based human-readable content and metadata. Having human-readable content and metadata allows for easier tracking of model and report changes by using *source control*. Source control is when you view and manage changes *within* content to its code and metadata.

Tip

When developing semantic models and reports by using Power BI Desktop, we recommend that you use .pbip files instead of .pbix files. When developing semantic models by using XMLA tools, we recommend that you use the Tabular Model Definition Language (TMDL) format, instead of .bim files. For more information, see Decide the format of your content later in this article.

Power BI Desktop

You can use Power BI Desktop to create semantic models or reports, which you can save as either .pbix or .pbip files. There are additional custom content files that you also may use when you use Power BI Desktop. When using Power BI Desktop to create content, some key decisions you should make include:

- **Which file format to use**: You can save content either as .pbix or .pbip files. For more information, see Decide the format of your content later in this article.
- **How to manage custom content**: You can add themes, custom visuals, or images to Power BI Desktop files, which may require distinct considerations for lifecycle management. For instance, when content creators make their own custom visuals, they should save and manage the visual definition in a separate file.
- **How to manage preview features**: You can opt-in to preview features or settings in Power BI Desktop, which alters content and how you'll use it. For instance, you might take additional steps to validate content that uses preview features.

Web authoring

Certain content—such as dataflows, dashboards, and scorecards—can only be created in the Fabric portal. You can also create or modify some content—such as semantic models, reports, and paginated reports—in both the Fabric portal or by using local tools. When creating content by using web authoring, some key decisions you should make include:

- **How to manage changes**: You can make changes to many item types by using web authoring, but these changes might be saved instantly, overwriting previous versions. For instance, if you are collaborating with others, you might want to avoid web authoring on shared items, working instead on your own copy.
- **How to retrieve content backups**: You can create content like reports or semantic models by using web authoring, but these items cannot be downloaded to local .pbix files. For instance, you can choose to back up this content by retrieving and storing its metadata.

> **Tip**
>
> When developing dataflows and scorecards, we recommend that you retrieve the item definitions to manage changes and store a backup. You can automate dataflow and scorecard retrieval by using the Fabric REST APIs. Specifically, you can use the Get Dataflow and Get Scorecards endpoints, respectively.

> **Caution**
>
> Some content—like dashboards—doesn't have the option to retrieve a definition. For this content, you can't easily track changes or retrieve a backup.

Other tools

You can use other tools to create or manage certain types of content. These tools may provide added benefits, better suit your workflow, or be required to manage specific features or content types. You can use both other Microsoft tools or third-party tools to create and manage content. Other tools you can use to create or manage content are as follows.

- **Visual Studio or Visual Studio Code**: An integrated development environment for developers to create and manage semantic models or Fabric notebooks. With both Visual Studio and Visual Studio Code, developers can also perform source control management (SCM) by committing and pushing local changes to a remote repository.
- **Tabular Editor**: A third-party tool to develop and manage semantic models.
- **Excel**: A client tool for pivot tables and live connected tables that work with a semantic model.
- **Power BI Report Builder**: A desktop application for creating paginated report (.rdl) files.

When creating content by using other tools, some key decisions you should make include:

- **How to manage licenses**: Other tools may require additional licenses that you should manage.
- **How to publish content**: Other tools may require additional steps to publish content, such as by using XMLA endpoints or the Power BI REST APIs.

Once you decide how you'll create content, you next need to choose which format you'll use to save and manage it.

Decide the format of your content

Depending on what content you'll create and which tools you'll use, your content might use different file formats. However, even within a tool, you must choose between different formats. For instance, for reports and semantic models that you create with Power BI Desktop, you have to choose between .pbix and .pbip files. This decision has a functional impact on not only how you develop content, but also how you manage that content during its lifecycle, and how efficiently you can complete certain development tasks. For instance, you must use .pbip files if you want to develop content in Power BI Desktop but publish that content by using Git integration.

The file format for your content can change during its lifecycle. For instance, you might start creating .pbix files for reports and models, initially. However, when new content creators want to collaborate with you, or when you want to enhance your productivity, you might switch to a different format, like .pbip files.

The following sections give an overview of common formats you must choose from in Power BI.

Power BI Desktop (.pbix) file format

The default and most common format for Power BI reports and models is the .pbix file format. This binary format is easy to manage and use for most people. However, it is not possible to open or use the file contents to track changes or improve developer productivity.

Use the .pbix file format when:

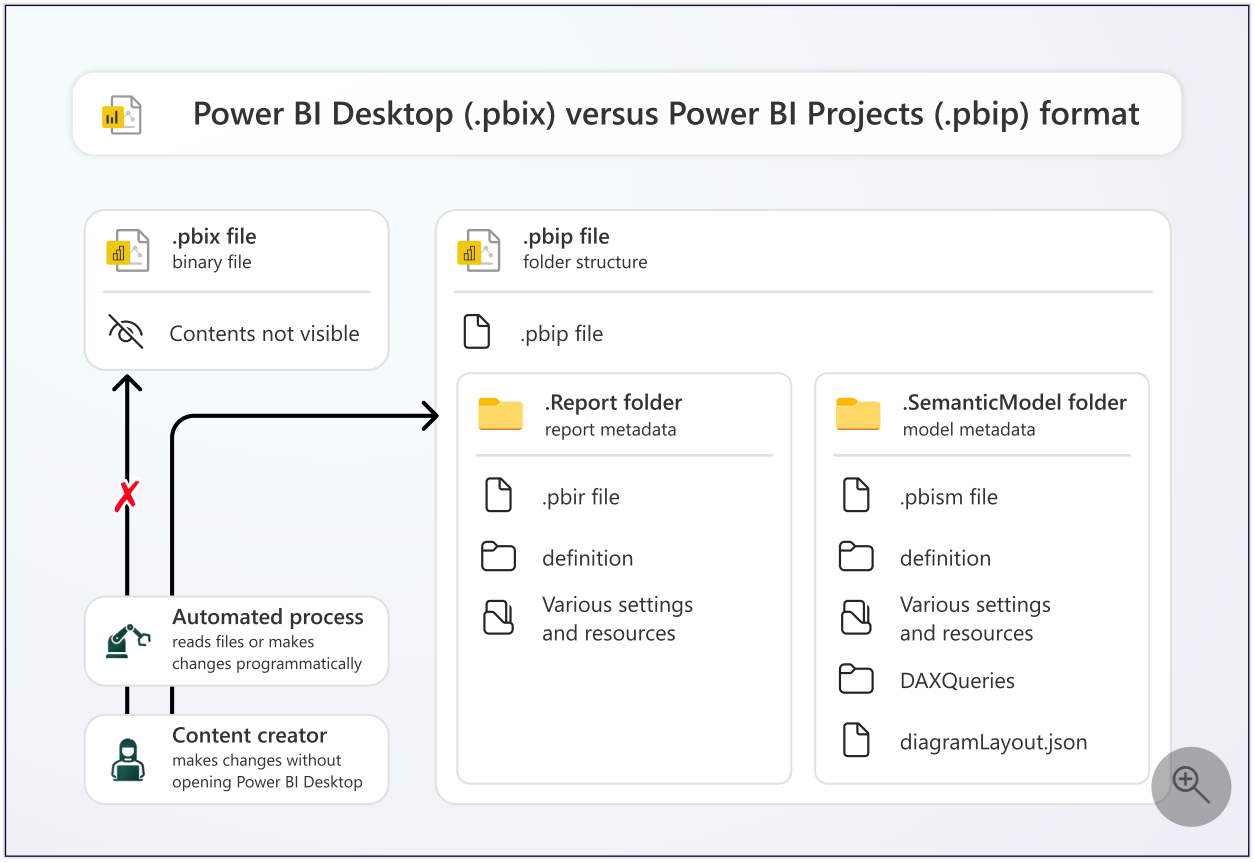
- You plan to publish content by using OneDrive refresh.
- Content creators want to use and share a single file rather than a folder of multiple files (like with the .pbip format).
- Content creators prefer the .pbix format, because they find it simpler.

Power BI Projects (.pbip) file format

Instead of a .pbix file, you can also save content using the Power BI Projects (.pbip) format. This format splits the file into a *folder structure*. In the folder structure of a .pbip, there are multiple metadata files which you can open and read that provide the definitions and configurations for models and reports. Because you can open and read

the file contents, you can make changes to them either manually or programmatically, which can produce significant productivity enhancements. Certain features in Fabric such as Git integration also require that you use the .pbip format instead of the .pbix format if you will develop content in Power BI Desktop.

The following image shows how .pbix files and .pbip files differ:



In summary, users or automated processes can view and modify the contents of a .pbip file without opening it in Power BI Desktop. In contrast, a .pbix file is a binary file, and there aren't supported methods to view or modify its contents. There are various scenarios where you want to be able to view or change these metadata contents, such as:

- You want to conduct source control by tracking and managing changes, with visibility on what has changed.
- You want to make changes in bulk, or search for something in the file.
- You want to re-use elements of the file, like a visual, table, or measure.
- You want to view properties or settings that aren't visible in the user interface of Power BI Desktop.
- You want to automate certain processes, such as scanning metadata for best practice violations before publishing content.

The .pbip file format can also use the TMDL or PBIR formats for semantic model and report definitions, respectively, which have their own advantages and considerations to

keep in mind.

Use the .pbip file format when:

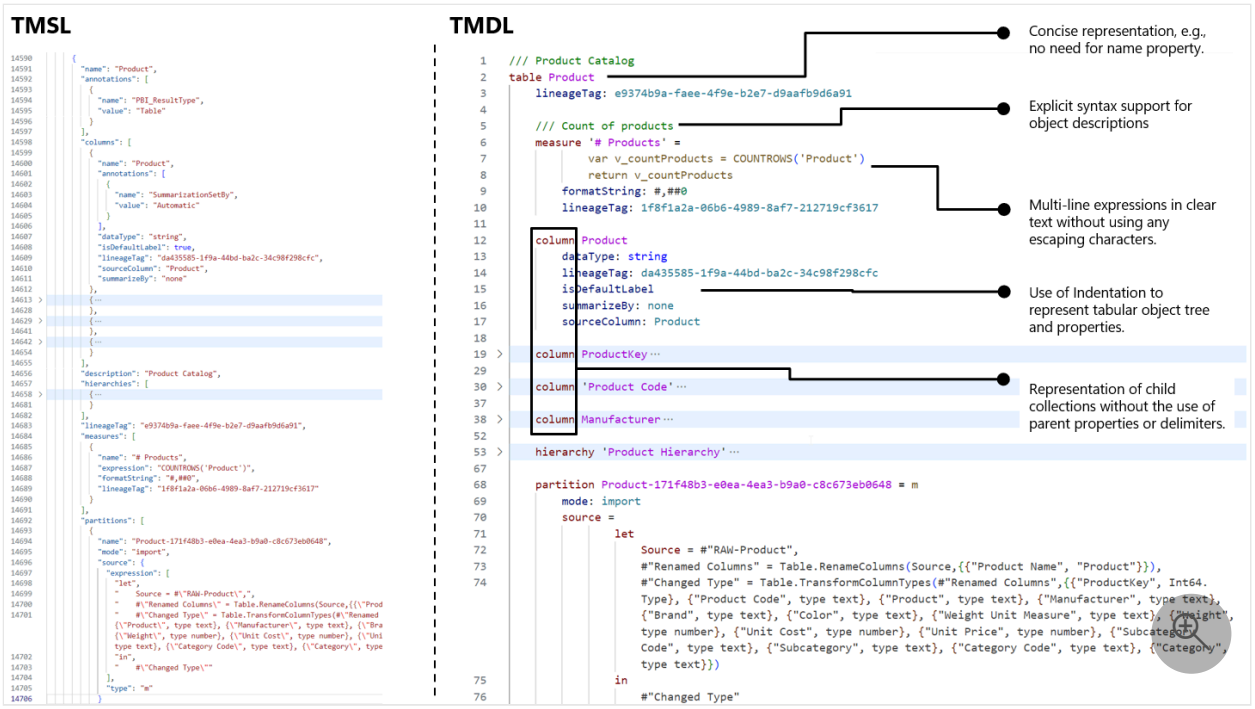
- You plan to use Git integration and not OneDrive refresh to publish or deploy content.
- You plan to use source control to track and manage changes.
- You plan to use Power BI Desktop in combination with other tools to develop models or reports.
- Multiple content creators will collaborate together on a models or reports.
- You want to benefit from productivity enhancements and save time making models or reports.
- You plan to automate parts of the development, testing, or deployment process.
- You want to structure your model and report files in different ways (like multiple reports in the .Report folder, and multiple models in the .SemanticModel folder).

> **Tip**
>
> When you develop content, you should be self-aware of scenarios where you feel like you are wasting time, such as with repetitious tasks. In these scenarios, you can often save time by leveraging these new formats together with other tools, such as Visual Studio Code, notebooks in Fabric, and more.

Tabular Model Definition Language (.tmdl) file format

When you save a model as a .pbip you can choose to save it as a single .bim file which uses the Tabular Model Scripting Language (TMSL), or in a folder structure of multiple .tmdl files, which use the new Tabular Model Definition Language. TMDL uses a YAML-like syntax for semantic model definitions that makes it easier to read and manage changes of a model.

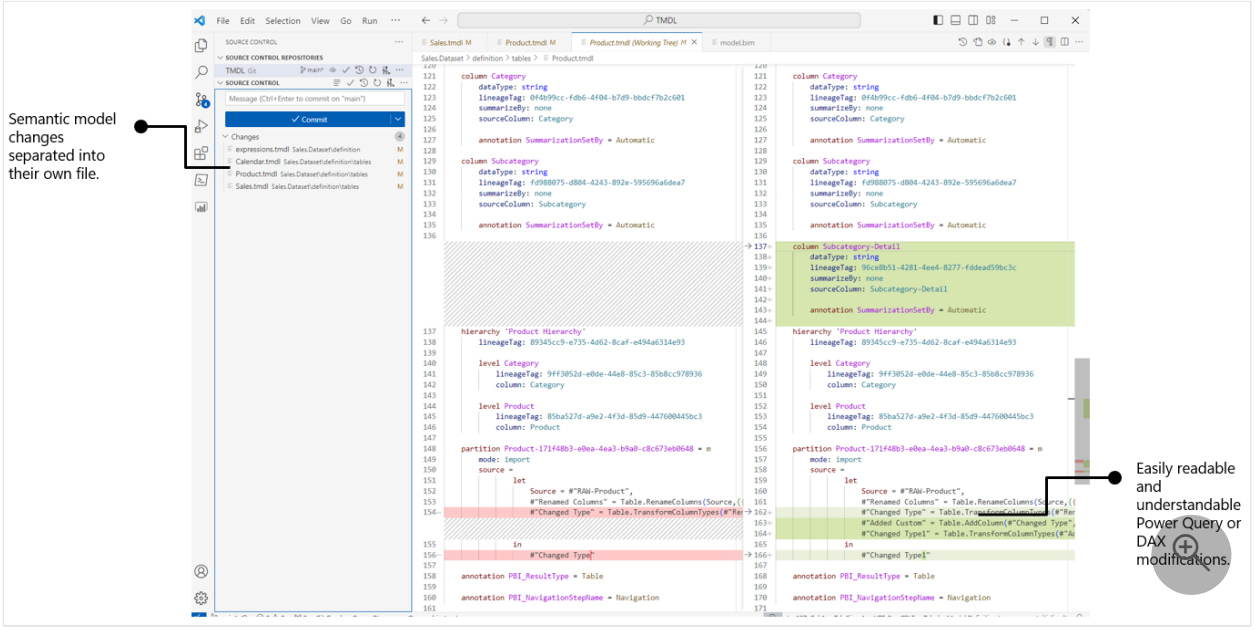
The following is an example of what the TMDL format looks like:



Some advantages of the new TMDL format are as follows:

- You can better use source control and Git integration, because model metadata is more concise, structured, and easier to read.
- You can more easily merge changes without creating conflicts.
- You can improve your efficiency for certain reporting tasks, such as re-using or copying model objects between models (like a date table).

The following is an example of what source control with the TMDL format looks like:



Use the TMDL format when:

- You plan to use source control to track and manage changes.
- You want to enhance developer productivity by making metadata changes.

- You develop your semantic model using other tools, such as Visual Studio Code or Tabular Editor.

> ⓘ Note
>
> TMDL format is different from the TMDL view in Power BI Desktop:
>
> - **TMDL** is a language and metadata format for semantic models.
> - **The TMDL view** is a scripting interface in Power BI Desktop for you to make programmatic changes to the model. It uses a similar YAML-like syntax for TMDL scripts as TMDL format files. You don't need to use the TMDL format to benefit from the TMDL view.

Power BI enhanced report (PBIR) format.

When you save content as a .pbip file, you can choose to use the default report format or the new PBIR format. This new format presents several advantages that can enhance developer productivity and collaboration, particularly when you use the PBIR format with .pbip files.

The following is an example of what the PBIR format looks like:

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/fabric/item/report/definition/visualContainer/1.0.0/schema.json",
  "name": "7c4826a02220bab75601",
  "position": {
    "x": 85.37883561975859,
    "y": 33.754423384555722,
    "z": 0,
    "width": 299.81870182752436,
    "height": 299.81870182752436
  },
  "visual": {
-   "visualType": "card",
+   "visualType": "gauge",
    "query": {
      "queryState": {
-       "Values": {
+       "Y": {
```

Some advantages of using the PBIR format are as follows:

- You can better use source control and Git integration, because report metadata is more concise, structured, and easier to read.
- You can improve your efficiency for certain reporting tasks, such as:
  - Re-using or copying visuals between pages by copying the visual.json.

    - Re-using or copying pages between reports.
    - Finding and replacing custom colours, fields, and other visual configuration all at once.
    - Fixing "broken" visuals with fields that were renamed or moved between tables.
- You can add metadata annotations to report metadata to facilitate certain automations or tasks that support continuous integration / continuous deployment (CI/CD).
- You can leverage tools like semantic-link-labs that have more utility with the new PBIR format.

Use the PBIR format when:

- You plan to use source control to track and manage changes.
- You want to enhance report creator productivity by making metadata changes.
- You want to make programmatic or automatic changes to the report.

> **⊗ Caution**
>
> Ensure that you first check the **limitations and considerations** before using the PBIR format. These limitations and considerations change over time, so check back regularly if there are certain items that prevent you from fulfilling a specific requirement for your Power BI content.
>
> Additionally, all report metadata irrespective of its format has the potential to include data points. For more information, see **PBIR Folder and Files**.

Power BI template (.pbit) file format

You can also save a Power BI report or semantic model as a .pbit file. However, .pbit files are intended for re-use of a specific report layout or model structure. You should not use the .pbit format to save and manage Power BI content during its development. Instead, you should use .pbit files when you want to create reusable templates to share with other people in your organization.

Other formats

When you develop other content in Power BI (such as dashboards, dataflows, or paginated reports), you might not have content files if you develop the item in Fabric. Alternatively, you could only have the item's definition saved to source control. For more information, see Decide where to store files in the previous article of this series.

Decide how you'll set up and use workspaces

When developing content, you should use multiple workspaces (or stages) to separate production content used by the organization from content being developed or validated. There are several advantages to using separate workspaces for your content:

- You can develop and test content without affecting the content that's currently in use. This avoids changes that can cause unintentional disruption to content in production.
- You can use separate resources for developing and testing content, like using separate data gateways or Fabric capacities. This avoids that resources used for development and testing disrupts production workloads, causing slow data refreshes or reports.
- You can create a more structured process to develop, test, and release content by having discrete environments for each of these stages. This helps you to improve productivity.

In Fabric and Power BI, we recommend that you use separate Fabric workspaces, as described in the tenant-level and workspace-level planning articles.

> **Important**
>
> Using separate environments is an essential step to ensure the success of your content lifecycle management approach.

There's multiple ways to use Fabric workspaces to separate environments. Typically, in addition to your local environment, you use two or more workspaces to manage content during its lifecycle.

Answer the following questions as you plan how you'll use separate workspaces throughout the lifecycle of this content:

- How many workspaces do you need?
- Will you separate workspaces by item type?
- Who will have access to each workspace?
- Will the workspaces belong to a Fabric deployment pipeline, or will you orchestrate deployment by using other approaches, such as by using Azure Pipelines?
- How will you manage cross-workspace publishing? For instance, do you need to ensure that reports in a test workspace for reports connect to semantic models in a separate test workspace for data items?
- Do you need to also use separate supporting resources for items in production workspaces, such as a separate on-premises data gateway cluster?

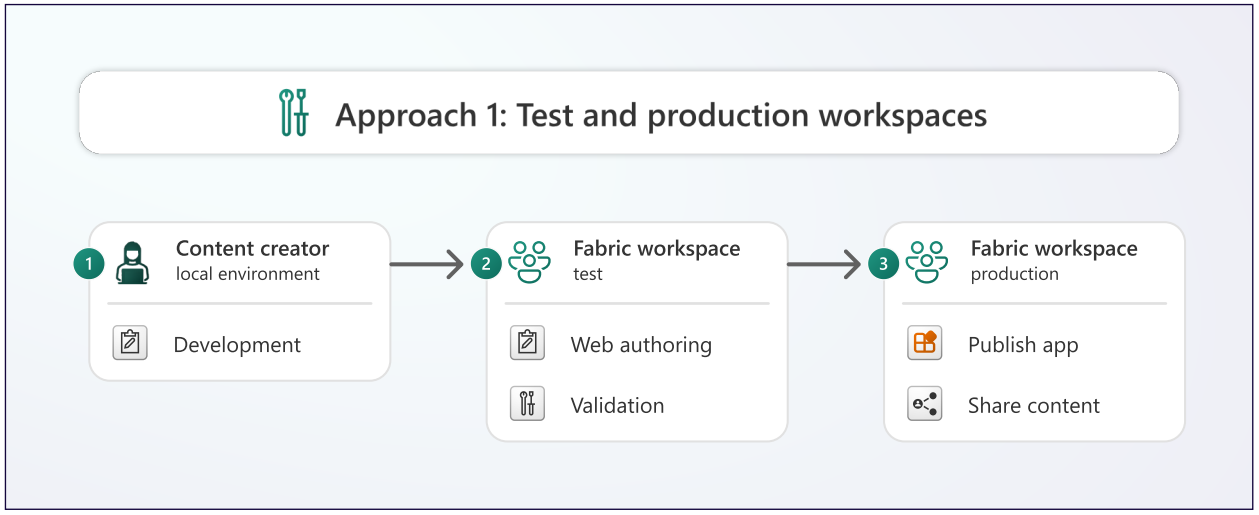
The following sections provide a high-level overview of the different approaches you can take to separate workspaces to facilitate lifecycle management.

> **Note**
>
> The following sections focus on how you can set up and use separate workspaces. You can deploy content to these workspaces by using one of the following four approaches:
>
> - Publishing from Power BI Desktop.
> - Deploying content from another workspace by using Fabric deployment pipelines.
> - Syncing content from a remote Git repository by using Git integration.
> - Deploying content programmatically by using the Fabric REST APIs, Power BI REST APIs, or XMLA endpoints.
>
> For more information about these approaches to deploy content, see Stage 4: Deploy content later in this series.

Test and production workspaces

When you deliver simpler content that's not critical for the organization, two workspaces can often suffice. In this scenario, content creators typically have limited collaboration on the same items, and develop content locally.

The following diagram depicts a high-level example of how you might use separate environments with only a test and production workspace.



The diagram depicts the following processes and components to separate workspaces in this approach.

Expand table

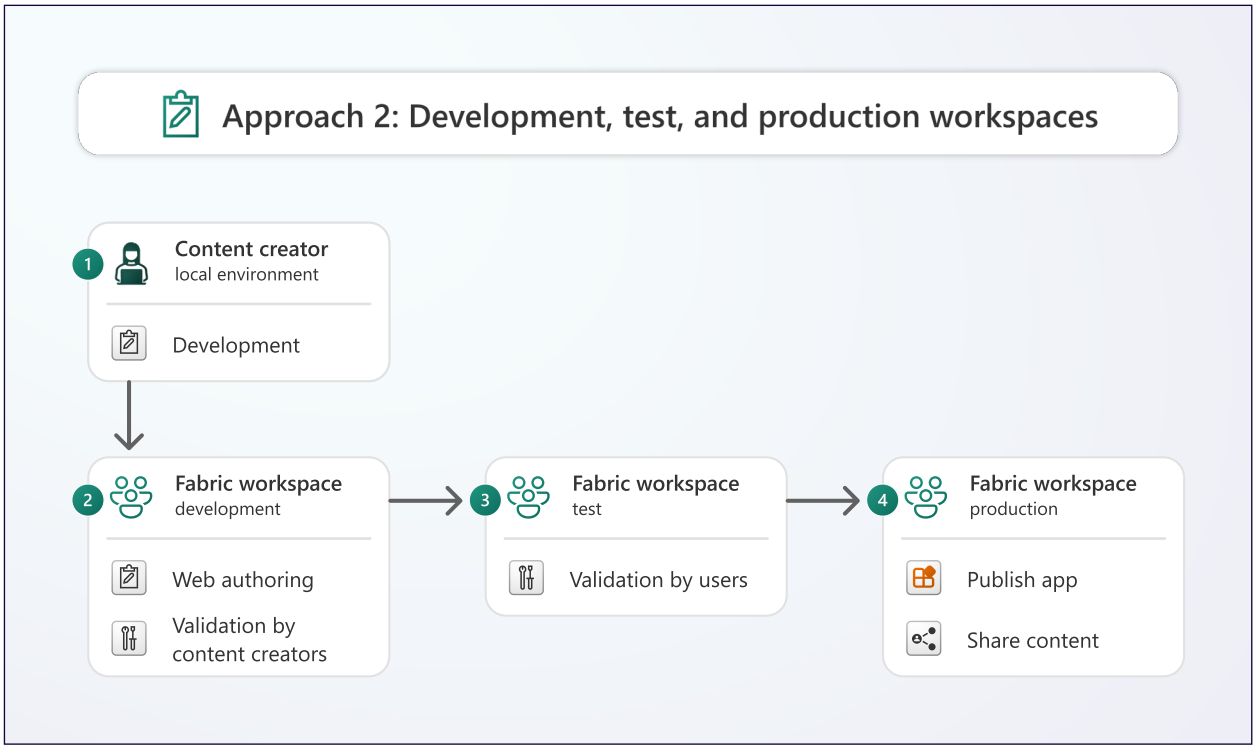
Item	Description
1	Content creators develop content in their local environment.
2	When ready, content creators publish content to a test workspace. Content creators can develop content that can only be produced with web authoring and perform content validation in the test workspace.
3	When ready, content creators deploy content to a production workspace. In the production workspace, content creators distribute content by publishing a Power BI app or sharing content from the workspace.

> **Note**
>
> There are many different ways to set up and use separate workspaces, and deploy content between these workspaces. However, we recommend that you don't perform local development, then publish content directly to a production workspace. This can lead to preventable disruptions and errors.

Development, test, and production workspaces

When delivering business-critical content, you typically use three or more separate workspaces. In this scenario, content creators often collaborate in an additional, development workspace that contains the latest version of the solution.

The following diagram depicts a high-level example of how you might use separate environments with a development, test, and production workspace.



The diagram depicts the following processes and components to separate workspaces in this approach.

Expand table

Item	Description
1	Content creators develop content in their local environment.
2	When ready, content creators publish content to a development workspace. In the development workspace, content creators can develop content that can only be produced with web authoring. Content creators can also validate content in the development workspace.
3	When ready, content creators deploy content to a test workspace. In the test workspace, users validate content, either in the workspace or an app.
4	When ready, content creators deploy content to a production workspace. In the production workspace, content creators distribute content by publishing a Power BI app or sharing content from the workspace.

Note

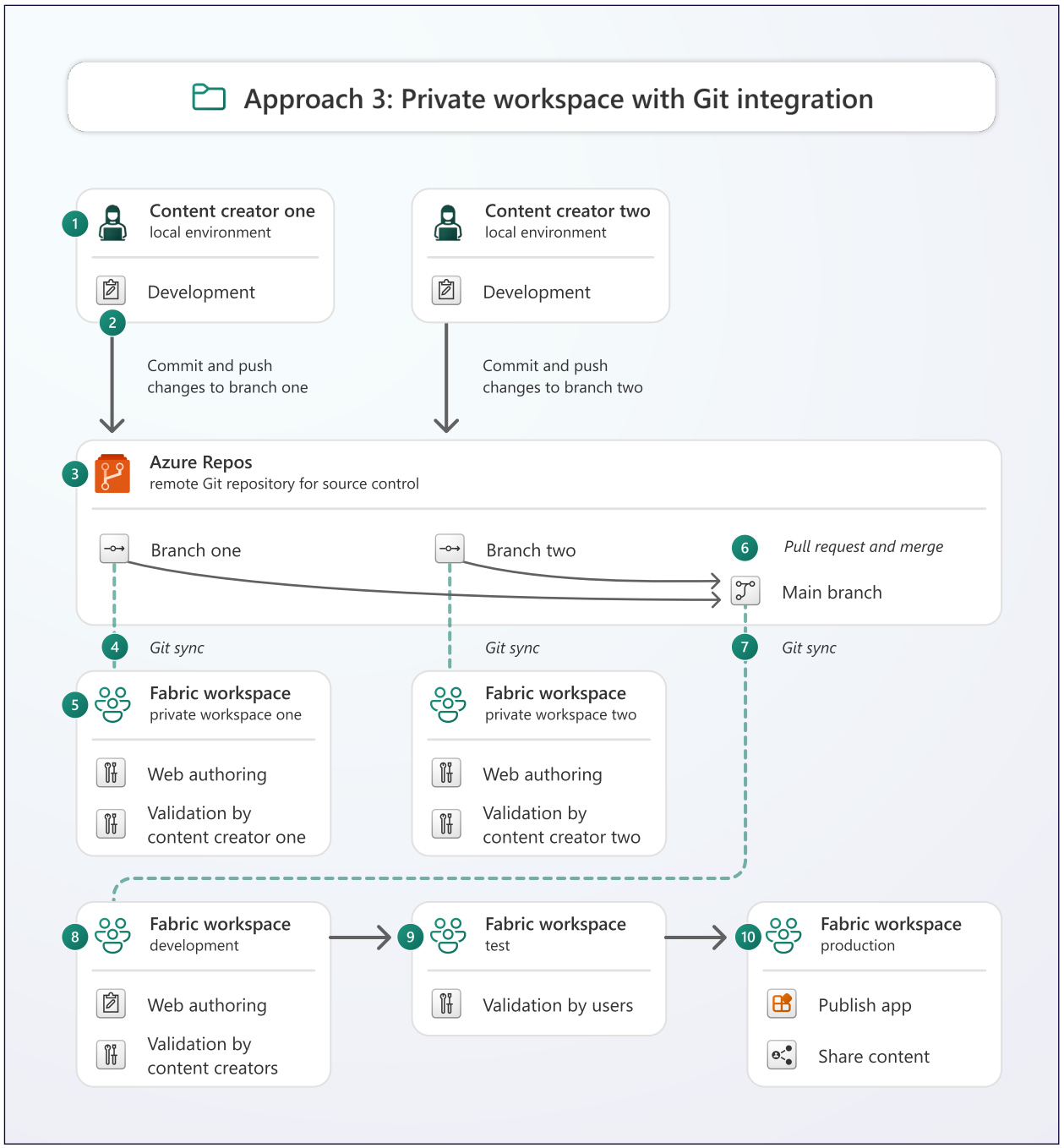
You can use up to ten different environments with deployment pipelines. For instance, you might want to have a pre-production environment between test and production that's specifically for special types of validation, like performance testing.

Private workspace with Git integration

When delivering business-critical content, each developer can also use their own, private workspace for development. In this scenario, a private workspace allows content creators to test content in the Fabric portal, or use features like scheduled refresh without risking disruption to others in the development team. Content creators can also develop content in the Fabric portal here, such as dataflows. Private workspaces can be a good choice when you are managing content changes by using Git integration together with Azure DevOps.

> **Note**
>
> Azure DevOps is a suite of services that integrate with Power BI and Fabric to help you plan and orchestrate content lifecycle management. When you use Azure DevOps this way, you typically leverage the following services:
>
> - Azure Repos: Allows you to create and use a remote Git repository, which is a remote storage location you use to track and manage content changes.
> - Azure Pipelines: Allows you to create and use a set of automated tasks to handle, test, and deploy content from a remote repository to a workspace.
> - Azure Test Plans: Allows you to design tests to validate the solution and automate quality control together with Azure Pipelines.
> - Azure Boards: Allows you to use boards to track tasks and plans as work items, and link or refer to work items from other Azure DevOps services.
> - Azure Wiki: Allows you to share information with their team to understand and contribute to content.

The following diagram depicts a high-level example of how you might use separate environments by using a private workspace with Git integration.



> **Note**
>
> The diagram depicts separate developers working on separate branches of a solution (branch one and branch two) before merging their changes into a main branch. This is a simplified depiction of a Git branching strategy to illustrate how developers can integrate their changes with a remote Git repository and benefit from Git integration in Fabric.

The diagram depicts the following processes and components to separate workspaces in this approach.

Expand table

Item	Description
1	Each content creator develops content in their own local environment.
2	When ready, content creators commit and push their changes to a remote repository, such as an Azure Repos Git repository.
3	In the remote Git repository, content creators track and manage content changes by using source control, and branch and merge content to facilitate collaboration.
4	Content creators sync a branch of the remote repository with a private workspace. After syncing, the latest changes that a creator commits and pushes to the branch are visible in that private workspace. Different content creators work on their own, separate branches as they make changes.
5	In the private workspaces, content creators can develop content by using web authoring, and validate their own changes. Changes to content made by web authoring can sync with the branch in the Git repository when the content creator commits and pushes these changes from the workspace. Different content creators work in their own, separate private workspaces.
6	When ready, content creators perform a pull request to merge their changes into the main branch of the solution.
7	After merging changes, the main branch syncs with the development workspace.
8	In the development workspace, content creators can develop content that isn't supported by Fabric Git integration, such as dashboards. Content creators also validate the integrated solution that contains all of the latest changes.
9	When ready, content creators deploy content to a test workspace. In the test workspace, users perform user acceptance testing of content.
10	When ready, content creators deploy content to a production workspace. In the production workspace, content creators distribute content by publishing a Power BI app or sharing content from the workspace.

Supporting resources for workspaces

When you use separate environments, you should also consider how this will impact various supporting resources that you use in these environments. For these supporting resources, consider whether you also need to separate them into the same number of stages, or else how you'll coordinate their use across these environments.

- **Gateways**: Consider using separate on-premises data gateway clusters and VNet gateways for production workloads. This is beneficial to prevent disruption, but also to ensure uptime when you need to update these gateways.

- **Apps**: Consider having separate apps for test and production workspaces. It's not possible to deploy or copy apps between stages. Apps in the test workspace are intended to help you test content and the app experience before you deploy changes to the production workspace. Apps in the production workspace are intended to deliver content to end-users in a structured and experience.
- **Azure DevOps**: If you intend to use Azure DevOps to collaborate and orchestrate source control and deployment, consider how you'll use branches and Azure Pipelines to deploy content between these environments. For more information about using Azure Pipelines to deploy content, see Stage 4: Deploy content.

Once you've decided how you'll set up and use workspaces, the next step is to decide how you'll perform version control to track and manage content changes.

Decide how you'll use version control

In Power BI, you can perform version control either by using SharePoint/OneDrive, or by using a remote Git repository, such as Azure Repos, which is a component of Azure DevOps. Instead of Azure DevOps, you can also use GitHub. In both approaches, you add local content files to a remote storage location to help track and manage changes. We recommend that you use SharePoint or OneDrive only for smaller and simpler projects, as it lacks the features and robustness to support larger or more complicated scenarios.

Here are some general considerations to help you set up and use version control.

- **Alerts**: You should set up alerts for when others add, remove, or modify critical files.
- **Scope**: Clearly define the scope of the remote storage location. Ideally, the scope of the remote storage location is identical to the scope of the downstream workspaces and apps that you use to deliver content to consumers.
- **Access**: You should set up access to the remote storage location by using a similar permissions model as you have set up for your deployment pipeline permissions and workspace roles. Content creators need access to the remote storage location.
- **Documentation**: Add text files to the remote storage location to describe the remote storage location and its purpose, ownership, access, and defined processes.

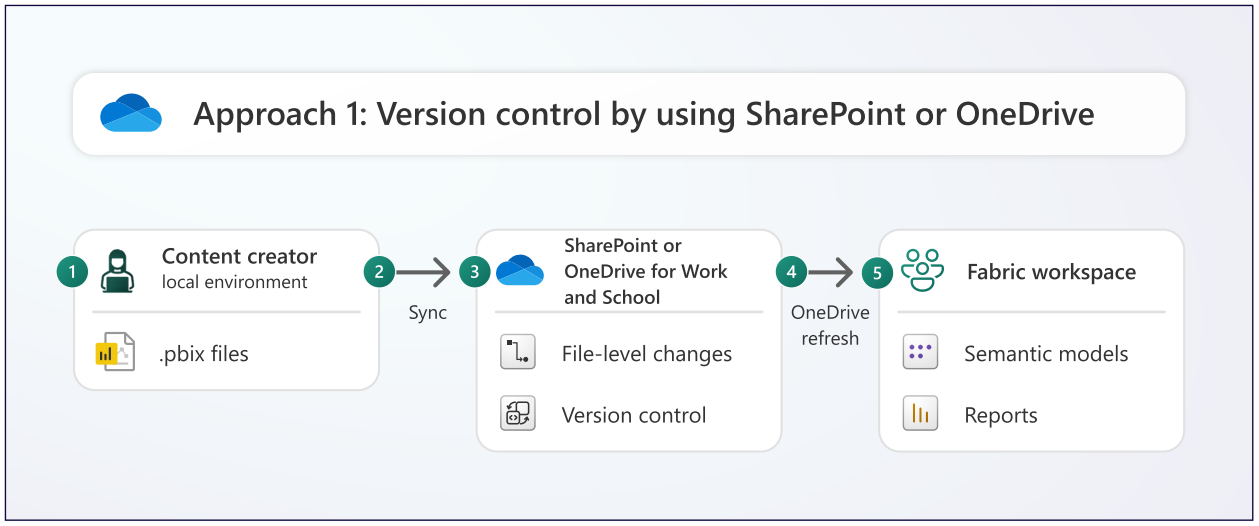
The following sections describe each approach and key considerations to decide which one you should use.

Version control by using SharePoint or OneDrive for Work and School

SharePoint has built-in version control for files. Content creators can develop semantic models or reports locally, and their changes are synchronized to a SharePoint or OneDrive for Work and School document library.

> **Tip**
>
> Use SharePoint or OneDrive only for version control in simpler, smaller scenarios, like self-service content publishing. For larger or more complicated scenarios, you should consider performing source control by using a remote Git repository.

The following diagram depicts a high-level overview of how you perform version control by using SharePoint or OneDrive.



The diagram describes the following processes and components.

Expand table

Item	Description
1	Content creators develop semantic models and reports in their local environment and save these models and reports as .pbix files.
2	When ready, content creators save their changes, which they sync to a remote SharePoint or OneDrive for Work and School library.
3	In the remote library, content creators track file-level changes which facilitates version control.
4	Content creators can link a published semantic model or report to a .pbix file to facilitate OneDrive refresh. OneDrive refresh automatically publishes content changes every hour.
5	In the Fabric workspace, content creators see the changes to Power BI content after OneDrive refresh completes.

> **Important**
>
> You can only perform version control by using SharePoint or OneDrive for Power BI Desktop files that contain semantic models or reports. You can't easily perform version control by using SharePoint or OneDrive for other Power BI item types like dataflows, or for Fabric item types like notebooks. For these other item types, you should perform version control by using a remote Git repository, like Azure Repos.

To summarize, content creators can link a published semantic model or report to a .pbix file that's stored in a SharePoint or OneDrive for Work and School library. With this approach, content creators no longer have to publish the semantic model to see changes. Instead, the changes are visible after an automatic *OneDrive refresh*, which occurs hourly. While convenient, this approach comes with some considerations, and it can't be easily reversed.

While easy to set up and manage, version control with SharePoint or OneDrive has more limitations. For example, it's not possible to view changes *within* the content, and it's also not possible to compare versions. Additionally, it's not possible to set up more sophisticated processes to manage these changes (like branching or pull requests, described later in this article).

Consider using SharePoint or OneDrive to track and manage changes in the following scenarios:

- Content consists of only semantic models or reports saved as .pbix files.
- Self-service users create and manage the content.
- Content creators collaborate by using Microsoft Teams.
- Content creators are inexperienced with Git and source control management.
- Content creators manage a single item with limited complexity and collaboration.
- The .pbix files have a sensitivity label applied that encrypts their contents.

> **Note**
>
> OneDrive and SharePoint support content that has sensitivity labels applied, except for some **limited scenarios**. In contrast, Fabric Git integration **doesn't support sensitivity labels**.

Avoid using SharePoint or OneDrive to track and manage changes in the following scenarios:

- Content consists of item types other than semantic models and reports.

- Content is complex or large in scope.
- Content creators need to work collaboratively on the same item at the same time.

The following sections describe additional considerations to effectively use version control by using SharePoint or OneDrive with Power BI.

Microsoft Teams integration

You might use the document libraries from Microsoft Teams if content creators use it for their collaboration. Document libraries are SharePoint sites, and using the document libraries (instead of a separate SharePoint site or OneDrive folder) ensures centralization of content, files, and collaboration.

Check-in and check-out files

You should check-out files that you're working on to avoid possible change conflicts. Once changes are complete, check-in the files with comments that describe the change. Checking-in and checking-out files helps you to improve collaboration between content creators when you use SharePoint or OneDrive for Work and School for version control. If you check-in and check-out files with multiple content creators, you can modify the SharePoint site library to view the last update and the comments of the last check-in.

Version history

Ensure that you back up content to a separate location in case of unexpected disruptions to your SharePoint site document library. You should also set limits to the number of versions kept in the SharePoint site and delete old versions. This ensures that version history remains meaningful and doesn't take up too much space.

For more sophisticated version control, you can store files in a remote repository like Azure Repos and manage changes by using source control.

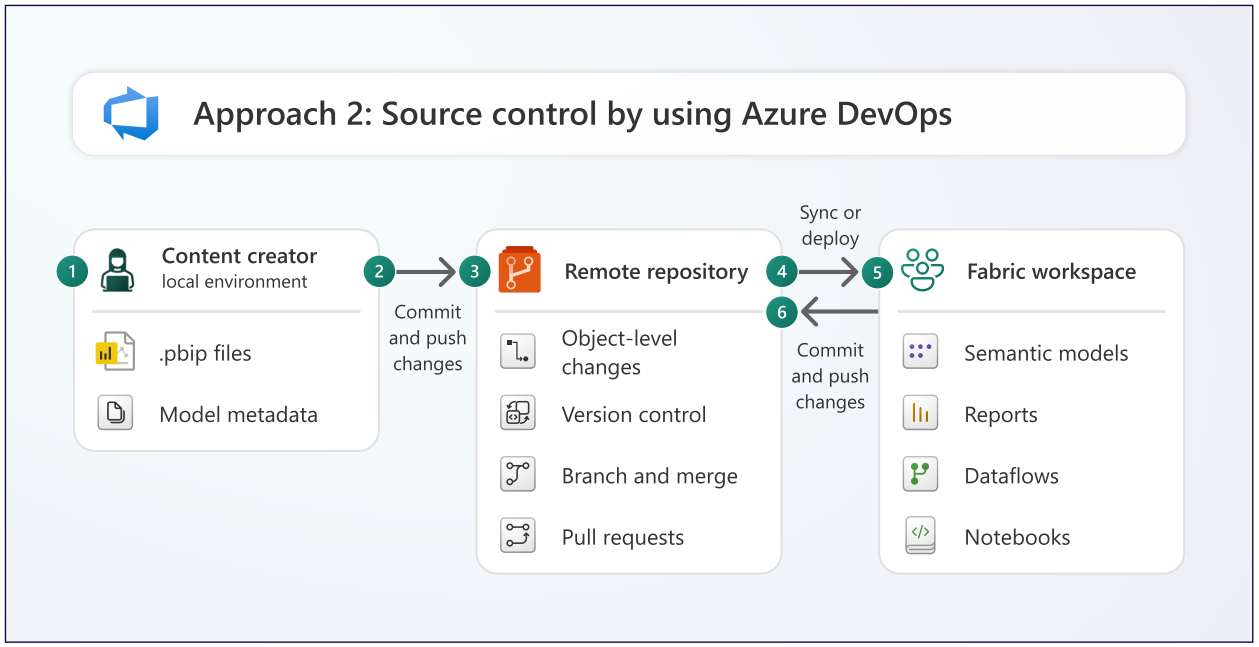
Source control by using a remote Git repository

A remote Git repository facilitates source control of files, which allows content creators more options to track and manage changes. In brief, content creators can develop content either locally or in the Power BI service, then commit and push those changes to a remote Git repository like Azure Repos or GitHub.

> **Note**

You can also perform source control of your content without using Git integration. In this scenario, you still track and manage content changes in a remote Git repository, but you deploy content using either the REST APIs or XMLA read/write endpoints. For more information about these methods of deploying content, see Stage 4: Deploy content.

The following diagram depicts a high-level overview of how you perform source control by developing content locally, then committing changes to a remote repository, which can sync with a Fabric workspace by using Git integration.



The diagram describes the following processes and components.

Expand table

Item	Description
1	Content creators develop semantic models and reports in their local environment and save these models and reports as .pbip files. Content creators can also develop semantic models and save model metadata.
2	When ready, content creators save their changes, which they commit and push to a remote Git repository at regular intervals.
3	In the remote Git repository, content creators track object-level changes, which facilitates version control. Content creators can also create branches to work on content, and merge their changes into a single branch by using pull requests.
4	Content creators can sync content from the remote repository by using Git integration. They can also deploy content by using other approaches, such as Azure Pipelines together with the REST APIs.

Item	Description
5	In the Fabric workspace, content creators see the changes to Power BI content after sync or deployment completes. Content creators can author content like dataflows or notebooks in the workspace. If they use Git integration, content creators link this workspace to a specific branch of the remote repository.
6	Content creators can commit and push changes from a workspace to the remote repository by using Git integration. These changes can import item definitions for supported content authored in the workspace, like dataflows and notebooks.

In summary, content creators store .pbip files, metadata files, and documentation in a central Azure Repos remote repository. These files are curated by a *technical owner*. While a content creator develops a solution, a technical owner is responsible for managing the solution and reviewing the changes, and merging them into a single solution. Azure Repos provides more sophisticated options for tracking and managing changes compared to SharePoint and OneDrive. Maintaining a well-curated, documented repository is essential because it's the foundation of all content and collaboration.

Consider using source control to track and manage changes in the following scenarios:

- Centralized or decentralized teams create and manage the content.
- Content creators collaborate by using Azure DevOps.
- Content creators are familiar with Git, source control management, or DataOps principles.
- Content creators manage complex or important content or expect the content to scale and grow in complexity and importance.

Here are some pre-requisites and considerations to help you effectively use source control with Azure DevOps.

- **Git**: To commit and push changes to a remote repository, content creators need to download and install *Git*. Git is a distributed version control system that tracks changes in your files. To learn Git basics, see What is Git?.
- **Tools**: To use Git, content creators need to either use a command line interface (CLI) or a graphical user interface (GUI) client for SCM, like Visual Studio or Visual Studio Code.
- **Licenses and permissions**: To create and use an Azure Repos Git repository, content creators must have the following:
  - Access level set to Basic (as opposed to Stakeholder).
  - Belong to an organization and a project.
  - Appropriate repository permissions.

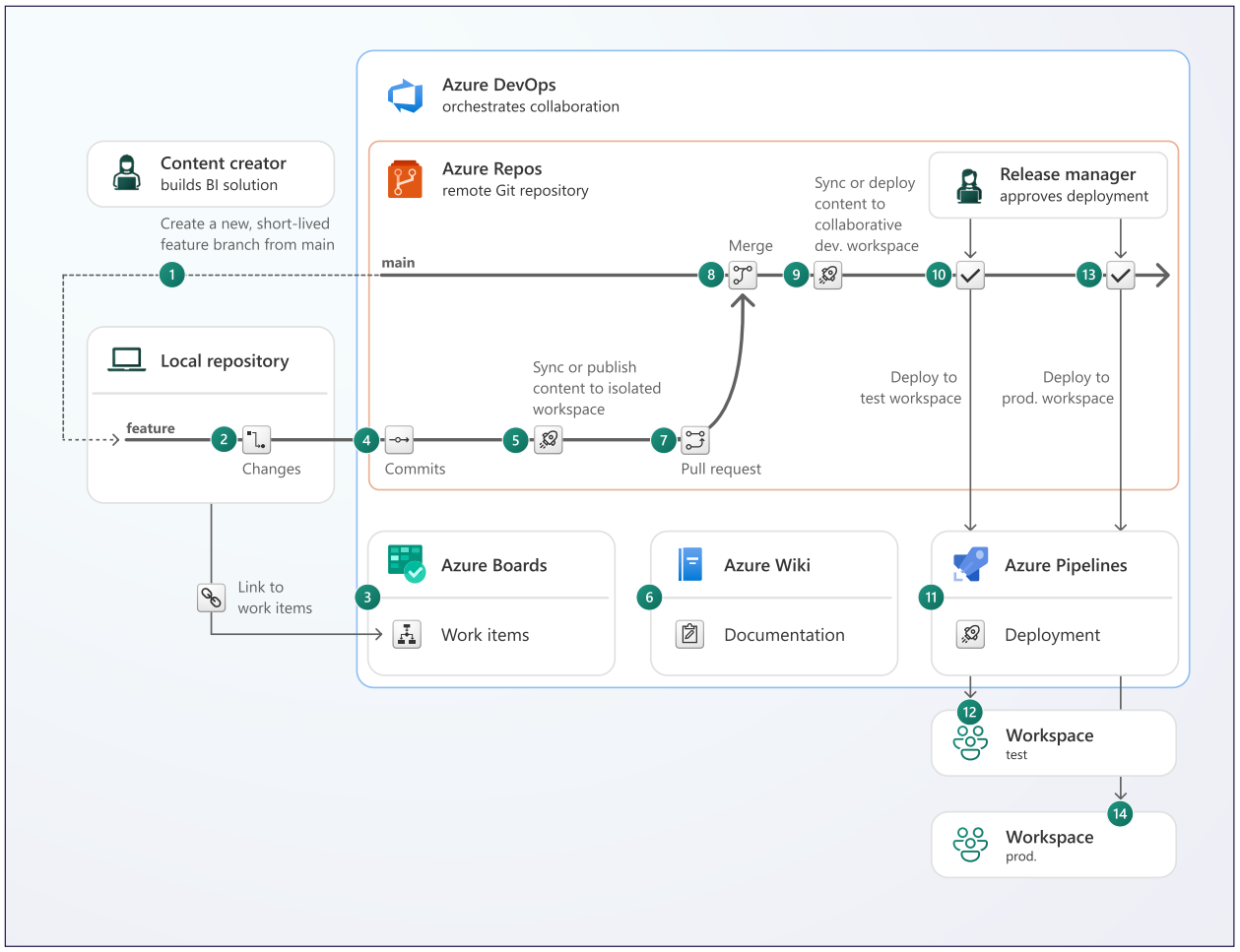
- **Fabric Git integration**: To sync content in a remote repository with a Microsoft Fabric workspace, content creators use Fabric Git integration. This is important to track and manage changes to content that's created in the Fabric portal, like dataflows.

> **Tip**
>
> To facilitate source control with local development, we recommend using a client application like **Visual Studio Code**. You use Power BI Desktop to develop content, then you can use Visual Studio Code to perform source control management of that content, by staging, committing, and pushing changes to your remote repository.

> **⚠ Warning**
>
> Fabric Git integration has **some limitations** with the supported items and scenarios. Ensure that you first validate whether Fabric Git integration best suits your specific scenario, or whether you should take a different approach to implement source control.
>
> Additionally, **sensitivity labels aren't supported with Fabric Git integration**, and **exporting items with sensitivity labels might be disabled**. If your content has sensitivity labels, you should plan how you can achieve version control while still honoring your data loss prevention policies.

A key benefit of using source control with Azure Repos or GitHub is improved collaboration between content creators and more customization and oversight regarding content changes and deployment.

The following diagram depicts a high-level overview of how Azure DevOps enables collaboration with source control. You can also use GitHub Enterprise instead of AzureDevOps, which would involve different services that perform a similar function.



> **Note**
>
> The previous diagram depicts one example of how to collaborate by using Azure DevOps. Choose an approach that best fits the needs and the way of working of your team.

The diagram depicts the following user actions, processes, and features.

Expand table

Item	Description
1	A content creator makes a new, short-lived branch by cloning the main branch, which contains the latest version of the content. The new branch is often referred to as the feature branch, as it's used to develop a specific feature or fix a specific issue.
2	The content creator commits their changes to a local repository during development.
3	The content creator links their changes to work items that are managed in Azure Boards. Works items describe specific developments, improvements, or bug fixes scoped to their branch.

Item	Description
4	The content creator regularly commits their changes. When ready, the content creator publishes their branch to the remote repository.
5	To test their changes, the content creator deploys their solution to an isolated workspace for their development (not shown in this diagram). The content creator can also sync the feature branch to the workspace by using Fabric Git integration.
6	Content creators and content owners document the solution and its processes in an Azure Wiki, which is available to the entire development team.
7	When ready, the content creator opens a pull request to merge the feature branch into the main branch.
8	A technical owner is responsible for reviewing the pull request and merging changes. When they approve the pull request, they merge the feature branch into the main branch.
9	A successful merge triggers deployment of the solution to a development workspace by using an Azure Pipeline (not shown in this diagram). When using Fabric Git integration, the main branch syncs to the development workspace.
10	The release manager performs a final review and approval of the solution. This release approval prevents the solution from being published before it's ready. In enterprise content publishing, a release manager typically plans and coordinates the content release to test and production workspaces. They coordinate and communicate with content creators, stakeholders, and users.
11	When the release manager approves the release, Azure Pipelines automatically prepare the solution for deployment. Alternatively, an Azure Pipeline can also trigger a deployment pipeline to promote content between workspaces.
12	Users test and validate content in the test workspace. When using Git integration with Azure Pipelines for deployment, the test workspace doesn't sync with any branch.
13	After users accept and validate changes, the release manager performs a final review and approval of the solution to deploy to the production workspace.
14	Users view content that's published to the production workspace. When using Git integration with Azure Pipelines for deployment, the production workspace doesn't sync with any branch.

The following sections describe additional considerations to effectively use source control by using Azure DevOps and Power BI.

> **ⓘ Important**
>
> Effective use of branching, commits, pull requests, and merges are essential to most benefit from source control when you manage the lifecycle of your Power BI

content.

Use branches

Content creators achieve collaboration by using a *branching strategy*. A branching strategy allows individual content creators to work in isolation in their local repository. When ready, they combine their changes as a single solution in the remote repository. Content creators should scope their work to branches by linking them to work items for specific developments, improvements, or bug fixes. Each content creator creates their own branch of the remote repository for their scope of work. Work done on their local solution is committed and pushed to a version of the branch in the remote repository with a descriptive *commit message*. A commit message describes the changes made in that commit.

When you use a branching strategy to manage Fabric content, consider the following factors.

- Which branch content creators should clone for their own work. For instance, if these branches are a clone of the main branch (known as *trunk based development*) or if they are other branches, like release branches for specific, planned versions of content.
- Whether you'll scope specific work to distinct content releases by using release branches.
- Which branches connect to which workspace when you use Fabric Git integration.

> **Tip**
>
> See **Adopt a Git branching strategy** for specific guidance and recommendations about how to best use a branching strategy to effectively facilitate collaboration.

Batch changes in commits

While developing content, creators should regularly stage changes into batches (or groups). These changes should address a specific work item for the solution (such as a feature or issue). When ready, content creators commit these staged changes.

Batching changes this way has several benefits.

- It helps structure development and gives content creators a chance to review and document the changes that they've grouped.

- It improves the alignment between planning and development, making it easier to link features and issues and get transparency about how the work is proceeding.
- Technical owners can more easily review pull requests from content creators if changes are batched appropriately and have clear commit messages.

When you stage and commit changes to Power BI content, consider the following factors.

- Whether you'll stage and commit changes from a local repository or from the Fabric workspace.
- Place .pbip files in top-level folders when you store multiple models or reports in a single repository. This will make it easier to identify and group changes that you make.
- Ignore innocuous or unhelpful metadata changes by using a gitignore file or a .gitattributes file. This will ensure that all changes you commit are meaningful.

> **Tip**
>
> See **Save your work with commits** for specific guidance and recommendations about how to stage and commit your work to a Git repository.

Create pull requests to merge changes

To merge the changes, a content creator opens a *pull request*. A pull request is a submission for peer review that can lead to the merge of the work done into a single solution. Merging can result in conflicts, which must be resolved before the branch can be merged. Pull request reviews are important to ensure creators adhere to organizational standards and practices for development, quality, and compliance.

When you use pull requests to merge changes to Power BI content, consider the following factors.

- What standards and practices you'll use to perform your review, if any. For instance, you can use rules from the Best Practice Analyzer for semantic models.
- How you'll review changes to report metadata, which isn't easy to read and understand without using third-party tools.
- How you'll address and resolve merge conflicts.

> **Tip**

See About pull requests and Merge strategies and squash merge for specific guidance and recommendations about how to best use pull requests to facilitate collaboration by merging changes to content.

Checklist - When planning where you'll store files, key decisions and actions include:

- ✓ **Choose your development tools**: Ensure your approach to develop content aligns with how you'll collaborate with other content creators and use version control.
- ✓ **Choose between .pbip and .pbix formats for models and reports**: Typically, the .pbip format is more beneficial for source control, but self-service users can find .pbix files easier to use.
- ✓ **Separate semantic model and report development**: Version control is most effective when you manage changes for different item types, separately. Separating model and report development is considered a good practice.
- ✓ **Decide how many workspaces that you need**: Using separate environments is critical for the success of content lifecycle management. Ensure that you've clarified how many workspaces you need and conduct appropriate workspace-level planning.
- ✓ **Decide how you'll implement version control**: Decide between a simpler approach by using SharePoint or OneDrive for Business, or by using Azure DevOps to facilitate source control.
- ✓ **Set up your remote repository**: Create a structured space in the OneDrive folder or Git Repo where you'll store content to track and manage changes.
- ✓ **If you're using source control, set up .gitignore and .gitattributes files**: Ensure that you set up your repository so that you're only tracking meaningful changes.
- ✓ **If you're using source control, define branching and merge strategies**: Ensure that you define clear processes for how you'll set up and use source control to best support development. Avoid overcomplicating your process. Instead, try to complement the current way of working in your development teams.

Related content

In the next article in this series, learn how to validate content as part of managing the content lifecycle.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Validate content

Article • 03/13/2025

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to validate content as part of managing the content lifecycle. It's primarily targeted at:

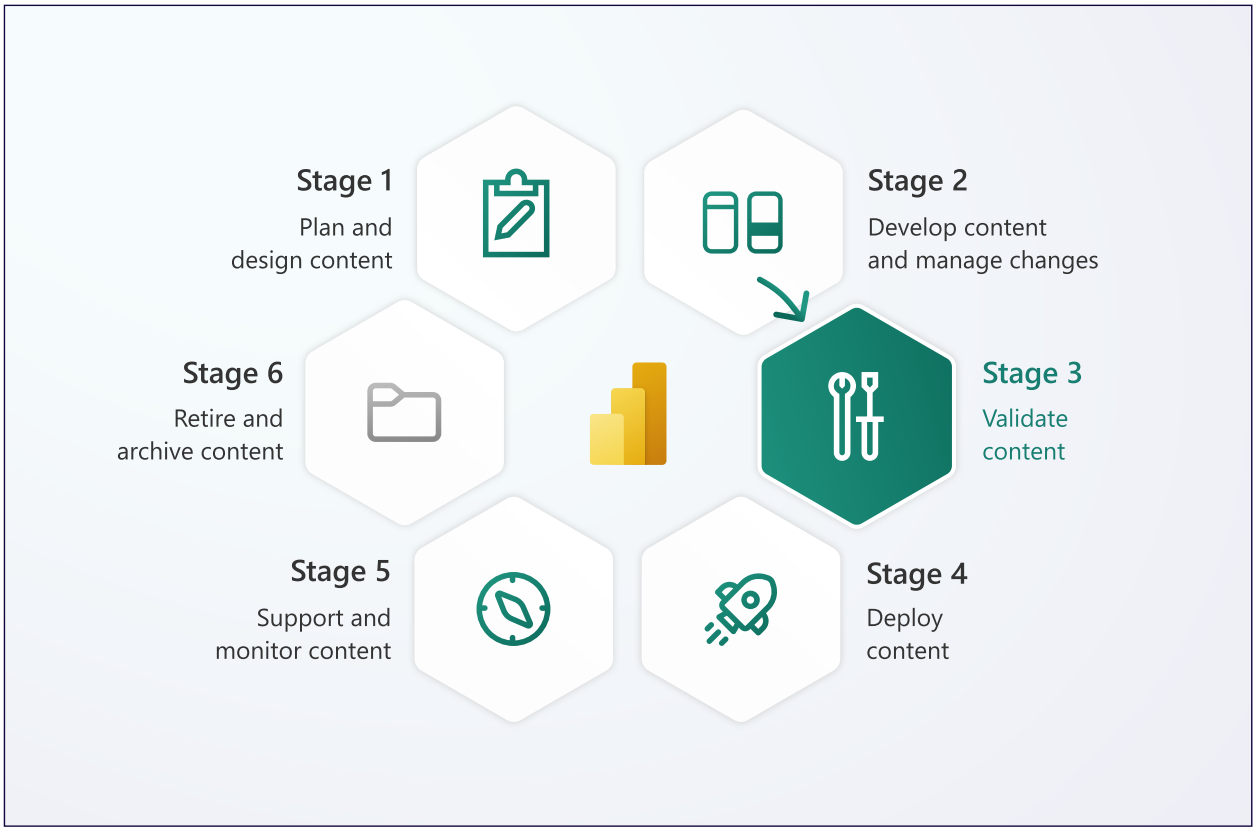
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content. These teams can also include release managers, who handle the lifecycle of content releases, and engineers who create and manage the components needed to effectively use and support lifecycle management.
- **Content creators and content owners**: Users who create content that they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the Power BI content that they create.

Lifecycle management consists of the processes and practices that you use to handle content from its creation to its eventual retirement. In the second stage of lifecycle management, you develop content and manage changes, which involve key decisions about how you'll develop content and set up workspaces and version control. In the third stage, you validate content to test whether it's ready for deployment.

> **Note**
>
> You typically iterate through stages two and three in successive development and validation cycles.

Validating content is critical to ensure the quality and trustworthiness of your solutions. For this reason, it's essential that you test content changes before you deploy them to production.

The following image depicts the lifecycle of Power BI content, highlighting stage three, where you validate content.



> **Note**
>
> For an overview of content lifecycle management, see the first article in this series.
>
> This article focuses on key considerations and decisions for validating content throughout its lifecycle. For more guidance about how to validate content, see:
>
> - Migrate to Power BI: Validate content: This article describes key considerations and decisions for validation when you migrate to Power BI from other technologies.
> - BI solution planning: Validate content: This article describes how to plan for iterative development and validation cycles when planning your Power BI or Fabric solution.

Validating content involves making specific decisions or actions to ensure that the content performs as expected.

When you validate content, you assess different aspects of the solution.

- **Functionality**: Whether the items and features that comprise the solution are functional. An example of testing functionality is whether a semantic model can

complete a scheduled refresh.

- **Data accuracy**: Whether the figures and results shown are complete and align to business expectations. An example of testing data accuracy is whether a value in a report aligns to a known baseline.
- **Performance**: Whether queries produce minimal impact on available user resources or user wait times. An example of testing performance is whether a dataflow reliably refreshes without reaching timeout or experiencing long refresh durations.
- **Security**: Whether unauthorized individuals are restricted from viewing or accessing information, or the entire solution. An example of testing security is the impersonation of a user or role when validating row-level security (RLS).
- **Effectiveness**: Whether the solution addresses the relevant business problem or process, and sufficiently supports business objectives as intended. An example of testing effectiveness is the gathering of user feedback when you conduct user acceptance testing (UAT).
- **Accessibility**: Whether the solution meets known accessibility standards so that it's usable by as many people as possible. An example of accessibility testing is checking that your report meets the Microsoft report accessibility checklist.

You validate content by conducting different kinds of tests. The following sections describe key considerations for decisions about how content creators and content consumers perform testing.

> ⓘ Note
>
> Many teams use test methodologies that originate from software development, such as unit tests, integration tests, and smoke tests. There are many equally valid approaches to content testing and validation. What's most important is that you test content by using an approach that works best for your needs and the way your team works.

Decide how creators should validate content

Content creators should validate their own changes to content to ensure the quality and functionality of the changes. Tests typically happen in the development workspace, which contains the latest working version of a solution. Content creators test their own changes before content is deployed to a test workspace for user validation.

> ⓘ Note

It's essential that content creators validate their own content before it's made available to users. If a solution is provided to test users with obvious issues, it erodes trust in the solution. Even when testing, users expect to see a reasonable representation of the final product. Additionally, a functional solution allows users to focus on identifying issues related to their business area.

There are two ways for content creators to validate content.

- **Manual testing**: Manual tests involve someone manually validating the content, either through subjective assessment or by comparing to some objective test criteria. Manual tests are easy to perform, but they're subject to human error or bias. Additionally, when content reaches a certain scale, manual tests can become laborious to properly perform. You can conduct manual testing in two ways.
  - Independent review, which involves you testing your own content, like semantic models and reports.
  - Peer review, which involves a subjective assessment of the content to critically appraise the solution and provide suggestions to improve it.
- **Automated testing**: Automated tests involve a preprepared test that's evaluated automatically without human intervention. Typically, automated tests check parts of the solution code against specific benchmarks or baselines. Automated testing is more difficult to perform and takes time and effort to set up. However, automated testing is essential in enterprise scenarios to ensure the quality and trustworthiness of larger implementations and business-critical solutions.

The following sections describe different ways that content creators can conduct manual testing, automated testing, and peer review.

Conduct manual testing

You should conduct your own manual tests on the content that you create. These tests should ensure that your changes work as expected and achieve desired quality standards. Typically, manual testing involves use and subjective assessment of the content or specific content changes, and describing and documenting the results.

Here are some considerations when testing your own content.

- Decide and document beforehand the test conditions and success criteria.
- Be thorough with testing and with documenting the test results. However, ensure that you avoid superfluous testing so that your test practices don't slow down development.
- Create a standard set of tests for each item type to improve repeatability.
- Document test results and conclusions.

- Test multiple times to ensure that your test results best reflect reality and not random chance.
- Use test conditions that are representative of your production environment.

The following sections describe other key considerations for manual testing.

Manually test semantic models

Semantic models are an important part of a solution in Fabric and Power BI because they're an upstream source for reports, dashboards, and other client tools and Fabric workloads. As such, it's important to validate your semantic models before deployment.

Answer questions such as the following to help you validate your semantic model.

- Do tables contain unexpected missing, duplicate, or incorrect values?
- Do DAX measures return the expected results without long query times?
- Does a scheduled refresh complete successfully without long refresh times?
- Do you observe *(Blank)* results in visuals, filters, or query results caused by referential integrity violations?
- Does data security, like RLS or object-level security (OLS), sufficiently prevent unauthorized individuals from accessing the model or its data?
- Are model objects (like DAX measures or table columns) organized into display folders?

You can use different tools and approaches to help you validate your semantic models.

- **Power BI Desktop**: Power BI Desktop allows you to validate different aspects of your semantic models by using various features. Examples of Power BI Desktop features that facilitate testing semantic models include:
  - **Visual canvas**: Test model functionality and accuracy with drag-and-drop visuals.
  - **DAX query view**: Test model accuracy and DAX code with DAX queries that you can save and reuse later.
  - **Query diagnostics**: Test refresh performance by getting diagnostic information about how queries are evaluated in Power Query.
- **Fabric**: Features and items in the Fabric portal allow you to validate aspects of your semantic model once it's deployed to a workspace.
  - **Security**: Test model security by impersonating security roles or users.
  - **Notebooks**: Test model functionality and accuracy by using semantic link.
  - **Monitoring hub**: Test and monitor data refresh of semantic models and other Fabric data items.

- **Third-party tools**: Third-party tools allow you to validate other aspects of your semantic model, either by providing more detail or other features that facilitate validation. Examples of third-party tools that facilitate testing semantic models include:
  - **DAX Studio**: Test and optimize the performance of DAX code by receiving detailed breakdowns of DAX query timings and query plans.
  - **Tabular Editor**: Test and debug accuracy of DAX code by receiving detailed breakdowns of how DAX queries are evaluated and what evaluation context is active.

> **Tip**
>
> You can use query diagnostics to manually validate and optimize the performance of Power Query from other items that use it, like dataflows.
>
> Additionally, you can use the DAX query view and third-party tools like DAX Studio to validate and optimize DAX queries for paginated reports and scorecards.

Manually test reports

Reports are a common way for users to interact with your data. Many users depend upon reports to make decisions and take actions to make progress towards their business objectives. As such, it's important to validate your reports before deployment.

Answer questions such as the following to help you validate your reports.

- Do reports fulfill the documented business requirements?
- Are the right visual types used to address the right question?
- Are report pages clear and concise without overwhelming colors or too many visuals?
- Does the report work as expected when filtering to a narrow subset of data?
- Does the report allow export to Excel, and if so, does it allow retrieving summarized data or the underlying data?
- Can the report be used for cross-report drill through or personalization of visuals?

You can use different tools and approaches to help you validate your reports.

- **Power BI Desktop**: Power BI Desktop allows you to validate different aspects of your reports by using various features. Examples of Power BI Desktop features that facilitate testing reports include:
  - **Visual canvas**: Test report functionality by using slicers, filters, and other interactive elements.

    - **Performance analyzer**: Test report performance by measuring visual rendering and DAX query times. You can copy visual DAX queries from the performance analyzer to use in other tools and save performance results for documentation.
    - **Query limit simulations**: Test report performance by simulating the memory limits in the capacity where it will be deployed.
- **Fabric**: Features and items in the Fabric portal allow you to validate aspects of your report once it's deployed to a workspace.
    - **Update app**: Test report functionality and security when distributing reports in Power BI apps and setting different app audiences to determine who can view which content. When you use app audiences, you can preview which reports they'll have access to, and test the app experience, yourself.
    - **Reading view in the workspace or app**: Test report functionality and accuracy by using it in the same environment as a user.

> ⓘ Note
>
> You can only develop and validate dashboards in the Fabric portal.

> ⓘ Important
>
> It's essential to test your reports in both Power BI Desktop and after deployment in the Fabric portal. Visual rendering might behave differently on your local machine compared to reports in a Fabric workspace. Additionally, be aware that the user experience of using a report from a workspace or app differs significantly from using a report in Power BI Desktop.

Manually test by performing a peer review

Another way to manually validate content is by performing a peer review. In a peer review, the content creator provides the solution or part of the solution to a colleague to evaluate. The purpose of a peer review is to improve a solution by using the collective experience and expertise of multiple content creators. You can perform peer review both during and after manual and automated testing.

> ⓘ Note
>
> Peer review is a standard approach that's used in many industries. This approach is commonly known to improve the quality of content, products, and processes.

> **Tip**
>
> If you're the only content creator for a solution, consider finding another content creator on a different team to review your solution, and offer to do the same for them.

There are different ways that you can conduct a peer review.

- **Functional review**: A functional review focuses on features, processes, or business requirements that the solution should fulfill. In a functional review, the reviewers use the solution as if they were an end user. They document any defects or issues they find, together with any subjective critique to improve the implementation.
- **Technical review**: A technical review focuses on technical aspects of the solution, like data modeling, code, or design. In a technical review, reviewers assess how certain features or changes were implemented, and suggest alternative approaches or highlight potential flaws or risks to the current approach.
- **Pull request**: When you perform source control, you create a pull request (PR) to merge your changes to the latest version of a solution. A technical owner reviews the proposed changes and evaluates the source code. This kind of review is useful for ensuring that code adheres to standard conventions, like formatting DAX or M code, or identifying anti-patterns or potentially problematic code.

> **Tip**
>
> We recommend that you perform some kind of formal peer review and approval before content changes can move to user acceptance testing. That's because poor quality content can harm trust in your data solutions, even during testing. Furthermore, peer review can also yield benefits to collaboration and knowledge-sharing among team members.

Once you complete a peer review cycle, you should document and incorporate any recommended changes. If necessary, you should resubmit the changes for approval before moving on to user testing. Typically, multiple iterations of peer review are only needed when there are many changes or a few complex changes to test.

Manually test Copilot and AI skills outputs

You can use Copilot or AI skills to allow users to ask questions about your data model by using natural language. However, using these tools leverage generative AI, and might

return low-quality or inaccurate outputs. It is always important to evaluate these outputs before you use them.

To test Copilot and AI skills outputs, you can:

- Ask simple questions testing known benchmarks, such as the total sales in a previous month, or the total number of customers or products.
- Compare outputs to results obtained from reports or *ad hoc* analyses.
- Compare outputs before and after making changes to a semantic model.

Automate testing

Content creators can automate testing so that tests are performed automatically prior to deployment. Automated tests typically involve preprepared test conditions that are run and orchestrated programmatically in response to certain actions, like saving content or submitting a pull request (PR). The results of automated tests are automatically stored for later reference and documentation.

The purpose of an automated test is to reduce the time and effort to validate content changes, while improving the consistency of the tests and reliability of their results. When content fails an automated test, it's typically prevented from being deployed until issues are resolved by the content creator.

Effective automated testing is a key part of implementing *DataOps*. DataOps allows teams to automate and scale processes by adopting practices that improve and accelerate delivery of data and analytics.

> **Important**
>
> To effectively automate testing, you should create well-designed tests. Creating such tests can take significant time and effort. If your test conditions and expectations are poorly defined, your automated tests won't be able to validate the right aspects of your content, and you'll receive little benefit automating these tests.

> **Tip**
>
> Automated testing is most beneficial when integrated with your solution deployment in **enterprise content publishing scenarios**. For example, you can automate testing by using Azure Pipelines as part of a validation pipeline, which

ensures content is ready to be deployed. For more information, see **Stage 4: Deploy content**.

The following sections describe key considerations to automatically test Power BI semantic models and reports.

Automate testing of semantic models

Automated testing of semantic models is possible, although it typically requires custom setup with third-party tools and frameworks.

You can use different tools and approaches to automate testing of semantic models.

- **Best Practice Analyzer (BPA)**: The Best Practice Analyzer allows you to specify rules that you can use to evaluate a semantic model. You can run the BPA by using Tabular Editor, which will identify any rule violations in a semantic model. You can automate checks for BPA rule violations by using the Tabular Editor command line interface (CLI) together with Azure DevOps, or as a part of another scheduled process.
- **Fabric notebooks and semantic link**: Notebooks in Fabric allow you to use semantic link to programmatically interact with semantic models. You can use notebooks to run frameworks like Great Expectations (GX) to validate data. Additionally, you can evaluate measures and DAX queries, then test the results against known baselines.
- **Power Automate**: Power Automate allows you to run queries against semantic models and export reports by using the Power BI REST APIs. You can check the query results against known baselines, and then perform downstream actions like triggering alerts to content owners.

> **Tip**
>
> Consider combining automated testing and the orchestration of your semantic models. For instance, you can conduct automated tests on a data source and semantic model prior to a refresh by using notebooks or Power Automate. If the tests fail, you can prevent the refresh, which also can prevent refresh errors or incorrect data from arriving in business reports.

Automate testing of reports

There are limited options available to automate testing of reports. These options rely on external tools or community solutions to automatically validate visuals or report properties, such as validating report metadata or simulating user interactions with reports.

You can use different tools and approaches to automate testing of reports.

- **Report best practice analyzers**: There are various third-party tools that support Best Practice Analyzer-like functionality to automate the detection of issues in reports by examining the report definition. Two tools that support this functionality are PBI Explorer and PBI Inspector.
- **Power Automate Desktop**: UI automation tools like Selenium for Python or Power Automate Desktop allow you to simulate user mouse interactions with reports. By defining a user flow, you can test navigation and interactions. These tests pass when they can complete the flow, and fail when they detect specific words or images on the screen (like an error message, or a blank visual).

Decide how users should validate content

Once content passes manual testing, automated testing, and peer review, it can move on to user testing. When users test content, they provide subjective feedback about whether that content fulfills business requirements and performs to their expectations, including returning accurate results.

User validation typically happens in a test workspace. When you set up a test workspace, take into account the following considerations.

- **Create a test app**: If you intend to distribute content by using a Power BI app, set up a test app for test users to validate content. The test app should be identical to the app you'll set up in production. In the navigation of the test app, consider including links to documentation, training, and feedback forms.
- **Provision access**: Identify a subset of users from the community who will validate the solution. Contact these users and form an agreement about when and why they should validate this content. Then, ensure that you provide them with access to the content, and add them to the appropriate security roles. Share links to the content or test app with the users so they can get started with testing.
- **Set up scheduled refresh**: User validation typically spans a longer period. It's worthwhile to set up a scheduled refresh of data items in the test workspace so that users are testing with the latest data.

> **Important**

When you deploy content to a test workspace, you need to manually update the app before changes to reports and dashboards are visible for users.

> **Note**
>
> You cannot deploy or copy apps from one workspace to another. Any changes to an app must be made manually in the configuration for that workspace.

Before you commence user validation, you should conduct the necessary preparations.

- Plan when user validation should occur.
- Specify whether user validation is confined to a specific period or part of an iterative process.
- Create a method to gather feedback, such as by using Microsoft Forms.
- Communicate to users involved in validation the planning and expectations.
- Organize a kick-off for user validation to guide users and manage expectations.
- Conduct training for users to demonstrate the validation and feedback process.

Here are some different ways to facilitate user validation of content.

- **Observatory testing**: Observatory tests are short sessions where content creators watch one or more users use the content without guidance or instruction. In these sessions, content creators use their observations to identify potential flaws, issues, or improvements to the solution. These tests can be valuable as they require little time and effort to organize, and can be limited to specific features or parts of a solution. Observatory tests are most beneficial to get early feedback about a design or approach, such as with a proof of concept (POC).
- **Focus group testing**: Focus group tests are limited sessions organized with a small group of users who go through the content together. These focus groups are curated to select key stakeholders and subject matter experts who can provide the best feedback about certain features or functionalities. Focus group tests can occur over multiple, interactive sessions. Focus group testing requires more time and effort than observatory testing, but can provide more detailed feedback about a solution.
- **User acceptance testing**: User acceptance testing (UAT) is a formal process where a larger group of individuals from the user community validate and provide asynchronous feedback about a solution. UAT requires the most time and effort to organize, but it's the most thorough way to conduct user testing. Once test users accept the solution and feedback issues are resolved, the content can be deployed to the production workspace.

Once you decide how to validate content, you can plan how you'll deploy it to and between workspaces.

Checklist - When planning for how to validate content, key decisions and actions include:

- ✓ **Design and document test conditions**: Describe the tests that you'll conduct, what they test, and how you'll perform them.
- ✓ **Decide on a peer review process**: Describe who else will validate the content aside from yourself.
- ✓ **Decide on an approach to manual testing**: Decide which tools and features you'll use to validate the content that you create.
- ✓ **Decide whether you'll use automated testing**: Identify whether the scale and scope of your content justifies that you set up automated tests. If so, ensure that you plan for the necessary time and resources to design and implement these tests so that they validate what you expect.
- ✓ **Deploy content from the development workspace to the test workspace**: Deploy changes from the development workspace to the test workspace so that changes are visible for users. Ensure that you've done the necessary post-deployment activities in the test workspace, such as setting up and updating a test app.
- ✓ **Decide on an approach to user testing**: Decide how users will validate content.
- ✓ **Identify test users**: Identify who among the user community will validate the content. Reach agreement with those individuals on the extent of their involvement and expectations.
- ✓ **Collect user feedback**: Set up tools and processes to automatically gather feedback. For instance, you can use Tasks and Planner in Microsoft Teams or Microsoft Forms.
- ✓ **Document test results**: Document the results of all content validation, and any changes made as a result of the test results. Ensure that this documentation is easy to find.
- ✓ **Plan your deployment to production**: Once user testing concludes, prepare to deploy content from the test workspace to the production workspace.

Related content

In the next article in this series, learn how to deploy content as part of managing the content lifecycle.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Deploy content

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

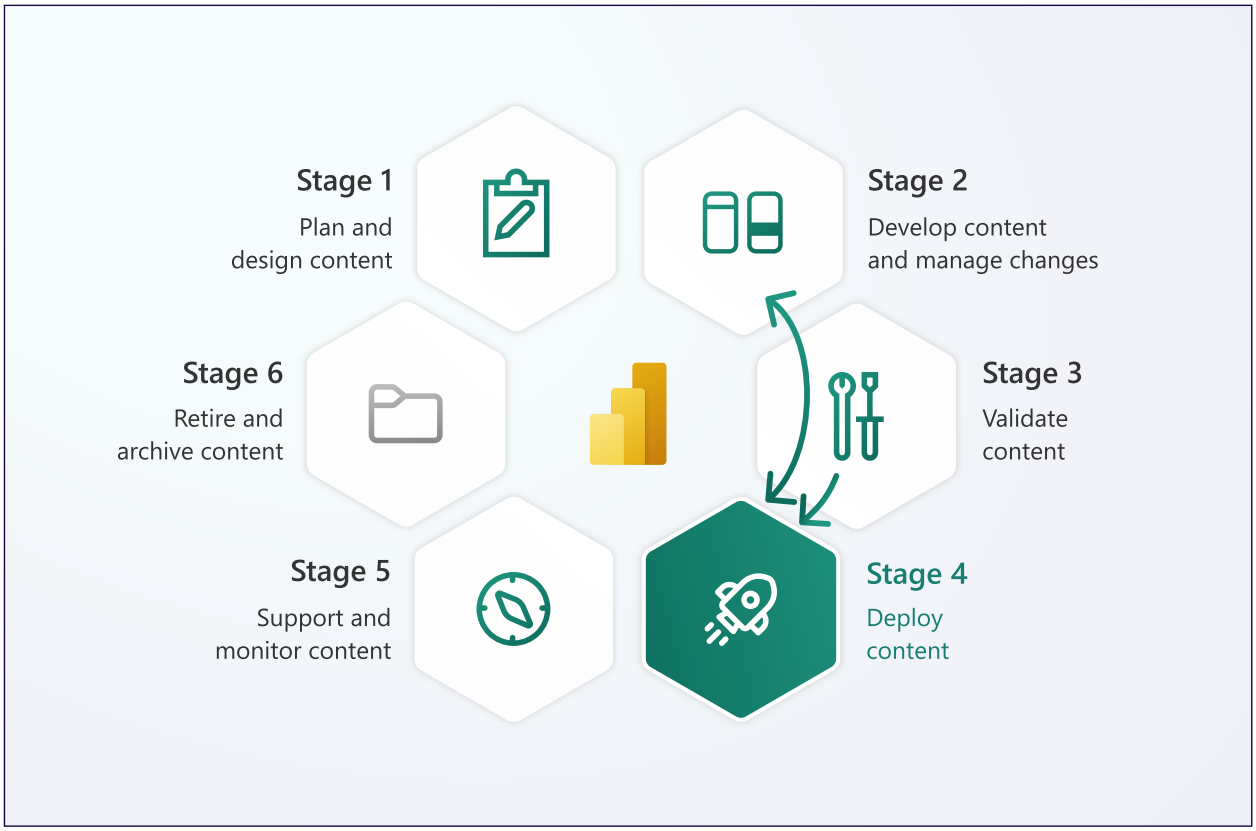
This article helps you to deploy content as part of managing the content lifecycle. It's primarily targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators might need to collaborate with other administrators, like those who oversee Microsoft 365 or Azure DevOps.
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content. These teams can also include release managers, who handle the lifecycle of content releases, and engineers who create and manage the components needed to effectively use and support lifecycle management.
- **Content creators and content owners**: Users who create content that they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the Power BI content that they create.

Lifecycle management consists of the processes and practices that you use to handle content from its creation to its eventual retirement. In the third stage of lifecycle management, you validate content changes, which involves validation performed by both content creators and users. In the fourth stage, you deploy content for consumers to use it.

To share Power BI content with consumers, you should first publish (or *deploy*) the content to a Fabric workspace. Deploying content also involves moving that content between environments, such as deploying from a development workspace to a test workspace, or from a test workspace to a production workspace.

The following image depicts the lifecycle of Power BI content, highlighting stage four, where you deploy content.



> **Note**
>
> For an overview of content lifecycle management, see the first article in this series.
>
> This article focuses on key considerations and decisions for deploying content throughout its lifecycle. For more guidance about how to deploy content, see:
>
> - Migrate to Power BI: Deploy content: This article describes key considerations and decisions for deployment when you migrate to Power BI from other technologies.
> - BI solution planning: Deploy, support, and monitor: This article describes how to plan for deployment when you first create your Power BI or Fabric solution.
> - Power BI Implementation Planning: Self-service content publishing usage scenario: This article describes how self-service users can deploy content by using OneDrive for Work and School and Fabric deployment pipelines.
> - Power BI Implementation Planning: Enterprise content publishing usage scenario: This article describes how central teams can deploy and manage content by using Azure DevOps.

You deploy content at two main points during the content lifecycle:

- When you publish content to a development workspace. At this point, you publish content to validate your changes.
- When you promote content between two workspaces (such as promoting content from a development workspace to a test workspace). At this point, you deploy content when it's ready for the next stage (such as when new content is ready to be tested).

The following sections outline approaches you can take to publish or promote content.

Decide how you'll publish content

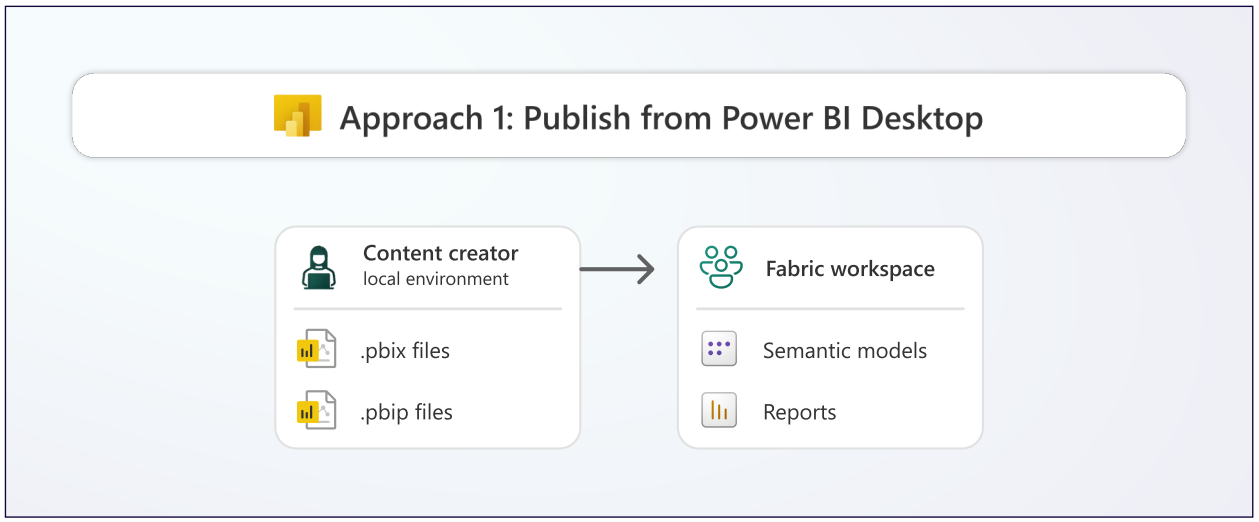
When you develop content on your local machine, you need to publish that content to a development workspace in the Fabric portal. You typically publish this content when you want to perform validation of the changes you made.

> **Note**
>
> In this article, we refer to *publishing* content as the initial deployment to the development workspace. However, in principle, publishing content is the same as deploying it.
>
> Content that's created in the Fabric portal (such as dataflows, dashboards, and scorecards) is created directly in the development workspace, and doesn't need to be published.

The following sections describe different approaches you can take to publish content.

Publish with Power BI Desktop

Power BI Desktop allows users to publish semantic models and reports from their local machine to a workspace in the Fabric portal. This approach is the simplest way to publish content; however, it can't be automated.

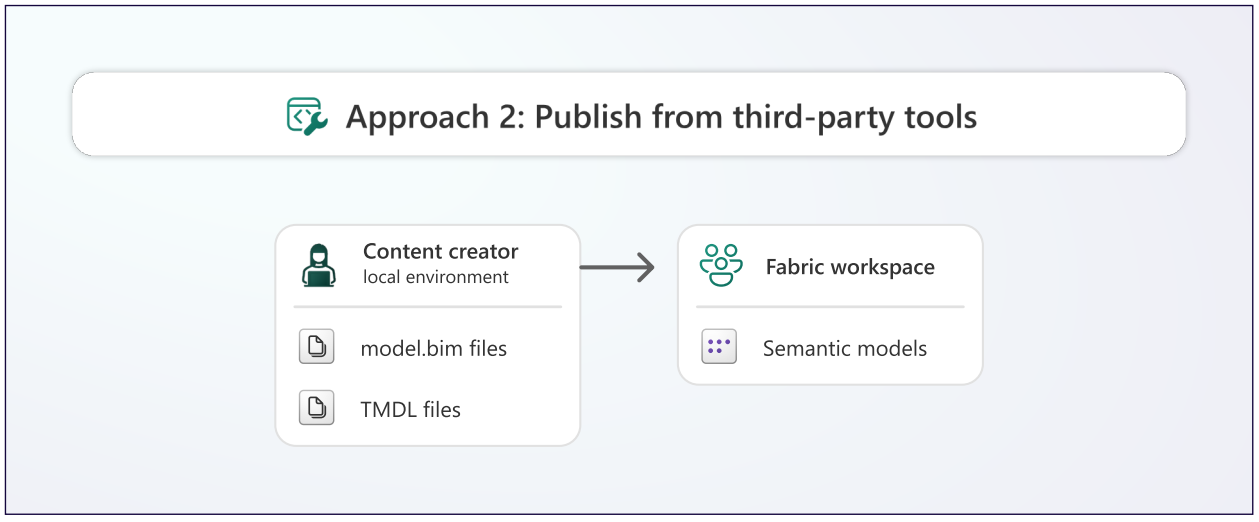


Consider using this approach when:

- Content creators prefer to manually control content publishing to the Fabric portal.
- Content creators are using Power BI Desktop to develop and manage content.
- Content creators aren't familiar with Azure DevOps or Git.
- Content comprises only semantic models or reports.

Publish with third-party tools

Third-party tools allow content creators to publish a semantic model by using the workspace XMLA read/write endpoint. For example, a content creator uses Tabular Editor to develop and manage model metadata, like TMDL (Tabular Model Definition Language) or .bim files.



> **Tip**
>
> For more information about how to use third-party tools to deploy semantic models, see the advanced data model management usage scenario.

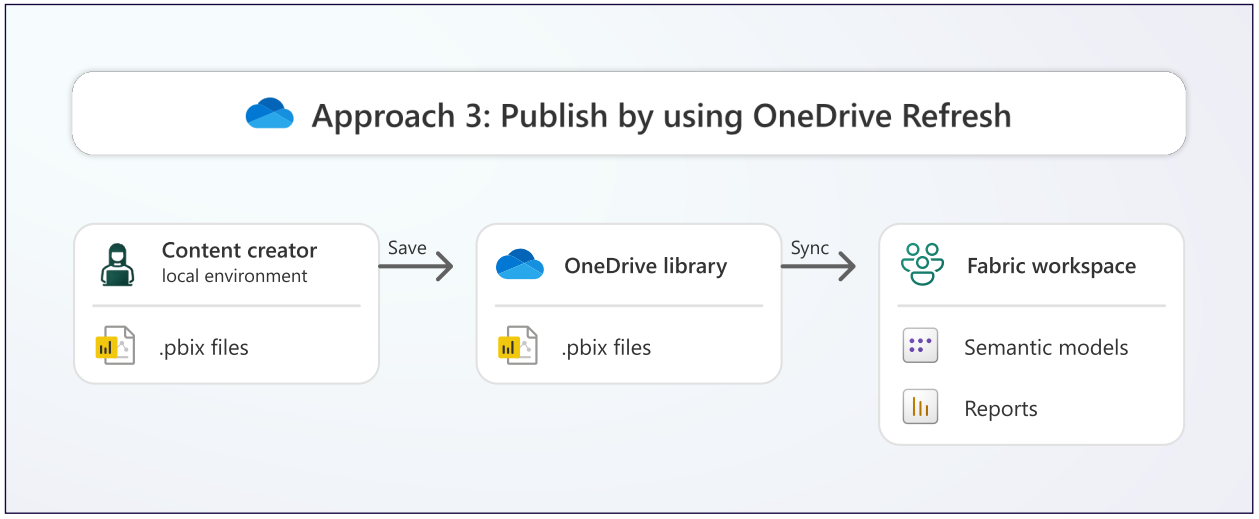
For more information about how you can enable and use XMLA read/write endpoints, see Semantic model connectivity with the XMLA endpoint.

Consider using this approach when:

- Content creators prefer to manually control content publishing to the Fabric portal.
- Content creators use a third-party tool to develop and manage the content.
- Content will be published to a workspace that uses Premium per user (PPU), Premium capacity, or Fabric capacity license mode.
- Content creators aren't familiar with Azure DevOps or Git.
- Content comprises only semantic models.

Publish with OneDrive refresh

OneDrive allows self-service content creators to automatically publish semantic models or reports to a workspace in the Fabric portal by using OneDrive refresh. Content creators can save Power BI Desktop (.pbix) files to a shared library in OneDrive. The shared library can also be a SharePoint or Microsoft Teams document library.



Tip

For more information about how to use OneDrive for Work and School with Power BI content, see the self-service content publishing usage scenario.

For more information about how you can set up OneDrive refresh, see Refresh a semantic model stored on OneDrive or SharePoint Online.

Consider using this approach when:

- Content creators want to automate content publishing to the Fabric portal.

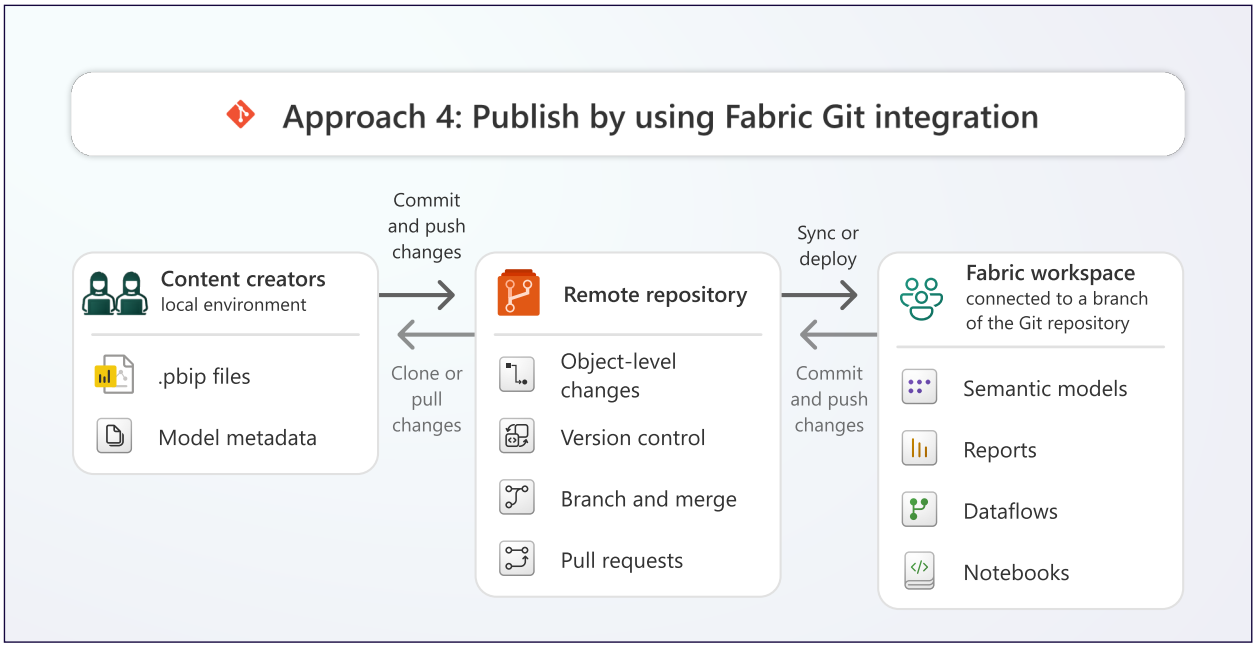
- Content creators aren't familiar with Azure DevOps or Git.
- Content creators perform version control of content by using OneDrive or SharePoint.
- Content creators save semantic models and reports as .pbix files.
- Content comprises only semantic models or reports.

Publish with Fabric Git integration

Fabric Git integration is a feature exclusive to Fabric capacities that allows content creators to sync a branch from a remote Git repository to a Fabric workspace. You can use Git integration together with Azure DevOps to sync content from Azure Repos, or you can deploy content by using Azure Pipelines (described in the next section).

> **Note**
>
> Azure DevOps is a suite of services that integrate with Power BI and Fabric to help you plan and orchestrate content lifecycle management. When you use Azure DevOps this way, you typically leverage the following services:
>
> - Azure Repos: Allows you to create and use a remote Git repository, which is a remote storage location you use to track and manage content changes.
> - Azure Pipelines: Allows you to create and use a set of automated tasks to handle, test, and deploy content from a remote repository to a workspace.
> - Azure Test Plans: Allows you to design tests to validate the solution and automate quality control together with Azure Pipelines.
> - Azure Boards: Allows you to use boards to track tasks and plans as work items, and link or refer to work items from other Azure DevOps services.
> - Azure Wiki: Allows you to share information with their team to understand and contribute to content.

To summarize, content that's been committed and pushed to the remote repository is automatically published to the workspace via this sync process. A key benefit of this approach is that it allows you to couple your source control management processes with content publication. For instance, it allows for easier rollback of changes or entire versions of a solution.



> **Tip**
>
> For more information about how to use Fabric Git integration to deploy Power BI content, see the enterprise content publishing usage scenario.
>
> For more information about how you can set up Git integration, see Tutorial: Lifecycle management in Fabric and Power BI Desktop projects: Git integration.

Consider using this approach when:

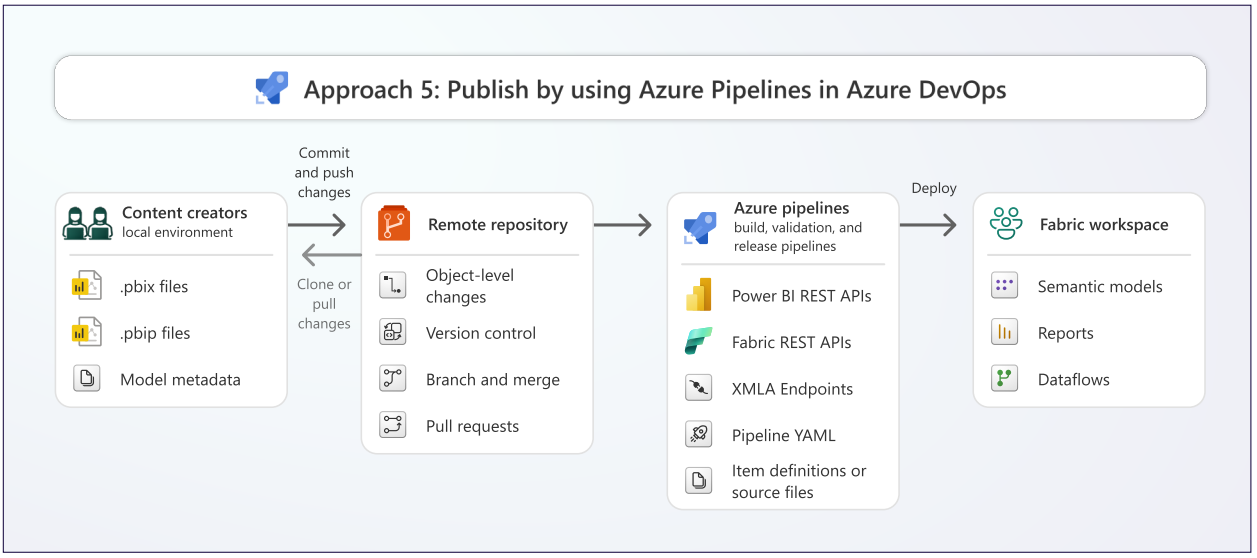
- Content creators are familiar with Azure DevOps and Git.
- Content creators are using Azure DevOps for collaboration and source control.
- Content creators save semantic models and reports as Power BI project (.pbip) files.
- Content will be published to a workspace on a Fabric capacity.
- Content is comprised of supported item types by the Git integration feature.
- Content doesn't have sensitivity labels.

> **Note**
>
> How you use Git integration to deploy and manage content is heavily dependent upon your branching and merging strategies, which you decide in stage two of lifecycle management.

Publish with Azure Pipelines

Azure Pipelines programmatically automate testing, management, and deployment of content. When a pipeline is run, steps in the pipeline execute automatically. Azure

Pipelines is more complex and requires more time and effort to set up compared to other approaches, but it allows the most control and flexibility to orchestrate the deployment process.



> **Tip**
>
> You can deploy content by using Azure Pipelines and Power BI REST APIs to workspaces that aren't on Fabric or Premium capacity. However, the Fabric REST APIs only work with Fabric, and the XMLA endpoints only work with Fabric or Premium capacity.
>
> For more information about how to use Azure Pipelines to deploy Power BI content, see the enterprise content publishing usage scenario.
>
> For more information about how you can integrate Azure DevOps with Power BI, see Power BI Desktop projects Azure DevOps integration and build pipelines.

Consider using Azure Pipelines to orchestrate content deployment when:

- Content creators are familiar with Azure DevOps and the Fabric REST APIs.
- Content creators are using Azure DevOps for collaboration and source control.
- Content creators aren't using Fabric Git integration.

Azure Pipelines and other code-based tools can programmatically deploy content by using one or more of the following APIs or endpoints:

- **Power BI REST APIs**: There are different Power BI REST API endpoints that you can use to deploy content. The Power BI REST APIs support only Power BI item types.
  - *Import*: You can publish supported items by using the Power BI REST APIs to import a valid source file to a workspace (such as a .pbix file).

    - *Deploy*: You can deploy supported items, promoting them from one workspace to another if they're stages in a deployment pipeline.
- **Fabric REST APIs**: There are different Fabric REST API endpoints that you can use to deploy content. The Fabric REST APIs support both Power BI and Fabric item types.
    - *Create*: You can create supported items by using the Fabric REST APIs together with a valid item definition.
    - *Update From Git*: You can update a workspace with content from a remote repository connected by using Git integration.
- **XMLA read/write endpoints**: You can *create* or *alter* semantic models by using XMLA endpoints together with a valid model.bim file. XMLA endpoints let you deploy changes to specific model objects instead of the entire model. Azure Pipelines can make use of third-party tools (like the Tabular Editor command line interface) to deploy semantic models by using the XMLA endpoints.

> **Tip**
>
> When you use the Fabric or Power BI REST APIs, you must first create an app registration in Azure (**described here for Power BI Embedded**). This requires a Microsoft Entra ID tenant and an organizational user, and can be a complex process to set up the appropriate permissions. However, you can execute the Fabric REST APIs in notebooks without creating an app registration. This streamlines the setup and use of APIs in your solutions, so that you don't have to manage credentials or configure any setup prior to using the APIs.
>
> To use the Fabric REST APIs without registering an app, use **semantic link** in a Fabric notebook with the **FabricRestClientClass** of **sempy** to call the API.

Together with automated testing, Azure Pipelines integration with Power BI helps you to achieve continuous integration and continuous deployment (CI/CD).

When using Azure Pipelines, pipeline owners can customize triggers, steps, and functionality to meet deployment needs. As such, the number and types of pipelines vary depending on the solution requirements.

There are three types of Azure Pipelines that you can set up to test, manage, and deploy your Power BI solution.

- Validation pipelines
- Build pipelines
- Release pipelines

> **Note**

It's not necessary to have all three types of pipeline in your publishing solution. Depending on your workflow and needs, you might set up one or more of the variants of the pipelines described in this article to automate content publication. This ability to customize the pipelines is an advantage of Azure Pipelines over the built-in Fabric deployment pipelines.

Validation pipelines

Validation pipelines perform basic quality checks of data models before they're published to a development workspace. Typically, changes in a branch of the remote repository trigger the pipeline to validate those changes with automated testing.

Examples of automated testing include scanning the data model for best practice rule violations by using Best Practice Analyzer (BPA), or by running DAX queries against a published semantic model. The results of these tests are then stored in the remote repository for documentation and auditing purposes. Data models that fail validation shouldn't be published. Instead, the pipeline should notify content creators of the issues.

Build pipelines

Build pipelines prepare data models for publication to the Power BI service. These pipelines combine serialized model metadata into a single file that's later published by a release pipeline. A build pipeline can also make changes to the metadata, like modifying parameter values. The build pipelines produce deployment artifacts that consist of data model metadata (for data models) and Power BI project (.pbip) files that are ready for publication to the Power BI service.

Release pipelines

Release pipelines publish or deploy content. A publishing solution typically includes several release pipelines, depending on the target environment.

- **Development release pipeline**: This first pipeline is triggered automatically. It publishes content to a development workspace after the build and validation pipelines succeed.
- **Test and production release pipelines**: These pipelines aren't triggered automatically. Instead, they're triggered on demand or when approved. Test and production release pipelines deploy content to a test or production workspace, respectively, after *release approval*. Release approvals ensure that content isn't automatically deployed to a test or production stage before it's ready. These

approvals are provided by release managers, who are responsible for planning and coordinating content release to test and production environments.

Decide how you'll promote content between workspaces

When you use different environments for development, test, and production, you must deploy content to all three environments. There are different tools and approaches that you can take to promote content between workspaces, depending on your specific workflow and needs.

The following sections describe approaches you can take to promote content between workspaces.

> **⊗ Caution**
>
> Avoid manually publishing content from your local machine to test and production workspaces. It can result in errors or disruptions due to mistakes. Generally, you should only publish to a development workspace, or to a **private workspace** if you use one.

Deploy with Fabric deployment pipelines

Deployment pipelines allow you to set up two or more stages (such as development, test, or production) and to deploy Fabric content between these stages. A pipeline administrator assigns a single Power BI workspace to each stage in the deployment pipeline. How you use deployment pipelines depends upon how you've decided to set up and use workspaces.

Consider using deployment pipelines when:

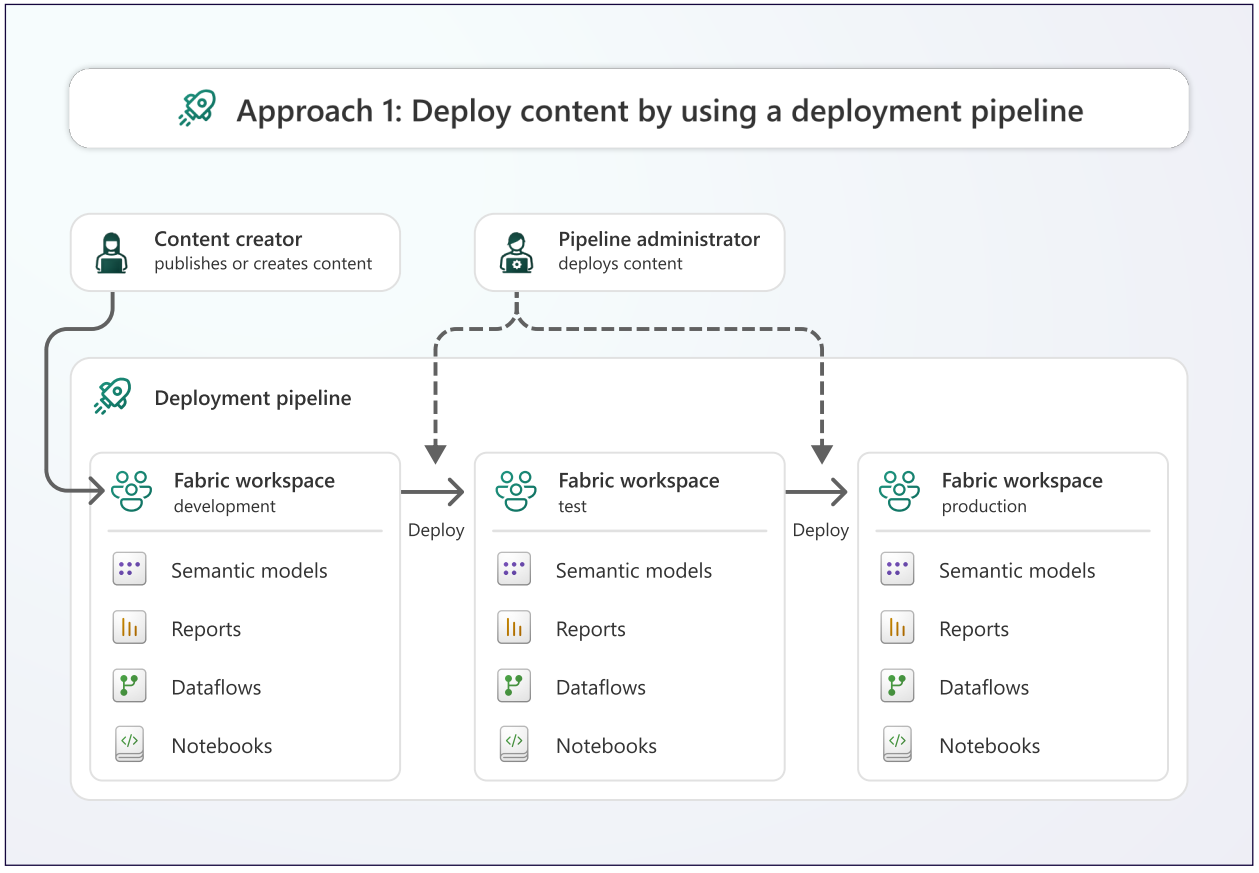
- Content is deployed to workspaces with PPU, Premium capacity, or Fabric capacity license mode.
- Content item types and scenarios are supported by deployment pipelines.

Consider another approach than deployment pipelines when:

- You prefer to deploy content from a remote repository, such as by using Azure Pipelines.
- You intend on using Git integration to synchronize different stages with different branches of your remote repository, instead of deploying the content.

> **Tip**
>
> For more information about how to use deployment pipelines to promote content between workspaces, see the self-service content publishing and enterprise content publishing usage scenarios.
>
> For more information about deployment pipelines, see Deployment Pipelines: Understand the deployment process.

The simplest way to use a deployment pipeline is when you publish all content to a single workspace and promote it to later stages within a single deployment pipeline. The following diagram depicts this first approach to deploy content by using a deployment pipeline.

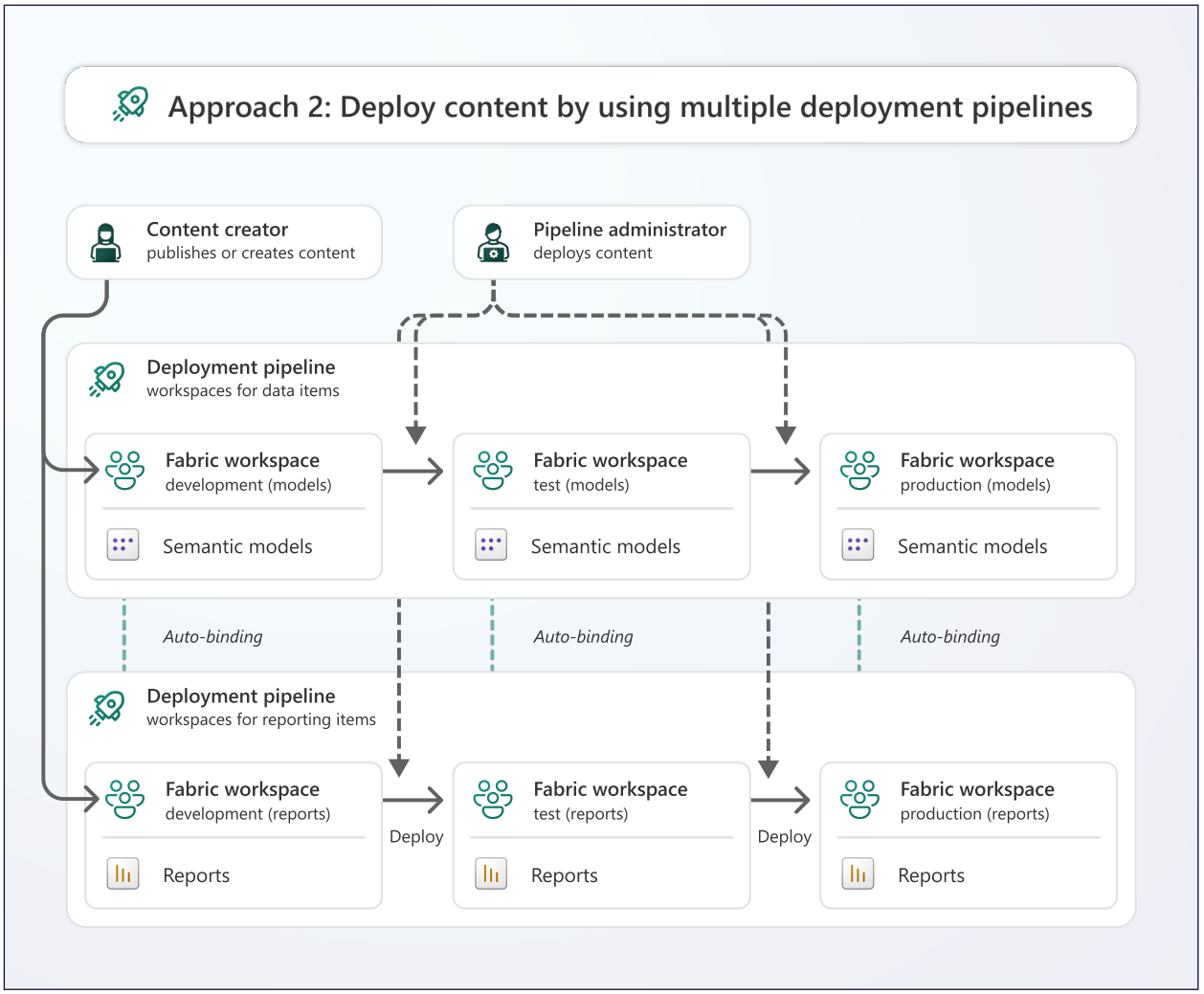


In summary, a content creator typically first publishes content to the initial stage of the pipeline. Then, to promote content to the later stages, a pipeline administrator triggers a deployment. When a deployment occurs, the deployment pipeline deploys content metadata from one workspace to the next.

When you separate content by item type in different workspaces, you'll use separate deployment pipelines to deploy this content. You can link content across workspaces with multiple deployment pipelines by using auto-binding. Auto-binding across deployment pipelines ensures that content remains linked to the appropriate item in the

respective stage. For instance, the report in the development stage will remain linked to the model in the development stage of the other deployment pipeline. However, you can also avoid auto-binding behavior if your scenario requires you link content across workspaces with a different pattern.

The following diagram depicts this second approach to deploy content by using multiple deployment pipelines.



In summary, deploying content by using multiple deployment pipelines is similar to using a single pipeline. A key difference is that you can optionally link content that's connected across workspaces and deployment pipelines by using auto-binding. Otherwise, it's identical to the first approach.

Deployment pipelines are a flexible, straightforward tool suitable to improve your content lifecycle management for both self-service and enterprise scenarios.

Access to both the workspace and the deployment pipeline is required for the users conducting a deployment. We recommend that you plan deployment pipeline access so that pipeline administrators can view deployment history and compare content. When collaborating with multiple content creators, consider restricting pipeline access to

release managers or technical owners who are best suited to oversee the deployment and release processes.

Additionally, consider using deployment rules to set different configurations for items in different stages. For instance, you might want a semantic model in the development workspace to source data from the development database, while the semantic model in the production workspace sources data from the production database.

> **Tip**
>
> If multiple people have access to your deployment pipeline, we recommend that you review the **deployment history** regularly. These reviews can help you identify unapproved deployments or deployment failures.
>
> If you're using **auto-binding** to link items across deployment pipelines, ensure that you also review item lineages to identify breaks in auto-binding caused by someone publishing linked content to the wrong stage.

You can trigger deployments manually or programmatically by using the Power BI REST APIs. In both cases, you should define a clear and robust process about when you promote content to each stage, and how to roll back unintended changes.

Manually perform deployment

You can deploy content manually by using the Fabric deployment pipeline. You can choose deploy all content or select items. Selective deployment can be beneficial when some content is ready to move to the next stage, but some items are still undergoing development or validation. Additionally, you can perform a backwards deployment when content changes exist in a later stage but not in an earlier one.

> **Caution**
>
> When using deployment pipelines, we recommend that you deploy content in a single direction, such as from development to test to production workspaces. Typically, you should avoid making changes to content in later stages before those changes have undergone the appropriate validation in development or test.

When you conduct a manual deployment, you can compare stages to identify content changes in the *change review* window. This approach is especially useful when you don't use a Git remote repository for source control.

Use the Power BI REST APIs to perform deployment

You can use the Power BI REST APIs to deploy content by using a deployment pipeline. A benefit of using the REST APIs is that you can automate deployment and integrate it with other tools, like Azure Pipelines in Azure DevOps.

Deploy with Azure Pipelines

Azure Pipelines allow you to orchestrate deployment between all stages. With this approach, you use the Fabric REST APIs to deploy and manage content, making use of different Azure Pipelines such as validation and release pipelines.

Consider using Azure Pipelines when:

- You want to centralize orchestration of deployment from within Azure DevOps.
- Content creators are using Azure DevOps to collaborate and for source control.

Consider another approach than Azure Pipelines when:

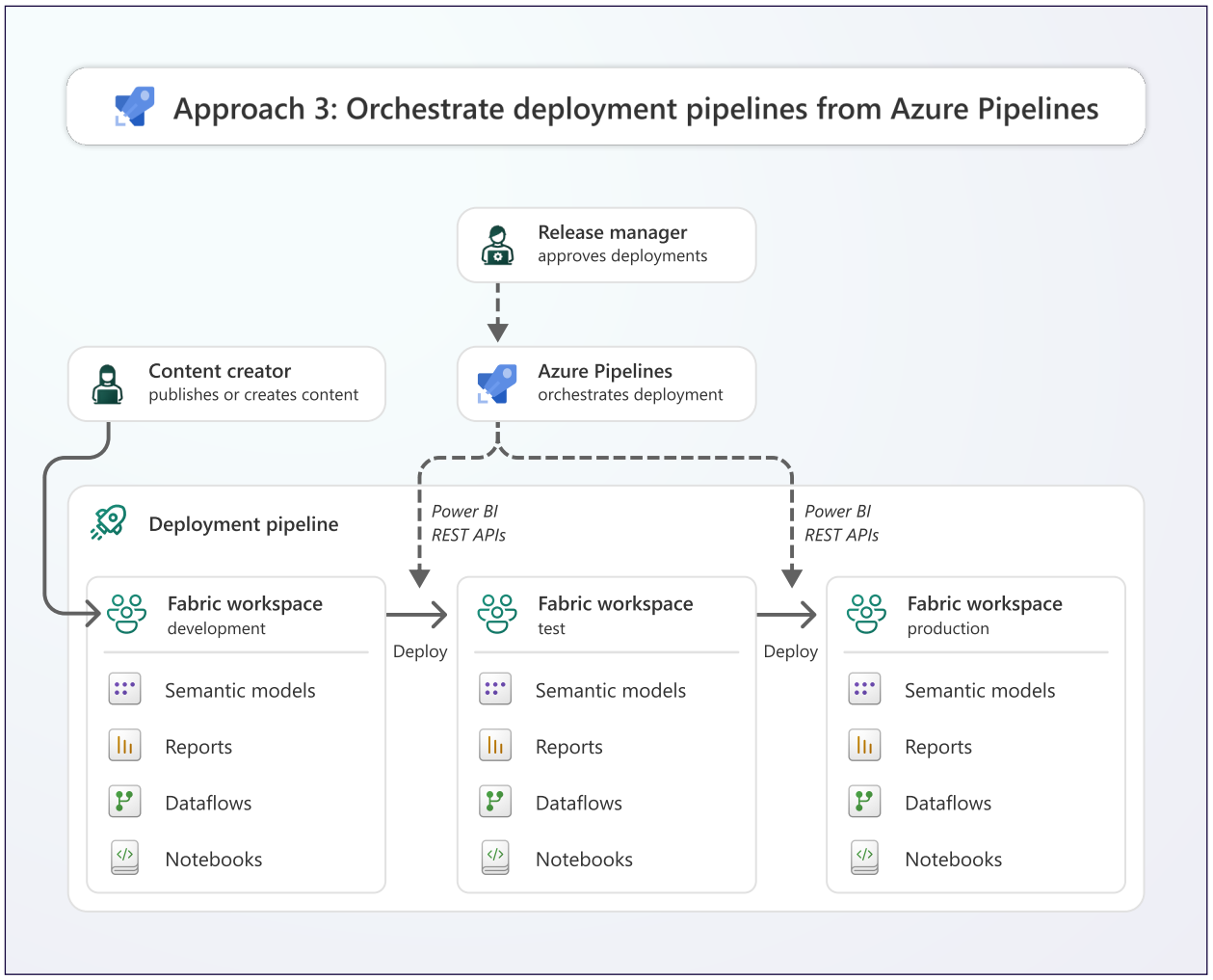
- Content creators aren't familiar with Azure DevOps or code-based deployments.
- Content includes item types that don't have a supported definition or source file format, like dashboards.

There are two different approaches to deploy content with Azure Pipelines. Either they orchestrate deployment pipelines, or they deploy content to a workspace without a deployment pipeline.

Orchestrate Fabric deployment pipelines by using Azure Pipelines

In this approach, release pipelines orchestrate content deployment to test and production workspaces by using deployment pipelines. Content is promoted through development, test, and production workspaces in Fabric.

The following diagram depicts how you orchestrate deployment pipelines from Azure Pipelines.

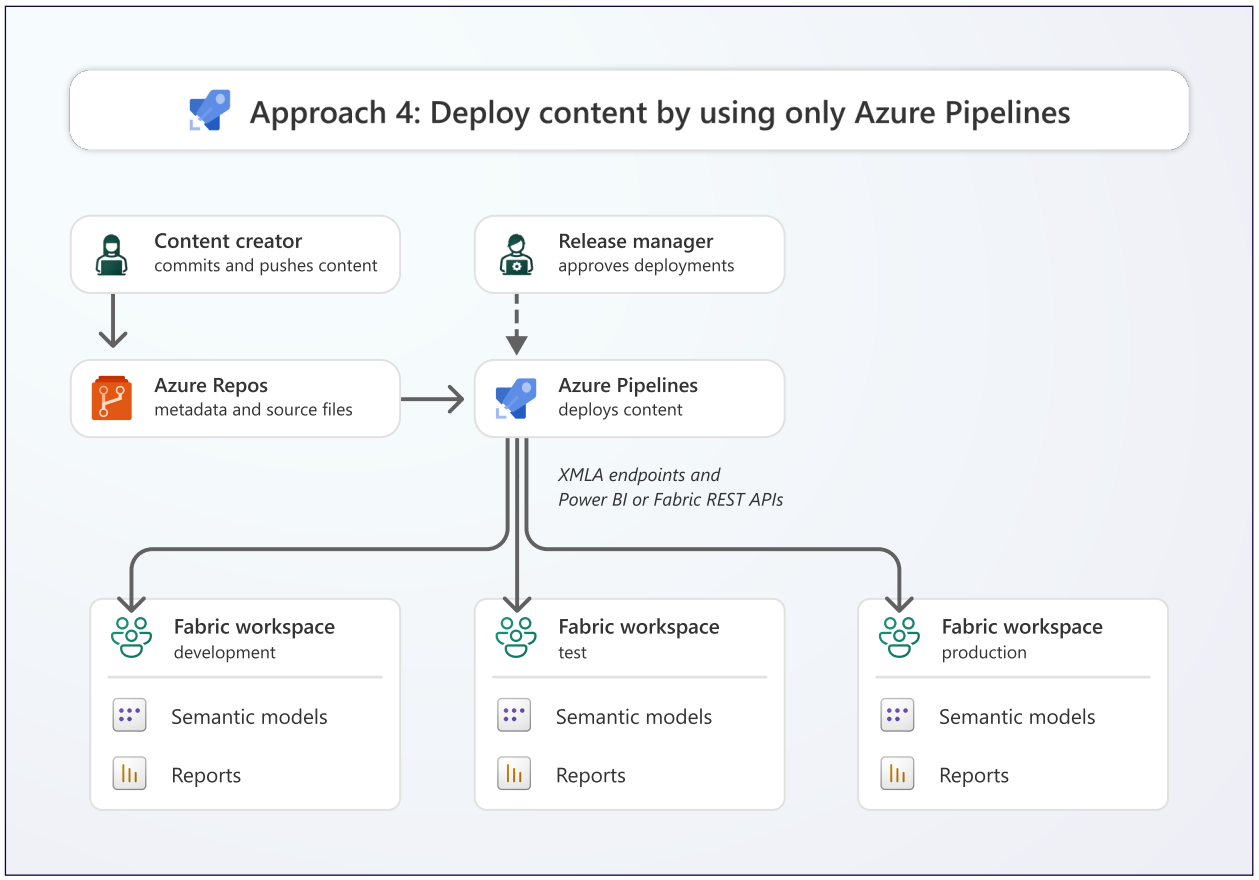


In summary, content creators publish content to the workspace in the first stage of the deployment pipeline. Then, a release manager approves the deployment, which triggers an Azure Pipeline. This pipeline uses the Power BI REST APIs to promote content between stages, so that the metadata is deployed to another workspace. One advantage of this approach is that you can orchestrate deployment of multiple Fabric item types through deployment pipelines, because some item types are developed in the Fabric portal and thus can't be deployed by Azure Pipelines alone.

Deploy content by using only Azure Pipelines

You can also deploy content to a workspace from Azure DevOps by using Azure Pipelines. This approach doesn't use deployment pipelines. Instead, it uses release pipelines to deploy source files or metadata files by using either the Fabric or Power BI REST APIs, or XMLA read/write endpoints. Typically, these files are stored in an Azure Repos Git repository.

The following diagram depicts how you deploy content by using only Azure Pipelines.



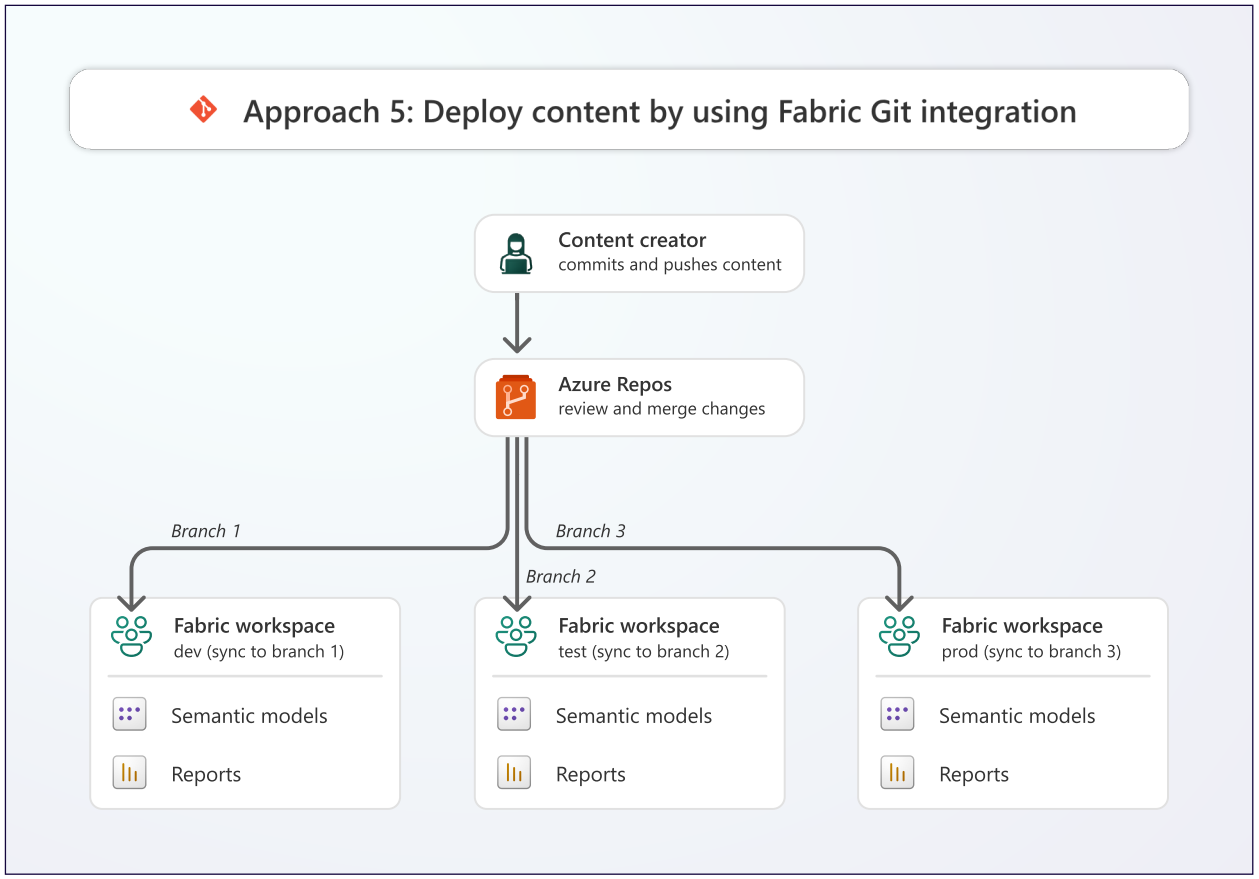
In summary, content creators commit and push content changes to a remote Git repository in Azure Repos. This content is used by Azure Pipelines for the deployment. Once the release manager approves the specific deployment, the Azure Pipeline will deploy content to the workspace, either by using the Power BI REST APIs (that is, for .pbix files), the Fabric REST APIs (that is, for item definitions), or XMLA endpoints (that is, for model.bim files). A separate Azure Pipeline exists for each workspace.

This approach doesn't require Fabric capacity or Premium licensing when you publish only Power BI Desktop files with the Power BI REST APIs. However, it does involve more setup effort and complexity because you must manage deployment outside of Power BI. Development teams that already use DevOps for data solutions outside of Power BI might be familiar with this approach. Development teams that use this approach can consolidate deployment of data solutions in Azure DevOps.

Deploy with Fabric Git integration

When you use Git integration, you can sync different branches to different workspaces instead of publishing or deploying content, explicitly. That way, you can have separate branches for development, test, and production workspaces. In this scenario, the *main* branch syncs with the *production* workspace. You then deploy content between workspaces by performing a pull request to merge the development branch into the test branch (to deploy to the test workspace) or to merge the test branch into the main branch (to deploy to the production workspace).

The following diagram depicts how you deploy content by using Fabric Git integration to sync branches to different workspaces. For simplicity, the diagram doesn't include details of branching or merging content.



In summary, content creators commit and push content changes to a remote Git repository in Azure Repos. Content creators open pull requests (PRs) to request to merge their changes into a specific branch. Depending on the branching strategy, different branches are connected to different workspaces. Once changes are merged into a branch, content creators sync the workspace with the remote Git repository to view the latest changes to the content in that workspace.

Consider this approach when:

- You want to orchestrate deployment between workspaces by using your branching and merging strategy.
- You don't intend to use Azure Pipelines or Fabric deployment pipelines to orchestrate deployments to test and production.
- The workspace doesn't contain unsupported items or scenarios.
- Content doesn't have sensitivity labels.

> **Note**
>
> There are many valid ways to deploy content. For instance, you can use a combination of the different approaches discussed in this article.

For instance, you can deploy content to a development workspace by using an Azure Pipeline, which allows you to benefit from continuous integration features and conduct automated testing (such as using the Best Practice Analyzer). Then, you can deploy content between workspaces by using either Git integration or a Fabric deployment pipeline.

Choose the approach that best fits your needs and the way your team works.

Decide how to handle post-deployment activities

After deployment, there are various post-deployment activities that must be handled. Many of these activities can be handled programmatically, for instance by an Azure Pipeline or notebook and the Power BI and Fabric REST APIs. For example, you can programmatically set data source credentials, manage scheduled refresh, and trigger refreshes after a metadata deployment. However, some tasks require manual intervention, such as a first-time setup or updating a Power BI app.

Ensure that you identify all relevant post-deployment activities for your content, and that you decide how they'll be handled.

Once you've planned how you'll deploy content, you should next consider how you'll support and monitor it.

Checklist - When planning how to deploy content, key decisions and actions include:

- ✓ **Identify the deployment options available to you**: Depending on your licensing and content, you'll have different options available to publish content or promote it between workspaces. Identify whether you can use deployment pipelines, Azure DevOps, Git integration, the Fabric REST APIs, and XMLA read/write endpoints.
- ✓ **Decide how you'll publish content**: Choose an approach to publish content that best suits your workflow and needs. Ensure that this approach aligns with your other strategies, like how you track and manage changes.
- ✓ **Decide how you'll promote content between workspaces**: Choose an approach to deploy content from development to test workspaces and from test to production workspaces. Ensure that this approach aligns with your other strategies, like how you'll publish content.

- ✓ **Plan your release strategy**: Determine whether a specific individual will be responsible for the final review of content before approving a release or deployment. Ensure that this individual is aware of this task and what they should do to safeguard the deployment process without blocking progress.
- ✓ **Plan post-deployment activities**: Ensure that you've decided on a process to perform activities such as updating a Power BI app or refreshing data items after a metadata deployment. Consider automating this process by using the Fabric REST APIs.
- ✓ **Perform first-time set up of deployment tools and processes**: Ensure that you set up the appropriate access, and that permissions align to how you set up access to your content.
- ✓ **Deploy content to production**: When you've planned and set up your deployment, deploy content to production.

Related content

In the next article in this series, learn how to support and monitor content as part of managing the content lifecycle.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Support and monitor content

Article • 03/13/2025

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

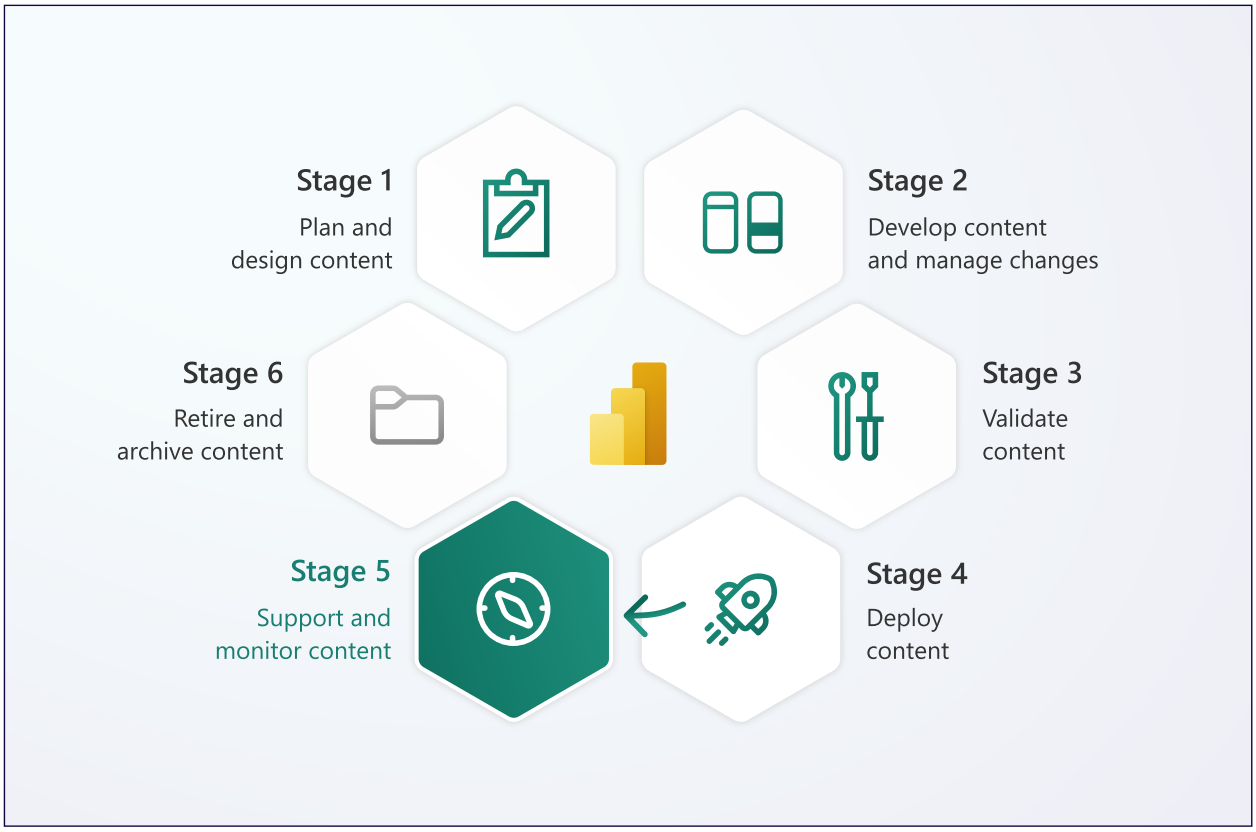
This article helps you to support and monitor content as part of managing the content lifecycle. It's primarily targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators might need to collaborate with other administrators, like those who oversee Microsoft 365 or Azure DevOps.
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content. These teams can also include release managers, who handle the lifecycle of content releases, and engineers who create and manage the components needed to effectively use and support lifecycle management.
- **Content creators and content owners**: Users who create content that they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the Power BI content that they create.

Lifecycle management consists of the processes and practices that you use to handle content from its creation to its eventual retirement. In the fourth stage of lifecycle management, you deploy content, which involves both initially publishing content to a development workspace and promoting content to a test and production workspace. At the end of the fourth stage, content consumers already use your content. In the fifth stage, you support this content and monitor usage to facilitate adoption and resolve issues.

Supporting and monitoring content is vital to ensure effective adoption of your solutions. Support refers to how you'll enable content viewers and consumers to effectively make use of the content you deploy. Monitoring refers to both auditing and monitoring activities that you perform to track and measure your solution, as well as how it's used by content viewers and consumers.

The following image depicts the lifecycle of Power BI content, highlighting stage five, where you support and monitor content.



> **Note**
>
> For an overview of content lifecycle management, see the first article in this series.
>
> This article focuses on key considerations and decisions to support and monitor content throughout its lifecycle. For more guidance about how to support and monitor content, see:
>
> - Fabric adoption roadmap: User support: This article describes general considerations and guidance for supporting users of Fabric and Power BI.
> - Power BI implementation planning: Auditing and monitoring: This series describes in detail the options available for you to audit and monitor your content.
> - BI solution planning: Deploy, support, and monitor: This article describes some considerations to support content when you first create your Fabric or Power BI solution.
> - Migrate to Power BI: Support the solution: This article describes some considerations for how to best support a Power BI solution during a migration from other technologies.

At this point, content has been deployed and is being used by consumers. Your next action is to sufficiently support the content throughout its lifecycle.

How you handle support depends significantly on who owns and manages the content. Enterprise BI content is likely to have the most robust support model. However, business-led and self-service content created by decentralized teams should also be supported, although the people and processes involved might differ. Make sure it's clear to all authors what their obligations are for supporting the content that they publish.

> **Tip**
>
> If you expect your data and BI environment to thrive, don't underestimate how important it is to support production (non-development, non-test) content.

The following sections outline some key considerations and decisions for how you'll support and monitor content.

Decide on your support model

There are many valid ways to define a support model. Options range from informal intra-team and internal community support to the more formally organized help desk and extended support. However, it's common to use more than one method to support users. For more considerations about planning for the different types of user support, see User support.

> **Tip**
>
> The level of support that's required might also depend on the **governance** level that you've selected. Make sure to involve your **Center of Excellence** in decisions about support.

There are two main types of users to consider when planning your support model.

- **Consumer support**: The focus is on supporting delivery of accurate and timely content to *consumers*, or *view-only users*, who consume content that's produced by others in the organization.
- **Content creator support**: The focus is on supporting the needs of self-service BI *creators* who design, publish, secure, and manage content that other users consume. These self-service BI creators consume the solution that you've delivered to create other content.

> **Note**
>
> There are different topics important to plan your support model.
>
> - User tools and devices: Plan how content creators and consumers can get help to install and use tools to make and view content.
> - Mentoring and user enablement: Plan how you can improve the competencies of the user community so they can create and use content effectively.
> - User support: Plan how you can resolve issues for users with both internal and external support channels.

The remainder of this section introduces considerations for both content consumers and creators.

Content consumer support

Consider the following points when supporting consumers who only need to view content.

- **How to get help**: How will the consumer know who to contact for help? A few common situations include:
  - Having a strategy for read-only consumers.
  - Creating a request access workflow for consumers.
  - Addressing critical issues that block productivity, such as a data refresh, DirectQuery report connectivity failure, or when data-level security isn't set up properly for a user.
  - Addressing noncritical issues that aren't time-sensitive, like report issues with a bookmark or a button.
  - Resolving data discrepancies.
- **How to use the solution**: What sort of assistance is available to help consumers use and fully understand the solution? An overview page within the report or a short video tutorial can be helpful for users (for example, it could show how to drillthrough to another report). Providing that type of help can lead to increased adoption of the solution, greater return on investment, and fewer support cases.
- **How to accept feedback**: When the consumer has a new request or enhancement, how should they submit their request? A feedback loop (such as a form) allows you to accept ideas for improving the solution. Feedback loops are often considered alongside support planning because it's a closely related concept.

- **How to use data**: How can users leverage tools and features like personalize visuals, Copilot, or explorations to explore data from read-only experiences? When should they request build permissions if they need to make their own content, and how should they make this request? What training is available to support people in these features, or to transition from read-only consumers to individuals who might build and share their own content?

> **Note**
>
> Expect that once a solution is deployed to production, you'll receive different types and volumes of feedback compared to that which you received during validation. Anticipate that there will be higher volumes of requests and feedback during this *hypercare* period (the period immediately after a major change). Plan accordingly for this higher volume, and try to see it as an opportunity to build trust with the user community.

Content creator support

Consider the following points when supporting both data creators and report creators who need to create, publish, and maintain content.

- **How to publish new content**: How will the author know who to contact for help when they need to publish new content? The extent of help that they'll need depends on the tenant settings, and the author's workspace role. A few common requests that often arise include:
  - Managing who can create new workspaces and the workspace creation process.
  - Creating a deployment pipeline or getting access to Azure DevOps (see also Stage 4: Deploy content).
- **How to update security**: What's the process for an author to request security updates? The permissions for content might need regular attention, depending on how frequently users change. Several common actions include adding and removing users, security groups, and service principals from:
  - Workspace roles
  - Power BI app permissions and Per-item permissions
  - OneLake security
  - Files in OneDrive and SharePoint and Power BI files that consumers view from OneDrive and SharePoint
- **How to get access to new functionality**: What should an author do when they need to use functionality that they don't currently have access to? A few common situations include:

    - Workspace license mode
    - Capacity management and (SKU) size.
- **How to update data connectivity**: If an author needs access to a gateway or a connection, what's the procedure they should follow to request updates (if they don't have permissions)? Common situations include:
    - Adding a new data source connection
    - Adding a new user for a data source connection
- **How to use AI and improve productivity**: What should an author do to ensure that they remain effective and efficient? How should they best use Copilot and other assistive AI tools to augment their work without creating risk or wasting time?
    - Updating a data model to work well with Copilot

> **Tip**
>
> Be prepared to address the most urgent support issues quickly to allow users to continue being productive. When they're blocked, a user might find a workaround (which might not be ideal) to keep moving, especially when there's a slow response from their support channel.

Once you define your support model for both content consumers and creators, you next should plan how you'll support content. Typically, this involves auditing and monitoring content regularly.

Decide how to audit and monitor content

Auditing and monitoring of content throughout its lifecycle is important. Creating processes to audit content can help you in various ways.

- **Assess solution adoption**: You should regularly analyze two aspects of adoption:
    - **Usage of the solution**: Analyzing *content usage levels* involves understanding what content is used, when, by whom, and how. For more information about the types of questions you might ask, see Content usage.
    - **Effective use of the solution**: Analyzing the *effective use of content* is more difficult to define and measure (vs. usage). You can further examine the user activity data to determine whether actual user behavior matches your expectations. For example, if you expect a high number of report views per day but what you see is one export per user per day, you should investigate further.
- **Understand published items**: You can document the metadata for published items as of a point in time. For example, you can create a *tenant inventory* as of January 1

that might indicate that 12 semantic models and 94 reports existed then. For more information about the benefits of a tenant inventory, see Understand published items.

- **Reduce risk**: Because you understand more about what's happening with your content, you can act quickly to reduce risks. For example, you might identify security or data protection concerns with the content. For other considerations, see Mitigate risk.
- **Find performance and health concerns**: When auditing content, you'll find data that can be used as an input to performance tuning activities. It's also possible that you can resolve an issue before it's noticed by consumers of the content. For more information, see Performance monitoring.

> **Tip**
>
> There are many planning considerations and decision-making criteria related to auditing and monitoring that are covered in other articles. The scope for your auditing and monitoring activities could differ depending on your focus and your job role. For more information, see:
>
> - Report-level auditing for auditing and monitoring reporting items, like paginated reports.
> - Data-level auditing for auditing and monitoring data items, like semantic models.
> - Tenant-level auditing for auditing content across your entire tenant.

Checklist - When planning to support and monitor content, key decisions and actions include:

- ✓ **Assess how user support is handled**: Investigate how support is currently handled to become familiar with the current state.
- ✓ **Decide how to handle support**: Determine how you'll support the content throughout its lifecycle. Decide how to support both consumers and authors.
- ✓ **Clarify support roles and responsibilities**: Confirm who's expected to do what, and when, for supporting content after it's published. Clarify specific expectations for content owners (including self-service authors). Also clarify expectations for the Center of Excellence (COE), IT team, BI team, and help desk staff.

- ✓ **Create a support model for consumers**: Create specific processes and supporting documentation targeted at supporting content consumers.
- ✓ **Create a support model for content creators**: Create specific processes and supporting documentation targeted at supporting authors.
- ✓ **Set up a support tracking system**: Create a system to accept and track the status of requests from users.
- ✓ **Train the support staff**: Conduct knowledge transfer sessions or training sessions to prepare the help desk staff.
- ✓ **Communicate the support model to users**: Make sure your consumers and authors know what resources are available to them, and what processes to follow. In the Fabric tenant settings, customize the help menu links to direct users to your internal resources.
- ✓ **Determine specific data needs for auditing and monitoring**: Consider what report-level, data-level, and tenant-level auditing and monitoring activities should be conducted.
- ✓ **Prioritize auditing efforts**: Identify several key questions to be answered. Determine the focus and priorities for your initial auditing and monitoring efforts.
- ✓ **Clarify auditing roles and responsibilities**: Confirm who's expected to do what, and when, for auditing content. Clarify specific expectations for content owners (including self-service authors), as well as the COE and IT.
- ✓ **Make initial auditing architecture decisions**: Consult with your IT team(s) as appropriate to determine whether existing processes exist to extract and store auditing data. When possible, access existing data rather than creating a new process. When a new process is required, consider architecture decisions for extracting, storing, and securing the data.
- ✓ **Create initial auditing analytical reports**: Identify the requirements and audience for each auditing report. Create reports that are clear about purpose, what action should be taken, and by whom. Obtain feedback and iterate as appropriate.
- ✓ **Create documentation and training for content authors**: Provide guidance for both administrators and content creators relevant to the support and auditing capabilities that are applicable to their job role. Include recommendations and information about how to meet internal auditing requirements for support and auditing.

Related content

In the next article in this series, learn how to retire and archive content at the end of its lifecycle.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Retire and archive content

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to retire and archive content at the end of its lifecycle. This article is primarily targeted at:

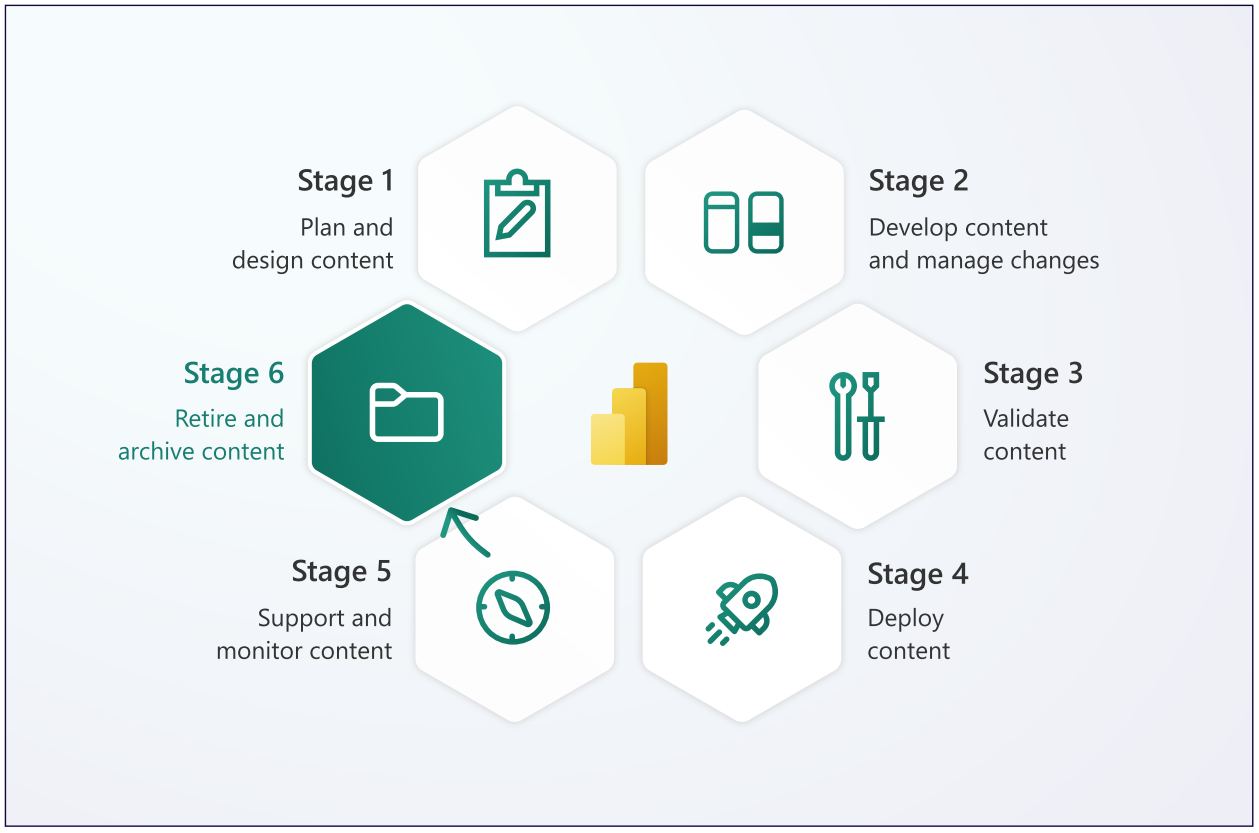
- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators might need to collaborate with other administrators, like those who oversee Microsoft 365 or Azure DevOps.
- **Center of Excellence (COE) and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams include decision makers who decide how to manage the lifecycle of Power BI content.
- **Data operations teams**: The teams that are responsible for overseeing the lifecycle management of enterprise data solutions. These teams can include release managers, who handle the lifecycle of content releases, and engineers who create and manage the components needed to effectively use and support lifecycle management.
- **Content creators and content owners**: Users who create content that they want to publish to the Fabric portal to share with others. These individuals are responsible for managing the lifecycle of the Power BI content that they create.

Lifecycle management consists of the processes and practices that you use to handle content from its creation to its eventual retirement. In the fifth stage of lifecycle management, you support and monitor content, which involves supporting users and published content to facilitate adoption and address issues. Eventually, you'll come to the sixth and final stage of the content lifecycle, where you might identify content that's no longer used or needed. At this last stage, you retire and archive the content.

Retiring and archiving content is important once the content has fulfilled its purpose. It ensures that resources aren't wasted supporting content that isn't needed, and it makes it easier for content creators and administrators to oversee the content that is needed. When you retire and archive content at the end of its lifecycle, it helps you to improve

governance and efficiency, because you focus effort and resources on relevant content and active areas of your tenant.

The following image depicts the lifecycle of Power BI content, highlighting stage six, where you retire and archive content.



> **Note**
>
> For an overview of content lifecycle management, see the first article in this series.

Eventually, consumers might no longer need or use content. There might be many reasons, such as replacement content being deployed, different business objectives, or changing priorities.

Retiring content helps you govern Power BI in different ways.

- Fewer items reduces clutter and keeps your environment more manageable. Fewer items also helps users to navigate active content, and it's easier to maintain.
- Removing unused items reduces confusion for consumers about what content they should use.
- Auditing becomes easier when it's focused on active content. That's especially true when you audit security roles and permissions.
- Having fewer items can potentially optimize capacity usage because resources won't be used accidentally.

Decide when and how you'll retire content

You should regularly evaluate usage patterns in your organization as you perform auditing and monitoring activities to identify candidate content you can retire. The steps are known as *content clean-up activities*. Content clean-up activities are the regularly scheduled actions you take to survey content, whereas content retirement is what happens when you decide to decommission content at the end of its lifecycle. Depending on the volume of items published in your tenant, you might conduct these activities more regularly, such as quarterly or monthly.

> **Tip**
>
> Tenant-level auditing and tenant-level monitoring are key activities to help you decide when content should be retired.

Plan content cleanup activities

The following key considerations and decisions should be made when you want to analyze your content to identify what could be retired and archived.

- **Select the scope of analysis**: Content clean-up activities can be done at the tenant level. However, it's also common to conduct an analysis at the workspace, capacity, or domain level. That way, you can make decisions and take action based on specific needs or user populations. For instance, workspace administrators can be responsible for regular content clean-up activities of the workspaces that they manage.
- **Decide on a time window for the specific content**: Consider what time window is relevant for the content you're analyzing. Although you might have a standard time window (such as no usage within three months), it can be difficult to use the same guidelines in every situation. Consider the following examples.
  - **Standard 3-month time window**: A workspace for the sales team exists to track quarterly bonuses. If there's been no usage within the past three months, the entire workspace is a candidate for retirement.
  - **Shorter 1-month time window**: An operational report with daily sales figures is intended for frequent daily or weekly use. If it hasn't been used within the past month, the report is a candidate for retirement.
  - **Longer 1-year time window**: A semantic model is produced by IT once a year to provide reports for external auditors. Although it doesn't have any usage in the most recent few months, it's critical to keep the semantic model. In this

situation, a time window of more than one year is justified before making it a candidate for retirement.
    - **Exception with no time window**: A scorecard containing key organizational key performance indicators (KPIs) is used intermittently by executives. Even though the usage isn't consistent and it has a few viewers, this specific content won't be considered for retirement based solely on low usage.
- **Create classifications**: Decide what terminology you'll use consistently. Create *common definitions* to classify terms such as frequently used, actively used, occasionally used, and unused. Consider how you'll apply these terms for analyzing content usage and user activity levels. For more information, see Create classifications.
- **Select relevant usage metrics**: In most situations you'll need to consider multiple metrics, including total actual views for content and total actual users (viewers) of content. Consider the following examples.
    - **High number of views**: One report might consistently have a high number of views from a small number of users. This activity might reflect personal BI and team BI usage scenarios.
    - **Large number of users**: A report might have many viewers from various areas throughout the organization. Although the content is used broadly, the actual views might be sporadic and inconsistent. This activity might reflect an enterprise BI usage scenario.
    - **Potential vs. actual viewers**: A Power BI app or a report might have permissions assigned so that there are many potential viewers. However, you might see actual usage from only a small percentage of potential viewers.
- **Determine who will conduct usage analysis and cleanup**: Based on how content ownership and management is handled in your organization, consider who should be responsible for content clean-up activities.

Once you plan how you'll perform a content clean-up analysis, you should next identify what content you'll retire.

Identify the content you'll retire

At this point, you've planned regular content clean-up activities, and made some key choices and decisions about how to analyze the content in your tenant. It's now time to analyze usage patterns and identify candidate content to retire. Typically, you want to find content with low usage and maintenance to investigate.

Consider the following actions for finding content that's sparsely used and no longer maintained.

- **Analyze views**: You should focus on the number of views for visual items, such as reports and apps.
  - Sort the items by the total views in your time window (described in the previous section). Focus on finding items that are unused or only occasionally used.
  - Analyze the recent usage based on your time window.
  - Check for a downward trend compared to past time periods.
- **Analyze queries**: You should focus on queries submitted for data items, such as semantic models.
  - Sort the items by the total queries in your time window. Focus on finding content that's unused or only occasionally used.
  - Analyze the recent usage based on your time window.
  - Check for a downward trend compared to past time periods.
- **Analyze the viewers**: Sometimes, the total number of viewers of the content in your time window can be relevant when it's also compared to the views and queries.
  - Sort the items by the total number of viewers in your time window.
  - Filter out viewers who are content creators, owners, or support team members so you can focus on consumers with view-only activity.
  - Filter out viewers who have used the report only once or twice to see the effect on the usage trend. It's possible that these users accidentally opened a report.
- **Analyze refresh dates**: For data items (such as semantic models or dataflows), check the last time the content was refreshed. If it wasn't refreshed recently, it probably supports the conclusion that the data items aren't used anymore.
- **Check content update dates**: It might be helpful to check the last time the content was updated or republished. Content that hasn't been updated recently supports the possibility that it might be ready for retirement.
- **Check for documentation**: Before exclusively relying on the usage statistics data, review any documentation or metadata that's available.
  - Locate documentation that exists for the content. Ideally, a report would include an introduction page to document its target usage and target audience. That type of information helps viewers, and it's especially useful when the content is used on an irregular basis.
  - Review available metadata. A workspace description could contain relevant information about its intended usage patterns. A workspace name might include *[Dev]* or *[Test]* suffixes that would clarify why usage varies considerably.
- **Communicate with the owner of the content**: When you find a candidate item to retire, inquire about the content with the contact or its owner. They might have other insight into expected usage patterns that's not readily apparent from the usage statistics data. Sometimes you might also need to speak with subject matter experts or consumers of the content to understand their future needs.

> **Tip**
>
> The **Get Unused Artifacts as Admin** REST API is one way to find unused items. However, be aware that the API searches the past 30 days of history. In most situations you'll want a larger time window so you can analyze trends over time. For more information, see **User activity data** and **Tenant inventory**.

Prepare to retire unused content

At this point, you've identified content that's a good candidate to retire. Next, you should perform some preparatory steps before archiving and deleting it.

- **Confirm scope of items to be deleted**: Before proceeding, ensure that you're clear on exactly what should be deleted. Usually, it's an entire workspace or specific individual items (such as a semantic model and the reports that use it).
- **Verify dependencies**: Analyze the data lineage and perform an impact analysis to confirm that you've accounted for all dependencies. For example, if a semantic model is going to be retired, all reports that depend on it will be retired as well.
- **Identify where content will be archived**: It's critical to have a backup, or an *archive*, of content before it's deleted. That way, you can recover it later should the need arise. The archive store should be a secure location with the minimum level of permissions to ensure that users don't accidentally find and use the archived content. There are several options for creating an archive. You might have source control in place (such as Git integration). Or you might choose to download the latest file from the Power BI service prior to deleting it. You might also have a semantic model backup in place. Generally, you should consider archiving content in the central repository where it's been stored during its lifecycle (such as a OneDrive or SharePoint document library, or a Git repository).
- **Confirm the recovery plan**: Once you've identified where the content will be archived (described in the previous point), consider what your process will be if you discover that you need to recover content or roll back. For example, you might ask your Fabric administrator to restore a deleted workspace, providing it's done during the retention period. Or you might choose to republish a specific item from the archive you created. Conduct a test of your recovery plan so that you're certain you can rely on the process.
- **Clarify whether approval is required**: Depending on the content and your processes, you might need to get approval prior to deleting content.
- **Create a change log**: If questions come up later, it's helpful to have documentation to refer to that includes what was deleted, when, why, and by whom.

- **Confirm who will handle the deletions**: Sometimes the person who decided to retire the content isn't the same person who will do the deletions. Verify that all required permissions are granted.
  - **Delete a workspace**: There are two ways to delete an entire workspace. Each way requires different permissions.
    - The workspace administrator role is appropriate when the user handling the deletion is permitted to access all content in the workspace.
    - The Fabric administrator role is well-suited to tenant administrators who don't require direct access to the content. In this case, they'll perform the deletion from within the admin portal.
  - **Delete one item**: A workspace role of contributor or above is required to delete an individual item from within a workspace.
- **Confirm when to perform the deletions**: Because the content isn't being actively used, the date and time to delete it shouldn't matter. However, to minimize risk, some organizations have an approved change window.
- **Communicate with the owner of the content**: Let the contacts or the owner know that the content is going to be archived and deleted. Usually, the consumers of the content don't need to be notified about inactive content. Although user communication might occasionally be appropriate, watch out for situations when users will be highly resistant to losing unused content because they aren't sure if they'll need it in the future.

> ⓘ **Note**
>
> The retention period for restoring a collaborative workspace might be different from a personal workspace. The *Define workspace retention period* tenant setting controls how long content can be recovered after it's been deleted. For more information, see **Workspace retention**.

Archive and delete unused content

At this point, you've identified content that can be retired, and taken the appropriate preparatory steps to remove and archive it. In this final step, you remove the content. This involves following through on the decisions and plans described in the previous section.

- **Archive the content to a secure location**: There are several ways you can back up the content, and allow you to recover it if necessary (described further in the previous section).

- **Delete the content**: The final step is to delete the entire workspace or individual items, as appropriate (described further in the previous section).

Checklist - When planning to retire and archive content, key decisions and actions include:

- ✓ **Plan for content clean-up activities**: Determine whether the scope of the analysis will be at the tenant level, domain level, or workspace level. Also consider who'll perform the usage analysis.
- ✓ **Decide on a time window**: Based on the characteristics of the specific content, determine the best time window for identifying candidates for retirement.
- ✓ **Select usage metrics**: Consider what usage metrics related to content views and viewers are most important for the content.
- ✓ **Create classifications**: Decide what classifications you'll use to consistently label content when you analyze usage patterns.
- ✓ **Analyze usage statistics**: Examine views, queries, viewers, refresh, and update activities. Sort, filter, and analyze the data to discover items that are candidates for retirement.
- ✓ **Communicate with the owner**: Contact the owner of the content to confirm your findings or to get additional insight into expected usage patterns. Be sure to coordinate archival and deletion activities with them.
- ✓ **Analyze data lineage and perform impact analysis**: Investigate dependencies. Confirm the scope of content to be deleted.
- ✓ **Determine how and where to archive content**: Decide what method will be used for backing up content.
- ✓ **Verify the workspace retention period**: View the *Define workspace retention period* tenant setting to understand what the retention period for deleted workspaces is set to in your organization.
- ✓ **Confirm and test the recovery plan**: Plan for how you'll recover content from the archive, should it become necessary. Test your recovery plan to ensure it's reliable.
- ✓ **Verify permissions**: Confirm that the necessary permissions are assigned to the user or administrator that'll do the deletions.
- ✓ **Archive the content**: Store a backup of the content in a secure location that's not accessible to consumers.
- ✓ **Delete the content**: Delete the entire workspace or individual items, as appropriate.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with your Power BI implementation, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Content distribution and sharing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to plan the distribution and sharing of content in Power BI and Microsoft Fabric. It's primarily targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators manage administration settings that affect how creators can distribute and share content in their organization. Fabric administrators can collaborate with other administrators to control the options available for creators to share their content.
- **Azure administrators**: The administrators who are responsible for overseeing identity management in the organization and creating external guest users to enable Microsoft Entra B2B.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams are responsible for defining and promoting a specific content distribution strategy.
- **Content owners and creators**: The teams and individuals that champion analytics in a team or department and that make and distribute Power BI or Fabric content. Content owners and creators should collaborate with Azure administrators to distribute or share content to external users.

After you create content like semantic models or reports, you'll usually want to share it with other users in your organization so that they can use it to inform their decisions and actions. There are many different approaches that you can take to share Power BI and Fabric content, and each approach has its benefits and considerations. It's important to take care because oversharing of content is a common governance and security challenge for organizations. To overcome this challenge and grant people access to the data and reports that they need, you should implement a content distribution strategy that fits your needs and scenarios.

> **Note**

The terms *sharing* and *distribution* are sometimes used interchangeably. In this article, we refer to content sharing and content distribution in the following ways:

- *Content sharing* involves giving others access to an item as part of the day-to-day activities of a content creator. For example, you can provide direct access to a report with someone else.
- *Content distribution* involves sharing content at scale for a wider audience of content consumers. For example, you can distribute reports to an app audience by using an app.

This article guides you to determine your approach to distribute content to both *content consumers* and other *content creators*:

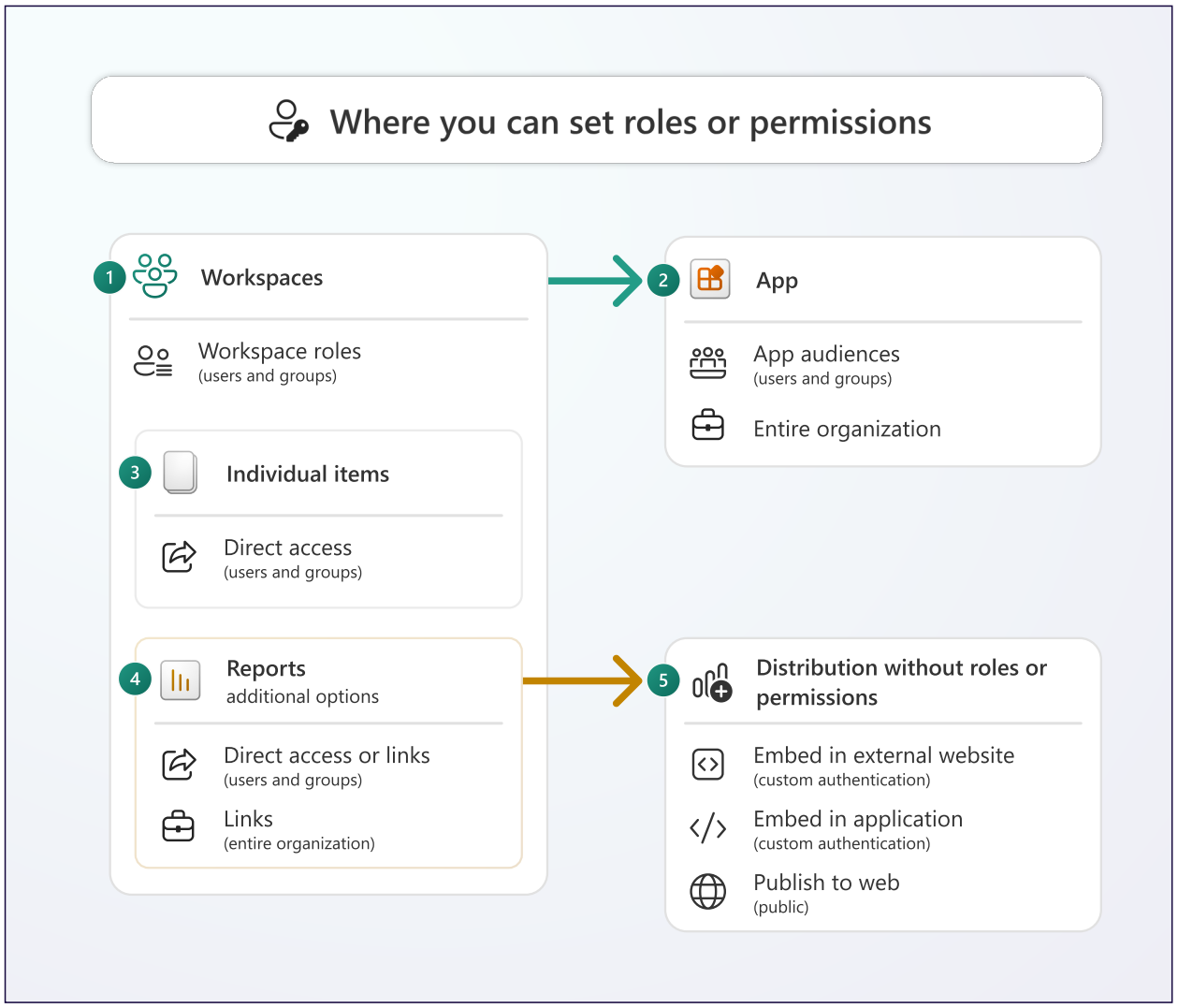
- **Content consumers**: Users who view content but don't create or share new items. These users typically have Read permission for reports, but more advanced content consumers might also request Build permission to perform ad-hoc analysis on data items in personal BI usage scenarios.
- **Content creators**: Enterprise or self-service users who create and distribute content to consumers. These creators often use existing items created by other users in the organization to make new content, for example, in managed or customizable managed self-service BI usage scenarios.

Set roles and manage permissions

You share content with consumers or creators by granting them access to that content. To grant access to users and groups, you set their roles and manage their permissions:

- *Roles* apply certain permissions to all content within a particular scope. For example, the Viewer workspace role applies read-only permission to all content within the workspace.
- *Permissions* define what an individual can do with that content. For example, applying Build permission to a semantic model allows a user to connect to and create reports based on the semantic model.

The following diagram shows where you can set roles and permissions in Power BI and Fabric.



The diagram depicts the following concepts and processes:

Expand table

Item	Description
1	You can share content in workspaces by assigning users and groups to workspace roles. Workspace roles can grant permissions to all items within the workspace.
2	You can share content from an app by adding people to app audiences, or by creating an app that shares content with the entire organization. • An app audience member will receive read-only permission to view content only in the app (and can't view individual items in the underlying workspace). • You can grant Build permissions to app audience members for the underlying semantic models of app reports and dashboards.
3	You can grant access to individual items by providing direct access and setting specific permissions. Some items (like dataflows) don't support sharing by direct access. For these items, you share them via workspace roles.
4	Reports have more options. For instance, you can provide direct access with a link that works for specific users and groups, or with a link for the entire organization. Links work

Item	Description
	for specific distribution methods, like embedding in SharePoint Online or securely embedding in an internal webpage. This article provides an overview of all possible methods later in this article.
5	It's possible to distribute Power BI reports without setting roles or managing permissions for users and groups. • You might choose to share Power BI reports by embedding them in a custom website or application. In this case, you manage custom authentication yourself. Embedding content in this way requires an F, P, EM, or A SKU. • You can also share reports publicly by using Publish to web, which is a Power BI feature that makes content available to anyone through the public internet.

Set roles for workspaces or apps

As depicted in the previous diagram, you can share a collection of content by assigning people to roles in workspaces or apps. A workspace is a collaborative location to publish and organize content. An app is a selection of reporting content that you organize and distribute to audiences.

- Workspace roles control whether a user can manage workspaces, content, and connected apps (Admin, Member, or Contributor), or only view content in that workspace (Viewer).
- App audiences control which content is visible in an app to a user or group. You can also publish an app to your entire organization.

Fabric administrators have access to all workspaces and workspace content. Also, Fabric capacities can contain one or more domains, which logically group workspaces together. Giving someone access to a domain *does not* grant them access to its workspaces, items, or apps.

Other supporting roles for content distribution or sharing

There are other roles that don't directly affect how you share content, but which you can use to support content distribution and sharing in certain circumstances:

- Data source connection roles control whether a user can use a data source connection of a data gateway. If a DirectQuery semantic model uses DirectQuery with single sign-on (SSO), then you must add a user or group as a *User* role so that they can use the semantic model or view reports. You might also decide to assign creators to a data gateway role so that they can add and manage their own data source connections.

- Data security roles control what data people can access in the semantic model. If a semantic model has a data security role, then you must add a user or group to that role before you share the model or any connected reports and dashboards.
- OneLake data access roles control what data people can access in a lakehouse. If you want to share only a subset of data in a lakehouse, you must add a user or group to the OneLake data access role after you grant the appropriate lakehouse Read permission.
- Deployment pipeline roles control who can deploy content between workspaces. Although deployment pipelines aren't directly related to content distribution and sharing, how you decide to deploy content and who deploys it might affect your strategy to share content with creators.

We recommend that you use security groups to manage permissions and role membership instead of using individual user accounts, where feasible. That way, it's easier to manage, especially when you have many users. You can also use the same security groups to manage other access control, like data security role membership.

For more information and use cases for using security groups in Fabric and Power BI, see Strategy for using security groups.

Manage item permissions

When you share content via workspace roles or app audiences, you apply certain *permissions* to a set of content. These permissions define what an individual can do with the content. For example, if you assign someone to the Viewer workspace role, they get Read permission on all items in that workspace. Then they can only view that content but not modify it or connect to data items to create their own content.

You can also share content by managing permissions to individual content items. For example, if you manage permissions of a semantic model, you can grant someone Build permission to connect to and create their own content based on that semantic model. When you share a content item via direct access, you're granting certain permissions to that item for a user or group.

You can assign users to the following permissions for Power BI or Fabric items:

- **Read-only (view) permission**: Let someone see the content without modifying it. Some items, like lakehouses, allow more granular control of how users can view the content (like ReadData or ReadAll permissions). Other items, like semantic models, allow users to see information about the item but not view its contents (like the data).

- **Reshare permission**: Let someone share the content with others. If someone has both Reshare and Build permissions, they can also grant Build permission to someone else.
- Build permission: Let someone connect to the data item (like a semantic model) to create their own content (like reports). Consumers might also need Build permission to view reports based on composite semantic models. If you grant Build permission to a report or app audience, it automatically applies to the underlying semantic models.
- **Write (modify) permission**: Let someone modify an item and save their changes.

> ⚠ **Warning**
>
> Don't confuse Build permission with Write permission. For example, Build permission *does* respect any data security (like row-level security) that's enforced for an item, whereas Write permission *does not* (because a user can set and change those security rules).

Content distribution and sharing overlaps with many other important areas. The following subjects are covered in the Power BI implementation planning series. Refer to the articles in this series as you plan how you'll share and distribute content:

- Workspaces: How you plan to set up and use workspaces affects whether you'll share content from a workspace or not. Workspace-level planning is important for your content distribution strategy, such as how you'll set up and manage workspace access.
- Data gateways: When you distribute content to other users and that content relies on on-premises or VNet data gateways, you might also need to manage gateway access. Ensure that you manage access to both the gateway and its data source connections.
- Content lifecycle management: How you decide you'll share content can affect how you'll manage the lifecycle of that content. You typically share content as part of your post-deployment activities.
- Information protection and data loss prevention: Oversharing content or sharing content inappropriately is a governance and security risk for organizations. One way to mitigate this risk is to practice data loss prevention for Power BI.
- Security: Sharing is an important topic in the context of securing your Power BI and Fabric content. Ensure that you conduct report consumer planning for sharing content to consumers, and content creator planning for sharing content to creators.
- Auditing and monitoring: You should regularly audit and monitor sharing activities and prevalence by conducting tenant-level auditing.

For an overview of the relevant administration settings a Fabric administrator can use to control how creators share content, see Export and sharing tenant settings.

The remainder of this article guides you to make decisions about which roles and permissions you should give to content consumers and creators to share content with them.

Plan to distribute content to consumers

Most people who use content in your organization will only need to view it (Read permission) and not change it (Write permission), or create their own content based on it (Build permission). Content consumers are typically business users who need to access reporting items (in which case they're often called report consumers). Some users might also need to request Build permission for data items to enable their own data discovery in Power BI Desktop, Excel, or other client tools.

Typically, consumers who request Build permission perform personal BI. However, over time, you can expect that some of these users will evolve from being strictly content consumers to becoming content creators who want to share their content with others.

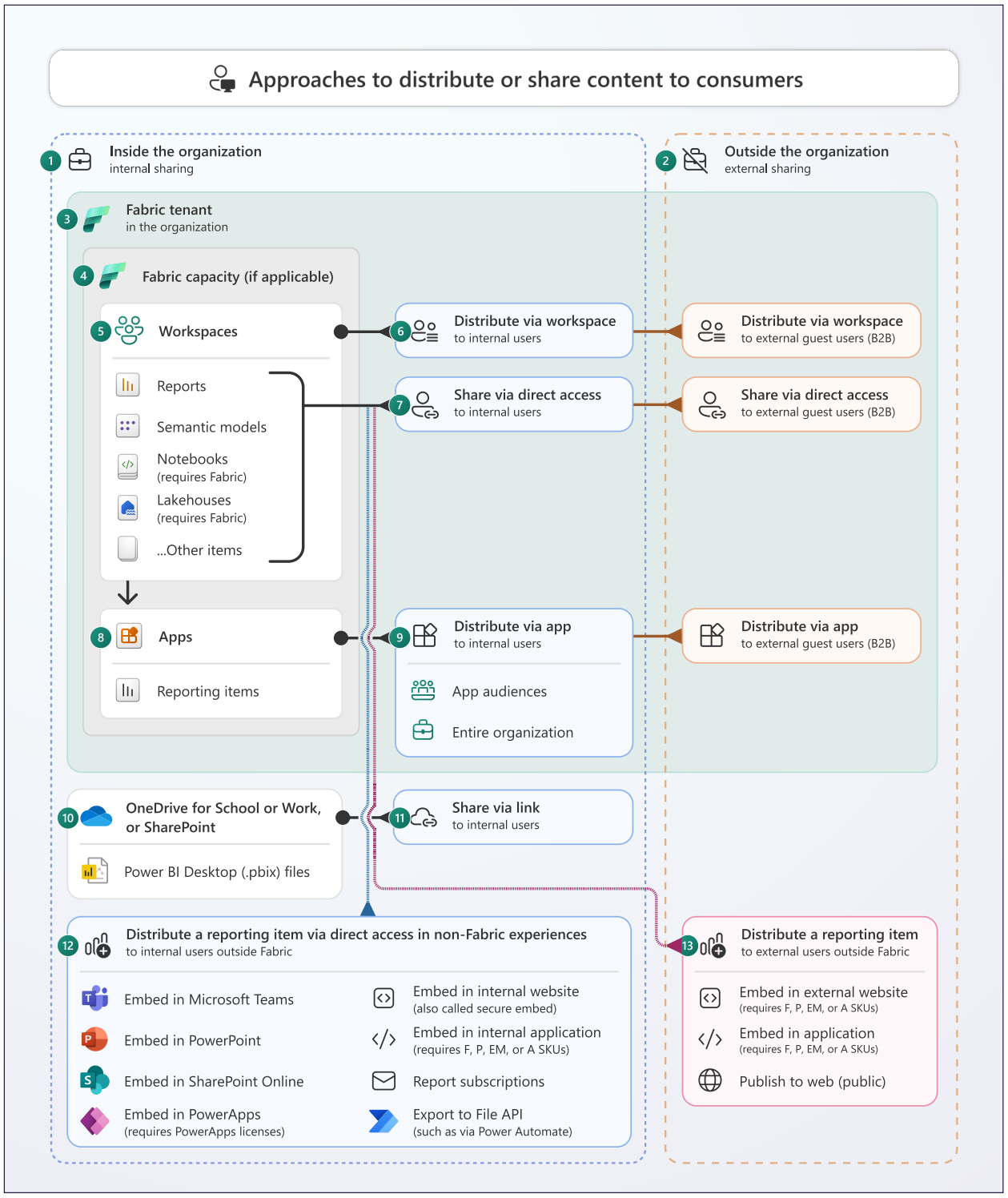
Anticipate early on how you'll support and educate these users not only on how to create content, but also on how they can appropriately share it.

Content consumers can be either internal or external to the organization, depending on whether they have a Microsoft Entra identity in the same organization as where you publish the content:

- **Internal users**: Users who belong to the same organization as the content that they consume.
- **External users**: Users who belong to a different organization as the content that they consume. These users can belong to an affiliate or subsidiary organization, despite sharing the same parent organization. They can also be temporary collaborators (like consultants) or customers.

The following diagram presents an overview of the different approaches that you can take to distribute Power BI or Fabric content to consumers inside or outside your organization.

This diagram depicts an overview of many possible approaches. The rest of this article helps you to understand these options and choose the ones that best fit your specific needs.



The diagram depicts the following concepts and processes:

Expand table

Item	Description
1	Content creators create and deploy content in their organization. They can distribute content to other internal users who use Fabric (such as the Fabric portal or the Power BI mobile app). Creators can also share reporting items by embedding them in other internal, non-Fabric services.
2	Creators can also share items outside the organization. They can distribute content to external guest users who have been added to the organization with Microsoft Entra B2B.

Item	Description
	Creators can also share reporting items to external users.
3	Content is deployed in a Fabric tenant, where consumers can use or view it.
4	Fabric capacity is required to create and share Fabric items. If there isn't a Fabric capacity, creators can still create, deploy, and share Power BI items (like dataflows, semantic models, and reports).
5	Content is deployed to a workspace, which can contain different items that might be organized into folders. Workspaces can belong to deployment pipelines and domains (not depicted in this diagram).
6	Creators can share content to internal users or external guest users who are content consumers by assigning them to the Viewer workspace role. This role grants read-only access to all content in the workspace.
7	Creators can share individual items to internal users or groups by providing direct access or sharing a link. People who use the link in the organization who don't have direct access can request it. Creators who share content via links can also modify the URL to filter the report by using query string parameters.
8	Selected reporting items in a workspace can be packaged into an app.
9	Creators can distribute content via apps by creating and adding internal or external guest users who are content consumers to one or more app audiences. Creators can choose to organize app content into audiences, with each audience having access to different subsets of content.
10	Creators might store content files, like Power BI Desktop (.pbix) files, in SharePoint or OneDrive for Work or School.
11	Creators can share a link to .pbix files from SharePoint or OneDrive for Work or School. When an internal user has a Power BI Pro license, they can preview the reports from these files without opening them (even when the files aren't published to a workspace). Recipients of the link who don't have access or a Power BI Pro license can't preview the files.
12	Creators can share reporting items via direct access by using non-Fabric experiences. They can: • Embed content in Microsoft Teams (as a tab in a chat or team channel). • Embed content in a PowerPoint presentation. • Embed content in SharePoint Online. • Embed content in a PowerApps canvas app (which requires PowerApps licenses, but not an F, P, A, or EM SKU). • Embed content in an internal website (also known as secure embed). • Embed content in an internal application (also known as *Embed for your organization*, which requires an F, P, A, or EM SKU). • Report subscriptions and paginated report subscriptions, which provide scheduled

Item	Description
	delivery of static reports via email. You can also use regular subscriptions and dynamic per recipient subscriptions for both reports and paginated reports. • Export to file API (such as by using Power Automate to deliver exported reports via email).
13	Creators also have various options to share reporting items to external users outside Fabric. They can: • Embed content securely in an external website (requires an F, P, A, or EM SKU). • Embed content securely in an external application (also known as *Embed for your customers*). • Publish to web for reports, which makes content available to anyone through the public internet. Publish to web is recommended only for content that's suitable for a public audience. Publish to web isn't suitable for sensitive or confidential reports. For more information, see Publish to web from Power BI.

There are many valid approaches to distributing content. You might even use multiple approaches to distribute the same content, depending on your needs and use cases. What's important is that you find an approach that's effective and sustainable.

The following sections describe the steps you take to determine your approach to sharing content with consumers.

Step 1: Identify who consumers are and how they get access

The first step when distributing content to consumers is to identify who will need access to what content. Depending on the number and type of consumers, you'll make different choices about how you distribute content.

Investigate and document answers to the following questions:

- How many consumers are there?
- What content do these consumers need access to?
- Do consumers need access to data gateway data source connections to view content that uses DirectQuery mode to query on-premises or private network data sources?

Ensure that you define who the consumers of the content are and how they'll use it. One group of consumers might have different needs and preferences than another group. The other group might benefit from a different content distribution strategy.

You can choose to either proactively distribute content, set up processes for users to request access themselves, or provision organization-wide access for all users.

The following sections help you to determine which is the better option for your circumstances.

Option 1: You distribute content to consumers

You can proactively distribute content to consumers, and then notify them about the content and how they can access it. With this approach, you gather a list of all the consumers and assign them to a distribution group, mail-enabled group, or security group to facilitate sharing the content. You can then announce the content availability via a centralized portal or another method of communication.

This approach is beneficial when:

- You need to distribute content to a defined, large audience, like an entire department or region.
- You want tighter control of content access and releases, such as when you want to conduct user training for an app or report before providing access.
- You haven't yet set up processes for consumers to discover content.
- You use a centralized model of content ownership and delivery.

For a step-by-step guide to create and add members to a Microsoft Entra security group, see this tutorial.

Option 2: Consumers request access to content

You can also take a discovery-driven approach, where consumers search for content that's relevant for them. With this approach, consumers browse a curated centralized portal for the content that's available to them. When they select an item that they don't have access to, they can request access from the content owner, who must approve it before the content consumer can view that item.

This approach is beneficial when:

- You curate a centralized list of content for users to explore and access, such as one maintained on a Microsoft Teams channel or SharePoint site.
- You have processes in place for content owners to quickly and reliably respond to access requests.
- There are many possible items that users might want to view, depending on their current tasks and objectives.
- You use a decentralized model of content ownership and content delivery scope.

You can use endorsement to promote or certify quality content when you distribute it this way. Endorsement helps make quality and trustworthy content more easily discoverable by consumers.

This approach works best to distribute content to creators (rather than consumers) in managed or customizable managed self-service BI usage scenarios. In these scenarios, you provide creators read-only access to semantic models that they can discover and request Build permissions to. These creators can then use these semantic models to make their own content.

Option 3: The entire organization can access the content

Some content might be relevant to the entire organization, like reports about organization-wide initiatives. With this approach, you share content either via apps or direct access. The entire organization then can access content in an app or by a link.

This approach is beneficial when:

- Content and the underlying data are suitable for the entire organization.
- You have processes in place for content consumers to find and understand this content, such as training, office hours, or frequently asked questions (FAQ) pages.

> ⚠ **Warning**
>
> You should limit which content is available to the entire organization. Some content owners might make apps and links available to the entire organization to circumvent the administrative overhead of managing content access. This approach can quickly lead to content oversharing and users who can't find the content that they need.
>
> To avoid this situation, regularly **audit your tenant** to identify and review the available organization-wide apps and links.

Expect your approach to distributing content to evolve over time as you scale and grow. For example, you might start by distributing content to self-service users proactively. As you progress to higher maturity levels in key areas like content ownership, content delivery scope, and governance, you might scale to an approach where users discover and request access to content themselves.

Step 2: Identify where consumers will view the content

After you define who the consumers are and how they should get access to content, you then identify where these individuals can view the content.

Consumers can view content in four different ways, specifically by using:

- Links to the individual content items.
- Workspaces.
- Apps.
- Non-Fabric experiences, like where content is embedded in other services, custom apps, or websites.

The following sections help you to determine which is the better option for your circumstances.

Option 1: View individual content items in Fabric

You can give consumers access to individual content items, which they can access via a link. When you share content this way (called *direct access*), you can also give consumers the ability to reshare the content with others or build content based on the underlying data items. If someone who receives the link doesn't have access to the item, they're prompted to request access from the original content owner.

This approach is beneficial when:

- You share content to a limited audience, rather than distributing it to a wide audience.
- Consumers need access to only a few items.
- You want to later embed that content in another internal service or website.
- You want to provide users with a collection of links to different content.

When you share individual content items, consider the following points:

- Use endorsements to help consumers find and request access to quality content.
- Audit and monitor published items to identify overshared or underutilized items to review.
- Create a process to regularly review pending access requests. Ensure that feedback is provided to users when their access request is denied. It should explain the reason and tell them where they might find alternative information.

> ⚠ **Warning**
>
> Managing content access can quickly become a burden for some content owners. As you improve your maturity level in content delivery scope, consider granting

Reshare permission and training users how to share content among each other, where appropriate.

Consumers with Reshare permission can make links to shared views, which can include state settings, like the current filter and slicer selections. Shared views can be helpful to facilitate discussions about interactive reports. They're also a good alternative to sharing static screenshots because recipients can interact with the content. For example, shared views are an effective way for users to report issues or discrepancies in reports because creators can easily see the user's active slicer selections.

Content owners can manage shared views in the **Manage permissions** pane.

Option 2: View content in a workspace

You can distribute content via workspaces. With this approach, you grant read-only access to consumers to workspace content by adding them to the Viewer role. Viewers can't modify the content or create new content from the items in the workspace. Instead, you distribute a link to the workspace that lets them browse and open the individual content items.

For more information about workspace roles, see Roles in workspaces.

This approach is beneficial when:

- Consumers prefer the workspace experience over the app experience.
- You separate workspaces by item type and have only reporting items in the workspace.
- You want to give users the ability to view data items in a workspace and request Build permission.
- You want the content reviewed before it's published to an app.

When you plan to distribute content to workspace viewers, consider the following points:

- Ensure that you use clear workspace naming conventions so that users can easily find workspaces that contain the content they need.
- Organize workspace items into folders. Workspace organization makes it easier for consumers to find what they're looking for.
- Place content not meant for direct consumption in other workspaces. Separating workspaces by item type ensures that consumers won't be confused or overwhelmed by items that aren't for them to use.

- Regularly audit workspace role membership. Monitoring role membership helps to avoid situations where a consumer is mistakenly granted elevated permissions.

Option 3: View content in an app

You can distribute content via apps, which bundle related content in a straightforward user interface that streamlines the experience for content consumers. Unlike viewing content in workspaces, apps don't confuse or overwhelm consumers by showing nonreporting items (like semantic models).

When you distribute content via apps, there's no risk of accidentally giving consumers Write permission. Apps require you to share content by adding consumers to an app audience. It's also possible to share an app with the entire organization.

You can't publish an app to provide access to content from multiple workspaces. If you want to share content that's published to multiple workspaces with the same consumer, you must either publish multiple apps or grant them access to the individual items or workspaces.

This approach is beneficial when:

- You want to distribute multiple workspace items (but not all items) to consumers.
- You need to distribute content from a single workspace to many consumers.
- You want to give different consumers access to different subsets of app content.

When you share content in an app, consider the following points:

- Organize content into folders so that it's easier for consumers to find what they're looking for.
- Add links to the app for users to request support, report issues, or provide feedback.
- Include a useful app description and contact person so that users know what the app is about.
- When reports use shared semantic models, which are located in other workspaces, ensure that consumers have Read permission for those semantic models so they can view reports in the app.

We recommend that you distribute content to consumers by using apps. Apps provide a more polished experience than providing access with workspace roles or by sharing a collection of links to individual items.

Option 4: View content from other services

You have various options to distribute content for consumers in your organization outside Fabric. In this situation, content consumers view the content from other applications and services.

This approach is beneficial when:

- Consumers intend to use the content alongside, or together with, other applications.
- You require other services or applications to deliver the best experience for consumers.
- You want to provide convenient access to content spread across multiple workspaces (such as from a SharePoint Online or internal website portal).

Consumers who view content from other services still require direct access to the Power BI items. Ensure that you first share the individual items with these users or distribute the items via direct access to a security group that the users belong to.

Each approach has specific use cases that they can support:

- Embed in Microsoft Teams: This option can be beneficial when you use Power BI reports to help coordinate project or other collaborative work.
- Embed in PowerPoint: This option can be beneficial when you want to integrate Power BI reports in your presentations or display multiple reports continuously on a screen. Examples might be reports that show incidents or open orders in a manufacturing plant.
- Embed in SharePoint Online: This option can be beneficial when you want to centralize reports in a SharePoint site to enable collaborative work, or when SharePoint is the preferred way to distribute and manage access to different kinds of content in the organization.
- Embed in Power Apps canvas apps: This option can be beneficial when you want Power BI content to inform user decisions and actions in an app built with Power Apps. For example, you might have an app for registering seats in an open office, and a Power BI report that shows data about seat availability and office occupancy. However, Power Apps requires licenses for both creators and consumers.
- Embed in an internal website or external application: This option can be beneficial when you want to integrate Power BI reports with other custom developed content. However, this approach requires more development effort. When done for an internal website, you can use the For your organization embedding approach. When done for an external application, you can use the For your customers embedding approach. This approach requires either Fabric capacity (F SKU) or Power BI Embedded (EM SKU).

- Report subscriptions: This option lets you subscribe consumers to reports and paginated reports, which can deliver static reports by email or to a SharePoint site on a schedule. For reports, subscriptions have the benefit of notifying consumers when the data in the underlying semantic model is refreshed. This approach is best combined with other methods, like sharing content from an app. That way, consumers know only to check the app when the content inside is updated.
- Export to file API: This option enables limited scenarios when you want to distribute static versions of reports as a PDF document (.pdf) or PowerPoint (.pptx) file.

> ⚠ **Warning**
>
> Avoid distributing content as exported files unless it's absolutely necessary. For example, it's common for users to request exported reports via email or SharePoint. This approach isn't a content distribution strategy that scales. Also, files can create governance and security risks. Users also can't benefit from interactive functionality, like cross-filtering or drillthrough in reports.
>
> Often, a preference for distributed files is an indication of a low maturity level and a need for **user mentoring and enablement** and **change management**.
>
> A good example of when you might distribute files is when reports must be kept for auditing purposes.

If you choose to distribute content via report subscriptions or exports, consider using sensitivity labels with Microsoft Purview Data Loss Prevention policy for Power BI. Appropriate use of sensitivity labels can prevent exported files with sensitive data from exfiltrating your organization.

Step 3: Identify whether consumers are outside the organization

In some cases, you might need to distribute content to external consumers. In this case, you have several options to distribute content outside the organization.

Option 1: Create duplicate identities for external consumers

In this approach, you create duplicate identities for external consumers that become internal consumers. An Azure administrator creates a new account for the external consumer in the tenant that they must use to access that content. Then, a Microsoft 365 administrator allocates the appropriate per-user licenses to the new identity for them to

access content. Although this option is the simplest, it involves overhead and cost to create and manage new accounts.

This approach is beneficial when:

- Limitations of Microsoft Entra B2B prevent you from doing what you need to do. Examples of limitations of Microsoft Entra B2B for consumers include:
  - External guest users might not be able to open files exported from reports that have a sensitivity label with protection settings.
  - External guest users can't use Power BI Desktop to connect to semantic models or dataflows that are in the Power BI service.
  - External guest users can't use Analyze in Excel.
- You don't have the development skills or resources to embed content in a custom application.
- There aren't many long-term, external consumers.

Option 2: Create a Microsoft Entra B2B guest account for external consumers

Microsoft Entra B2B is a good option to distribute content to external consumers. With this approach, an Azure administrator adds the external consumer as an external guest user in Microsoft Entra ID. If the external user is required to access content, the Azure administrator or a Microsoft 365 administrator can provision a per-user license for the identity.

External users can also use bring-your-own-license (BYOL) to bring licenses from their organizations to access content that requires a per-user license. When the guest user has an appropriate license, creators can then share content with them.

If you need to distribute content to consumers outside your organization, we strongly recommend that you read Distribute Power BI content to external guest users using Microsoft Entra B2B.

External users are made external guest users by an Azure administrator. If the guest user requires a license, a Microsoft 365 administrator can then assign a license to the guest user. These guest users can then access Fabric, including workspaces, apps, and items. Access to Fabric by external guest users can be controlled in the tenant settings by a Fabric administrator.

Consumers don't require a Power BI Pro license to view content in a Fabric capacity. However, creators do require a Power BI Pro license to create and share Power BI content. Consumers require a Power BI Pro license to view content in a workspace that uses Pro or Premium per user (PPU) license mode.

Microsoft Entra B2B requires that you enable specific settings in the Fabric admin portal. For an overview of the relevant settings a Fabric administrator can use to control how creators share content, see Export and sharing tenant settings.

Here are some scenarios where you can use Microsoft Entra B2B to deliver content to external consumers:

- Direct collaboration between organizations
- Parent and subsidiaries or affiliates
- Shared tenant among partner organizations
- Distribution to many external partners

This approach is beneficial when:

- There's a manageable number of external consumers.
- You have Fabric capacity and want external consumers to benefit from free viewership, or the external consumers have their own Power BI Pro licenses to use with BYOL.
- You don't have the development skills or resources to embed content in a custom application.

Microsoft Entra B2B is subject to limitations that you should keep in mind. Plan carefully and conduct a test with one or more external users before you commit to your content distribution approach.

Option 3: Embed content in a custom website or application

In this approach, you use either an F, P, A, or EM SKU to embed content in a custom external website or application to (embed for your customers). This approach requires skills, time, and resources to develop a custom application and manage authentication. It does provide flexibility and control over the access process.

This approach is beneficial when:

- You have the expertise, time, and resources to develop and maintain a custom application.
- You have an existing custom application in which you want to embed Power BI content.
- You distribute content to many external parties or customers.
- You want to monetize access to proprietary Power BI content to earn revenue.

Option 4: Publish content for a public audience by using Publish to web

Publish to web provides access to Power BI reports to anyone on the internet. It doesn't require authentication and access isn't logged for auditing purposes. Because report consumers don't need to belong to the organization or have a Power BI license, this technique is well suited to data journalism. In this process, reports are embedded in blog posts, websites, emails, or social media.

This option is beneficial when:

- Content doesn't contain sensitive information and is intended for a public audience.
- You want to embed or link to the content in a public website, for example, by using an HTML iframe element.

The Publish to web feature is a powerful technique that shares Power BI reports with the world. It has the potential to expose private or sensitive data, whether accidentally or intentionally. For this reason, this feature is disabled by default. A Power BI administrator must enable the feature and should restrict its use to specific trusted users or security groups.

Publish to web is subject to many limitations. Test that your report design works when you use Publish to web before you commit to this approach.

Step 4: Identify consumer licensing considerations

By now, you're ready to decide how to distribute content to consumers. You should now ensure that consumers have appropriate licenses to access and view that content. Depending on how you decide to distribute content, you might need per-user, per-capacity, or other licenses.

- **Per-user licenses**: Content consumers might require Microsoft Fabric Free, Power BI Pro, or Premium Per User (PPU) licenses to view content, depending on other factors (described later in this section).
- **Per-capacity licenses**: Your organization requires either a Fabric capacity (F SKU) or Power BI Embedded (EM SKU) to embed content in a custom application.
- **Other licenses**: If you distribute content by using other services or applications, content consumers might require other licenses. For example, both content creators and consumers require a Power Apps license to distribute and view content by using this service.

Before you decide on your approach to distribute content, ensure that you estimate the monthly (and annual) licensing costs (if new licenses are required). This proactive approach can help to avoid surprises that could delay content delivery or lead to unexpected issues and costs later on.

For guidance about per-user and per-capacity licenses, see subscriptions, licenses, and trials.

The following sections outline specific per-user licenses you might need for consumers.

Scenario 1: Consumers require no license

Although content creators always require a license to publish and share content, there are certain circumstances where consumers don't require a license to view published content.

- Content is distributed publicly via Publish to web.
- Consumers access embedded content via an external website or application (which uses the For your customers embedding scenario).

Scenario 2: Consumers require Microsoft Fabric Free licenses

When you distribute or share content that's published to a Fabric capacity, consumers still require a license to access that content. They can do so at no cost with a Microsoft Fabric Free license, and they don't need any other per-user license, like Power BI Pro. An organization has unlimited Microsoft Fabric Free licenses that they can use to provide access to Fabric and Power BI content.

Scenario 3: Consumers require Power BI Pro licenses

Content creators always require a Power BI Pro license to share Power BI content, even if that content is published to a workspace on Fabric capacity. However, content creators don't require a Power BI Pro license to create or share Fabric content.

Content consumers require a Power BI Pro license to:

- Consume content that's in a workspace that doesn't use Fabric capacity license mode.
- Share Power BI content with other consumers.
- Preview shared .pbix files that are stored in OneDrive for Work or School, or SharePoint.
- Use a Microsoft Entra B2B guest account that can't use BYOL.

Scenario 4: Consumers require Power BI Pro or PPU licenses

If content is published to a workspace that uses PPU license mode, then both content creators and content consumers require a PPU license to share and view that content.

Granting a user a Power BI Pro or PPU license doesn't automatically grant them Build or Write permissions to Power BI items that are already deployed to a workspace. Licenses affect what a user can do but have no effect on item permissions or the various roles within Fabric.

Step 5: Identify whether consumers require Build permission

Some consumers might have more advanced needs that aren't met with the available read-only content. These consumers could benefit from having Build permission on the underlying data items (like semantic models) so that they can create new content (like reports).

Eventually, consumers might want to share their content with others, at which point they also become content creators.

If you grant Build permission, ensure that you train content consumers about how to create their own content by using Power BI or Excel. If you don't, these users might use their permissions to export data to Excel, which invalidates the benefit of them having Build permission in the first place.

Option 1: No Build permission

The simplest scenario is when users only need to view content, and they don't want to produce their own content.

This approach is beneficial when:

- You currently lack the resources or processes to support consumers so that they can successfully create their own content.
- You haven't yet trained users how to connect to, and create content from, data items.
- Consumers express that they only want to view content and not create it.
- There's no business need for consumers to create their own content.

In these scenarios, you should seek to provide alternative approaches to granting Build permission when consumers still want to analyze and explore data.

Consider the following Power BI features that provide flexibility to consumers:

- **Field parameters**: You can create field parameters in semantic models that support the dynamic selection of dimensions and measures in reports. This option provides

users with the ability to select a field from a slicer, and that field will be then used in visuals.

- **Decomposition tree visual**: You can use the decomposition tree visual in reports to let consumers explore data across multiple dimensions that they choose.
- **Personalize visuals**: You can enable personalization of visuals in a report, which lets consumers change visuals to suit themselves. They can modify any visual, including the visual type, format settings, and which fields the visual uses. They can also customize the report page to any field you preselect from the semantic model, enabling them to perform ad-hoc analysis.

If consumers don't require Build permission now, they might need it in the future. Business objectives and data needs can evolve quickly. Teaching users to create their own content and perform their own analysis is a good way to scale your implementation by enabling users and advance your data culture maturity.

Option 2: App audience Build permission

You can grant Build permission to members of an app audience, which applies to all the underlying semantic models for the reports and dashboards that are included in the app.

This approach is beneficial when:

- You distribute content by using Power BI apps.
- Consumers require Build permission only for semantic models.
- You need to give multiple users Build permission to multiple items in the same workspace.

Don't grant Build permission by adding consumers to the Admin, Member, or Contributor workspace roles. These roles grant *Write* permission, and not *Build* permission. When users have Write permission, row-level security isn't enforced, and it allows them to view and modify the security rules.

Option 3: Item Build permission

You can grant Build permission to individual data items in a workspace.

This approach is beneficial when:

- You provide access to workspace content by adding consumers to the Viewer role.
- You need to grant Build permission only on specific items.
- You plan to grant Build permission to a small subset of users.

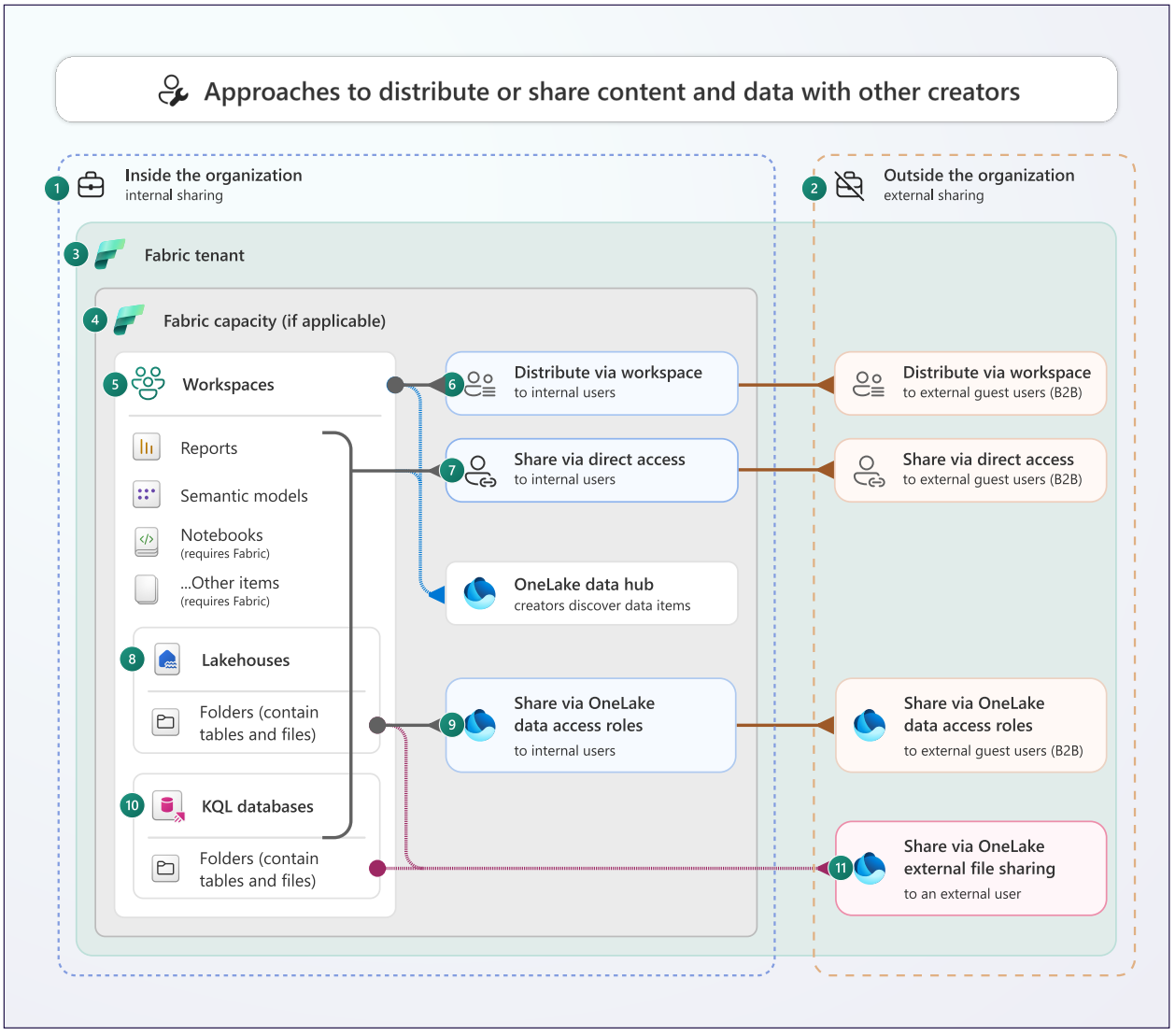
Checklist - When planning content distribution to consumers, key decisions and actions include:

- ✓ **Define who you'll distribute the content to**: If you haven't done so already, determine how many users need access to the content. Define any relevant user segments and whether these segments have different needs or ways of using the content that could affect the distribution method.
- ✓ **Plan your data security model**: Determine whether your content has data security applied and how you plan to manage security role membership.
- ✓ **Plan your permission model**: Determine who should get access to which content and how this access affects your distribution methods.
- ✓ **Create Microsoft Entra security groups and add users to groups**: Using security groups to manage role membership, access, and permissions is more efficient and scalable than managing individual accounts.
- ✓ **Add security groups to data security roles**: If you use row-level security or object-level security, ensure that you assign security groups to the appropriate roles.
- ✓ **Share data gateways**: If necessary, grant data gateway access by adding consumers to data source connections by using the User (or User with sharing) role.
- ✓ **Decide how you'll handle external users**: Identify whether you need to share content to any consumers outside the organization. If so, decide the approach you'll take to share content with these users.
- ✓ **Add external guest users**: If you plan to share content by using Microsoft Entra B2B, request that the relevant administrators add the guest users.
- ✓ **Allocate per-user licenses**: Ensure that consumers have the appropriate licenses to access content. If you plan to embed content in other services, ensure that you check whether those services have other licensing requirements.
- ✓ **Decide on the approach to share content**: Decide whether you'll share content via apps, workspaces, or direct access to individual items. If you use direct access, decide how users should access the items (such as via a centralized portal or via content embedded in another service, like Teams).
- ✓ **Decide how you'll grant Build permission**: Decide whether you'll grant Build permission at all, and if so, whether you grant it via app audiences or individual content items.
- ✓ **Grant Build permission**: If necessary, grant Build permission to users who will need it to perform their own analysis on the underlying data items.

Plan to share content with other creators

In enterprise and self-service scenarios, you'll need to share content with other creators. Sharing content with other creators facilitates collaboration among developers who work on the same content. It also facilitates data discovery and data democratization among self-service users who create their own content.

The following diagram presents an overview of the different approaches that you can take to share content with other creators.



The diagram depicts the following concepts and processes:

Expand table

Item	Description
1	Content creators create and deploy content in their organization. They can share content with other internal users who use Fabric (such as the Fabric portal or the Power BI mobile app).

Item	Description
2	Creators can also share items outside the organization. They can distribute content to external guest users who have been added to the organization with Microsoft Entra B2B. In some scenarios, creators can also share their content files to external users.
3	Content is deployed in a Fabric tenant, where creators can use or view it.
4	Fabric capacity is required to create and share Fabric items.
5	Content is deployed to a workspace.
6	Creators can distribute content with other internal users or external guest users who are content creators by assigning them to the *Admin*, *Member*, or *Contributor* workspace roles. We recommend that you use the principle of least privilege when you share content by workspace role.
7	Creators can also share content with other internal users or external guest users who are content creators by sharing individual items via links. When creators have access to data items via direct access or workspace roles, they can discover this content by using the OneLake catalog.
8	In a workspace on Fabric capacity, creators can make and use lakehouses, which comprise tables and files in folders.
9	Creators can share data in a lakehouse folder to other internal users or external guest users who are creators by using OneLake data access roles, which is also known as role-based access control (RBAC).
10	In a workspace on Fabric capacity, creators can make and use KQL databases, which comprise tables and files in folders.
11	Creators can share data from a lakehouse or KQL database folders with external users outside the organization by using external data sharing.

Step 1: Identify who creators are and how they get access

The first step when sharing content with creators is to identify who they are and how they should get access to the items or data that they need to make and share their own content.

You're then ready to complete the following tasks:

- **Share gateways**: If necessary, set up data gateway access by adding consumers to data source connections in the User (or User with sharing) role.
- **Create workspaces**: Create and share access to a workspace where creators should publish their content, if they don't have one already. Creators might need access to

more than one workspace if they plan to use separate workspaces to develop, test, and share content with consumers. Alternatively, you might share access to an existing workspace that contains other content that creators need access to (described later in this article).

- **Allocate licenses**: Creators need different licenses depending on the content that they produce.
  - Power BI Pro licenses are always required to create and share Power BI content.
  - PPU licenses are required to create and share content when the workspace uses PPU license mode.
  - No license is required to create and share Fabric items when the workspace is on Fabric capacity.

Option 1: You provide access to creators

You can proactively share content with creators, like the approach described for content consumers.

This approach is beneficial when:

- Creators need access to only a few data items.
- You want more control over what content creators can find and use.

Option 2: Creators find items and request access from the OneLake catalog

You should encourage users to use the OneLake catalog to discover content that they can use as a data source. There, and when necessary, content creators can request access to these data items. When they're granted Build permission, they can then create new content based on these items.

This approach is beneficial when:

- You use endorsements to promote and certify quality content.
- You mark content as discoverable so that users can find it from the OneLake catalog.
- You've created a process to sustainably manage access requests.
- Users have undergone training on how to find and use data items in Fabric.

To see a data item in the OneLake catalog, creators must either have access to that item or it must be marked as discoverable. For example, you can give creators Read permission to semantic models or mark those semantic models as discoverable, which allows users to find them and request Build permission to use them.

Step 2: Decide on creator permissions

Depending on how creators will use the content, they might need different permissions.

Option 1: Write permission

You can grant content creators Write permission to edit and collaborate on an item. When you grant content creators Write permission, ensure that you also grant access to content files, if necessary.

Grant creators Write permissions when:

- Creators need to change or manage the item.
- You'll collaborate with the creators on the item.

If content creators will collaborate, you should plan how they can best do so. For example, consider how you can use version control to track and manage changes.

Option 2: Build permission

You can grant content creators Build permission to connect to and use semantic models to create new content. With Build permission, content creators won't be able to modify the original item.

Grant creators Build permissions when creators will create new content from existing semantic models.

Some data item types, like lakehouses or data warehouses, don't support Build permission. Instead, if creators want to use these data items to create their own content, you should grant them Read permission.

Option 3: Read permission

You can grant creators Read permission to certain Fabric data items, like lakehouses. That way, they can make their own content (like semantic models and reports) without modifying the original data item.

Grant creators Read permission when:

- Creators should discover semantic models in the OneLake catalog to view their metadata and then request Build permission from the content owner.
- Creators should use Fabric data items like lakehouses or data warehouses to make their own content.

Step 3: Identify whether creators are outside the organization

If content creators outside the organization require access to items or data, then you'll need to take specific actions. The scenarios and considerations for content consumers also apply here.

You have several options to share content with external creators. The first two options are similar to how you share content with external consumers, except you grant Write permission to creators.

Option 1: Create duplicate identities for external creators

You can create a new account for the external consumer in your tenant that they must use to access that content.

Consider using duplicate identities for external creators when:

- External creators need to access other internal services, or they prefer to use a separate, internal account.
- Organizational policies or administrator settings don't allow for business-to-business (B2B) sharing.
- Limitations of Microsoft Entra B2B prevent you from doing what you need to do. Examples of limitations of Microsoft Entra B2B for creators include:
  - External guest users can't use Power BI Desktop to connect to semantic models or dataflows that are in the Power BI service and can't publish from Power BI Desktop.
  - External guest users can't set sensitivity labels.
  - External guest users can't install an on-premises data gateway.

Option 2: Create a Microsoft Entra B2B guest account for external creators

You can add the external creator as a guest user and share the item with them. Guest users can bring their own Power BI Pro or PPU license to use in the host tenant. However, guest users have several key limitations when creating and sharing Power BI content with others, such as the limitations for in-place semantic model sharing.

Consider distributing content to external guest users when:

- External creators have their own Power BI Pro and PPU licenses.
- You'll share a few content items with a few external creators to use.

- You'll distribute content to many external creators who have.

> **Important**
>
> Ensure that you test whether Microsoft Entra B2B is the best approach before you decide to use it. We recommend that you conduct a small trial with several creators before you commit to this content distribution strategy. During this trial, ensure that the creators perform all their necessary activities to discover any possible caveats or limitations that might prevent them from doing what they need to do.

External content consumers can use the OneLake catalog to view and access semantic models in their own tenant that have been shared with them. This concept is known as *in-place semantic model sharing*. It requires that Fabric administrators enable the Guest users can work with shared semantic models in their own tenants tenant setting.

External content consumers can connect to these semantic models to create composite semantic models and reports that they can publish to workspaces in their own tenant.

Microsoft Entra B2B and in-place semantic model sharing both have various considerations and limitations that you should keep in mind.

Step 4: Identify what creators need to access

You can grant access to creators at four different levels, depending on what they need to do:

- **Deployment pipeline access**: Provides creators access to a deployment pipeline and lets them trigger deployments to promote content between stages. Creators require access to the underlying workspaces to use deployment pipelines.
- **Workspace access**: Provides creators access to all the content in a workspace.
- **Item access**: Provides creators access to a single content item within a workspace.
- **Data access**: Provides creators access to specific data within an item. Creators require Write permission for an item to apply data access to data within that item.

Creator access might involve multiple options, depending on the circumstances. For example, you might provide a creator item access to a lakehouse, and data access to limit what they can use from that lakehouse. Then, you might provide that creator access to several workspaces to publish their semantic models and reports, and a deployment pipeline to deploy their content between those workspaces.

Follow the principle of least privilege when you provide creators access to deployment pipelines, workspaces, and items. For example, avoid assigning creators to workspace

roles with permissions that they don't need, like the Admin role. If they don't need to update the app or workspace configuration, don't assign creators the Member role.

The following sections help you to determine what access creators should get.

Option 1: Deployment pipeline access

You can give creators access to a deployment pipeline to manage content deployment between multiple workspaces. These creators can also change the deployment pipeline and review deployment history.

This approach is beneficial when:

- You'll use deployment pipelines to deploy content between workspaces.
- The content creator will be responsible for reviewing content during the deployment process.

Limit deployment pipeline access to only a few individuals who will be responsible for deciding when to deploy or release content to the next stage.

Option 2: Workspace access

You can give creators access to all content in a workspace by assigning them to either the Member or Contributor workspace role.

This approach is beneficial when:

- Creators will also get deployment pipeline access.
- Creators should collaborate in the workspace.
- You use Git integration to track and manage content changes in a remote repository that's linked to the workspace.
- Creators need to use a dataflow.

Ensure that you provide content creators access to a workspace to publish their content. If they should publish content to the same workspace as the original item they're using, ensure that you make it clear to them.

Option 3: Item access

You can give creators access to individual content items in a workspace by sharing via direct access. With this approach, you share a link to the item (similar to when you share individual items with consumers), or creators request access after they discover the content in the OneLake catalog.

This approach is beneficial when:

- Creators will only use one or a subset of items in a workspace.
- You want to share items with creators without providing workspace access.
- Creators should publish the content they create to their own, separate workspace.
- Creators should discover data items from the OneLake catalog.

> **Note**
>
> Sharing access to a data item like a lakehouse or semantic model will also give that user access to the underlying data in that item. It's not the same as granting data access to objects like tables or files by using the OneLake data access roles (or RBAC).

You can't share some individual content items, such as dataflows. For these items, you must provide workspace access for creators to view and use them.

Option 4: Data access

In some circumstances, content creators will need access to underlying data. In this case, creators might want to transform or enrich that data for other purposes. You have two options to share data in Fabric:

- OneLake data access role: You can assign someone to a data access role for data in a folder of a lakehouse. This approach lets you apply RBAC to OneLake data to limit what data someone can see.
- OneLake shortcuts: Though this isn't strictly a sharing technique, you can provide access to data by creating internal OneLake shortcuts to reference data within existing Fabric items. This approach lets you reuse data across teams and workspace while maintaining a single copy of the data.

These approaches are beneficial when:

- Your organization has Fabric capacity and uses lakehouses to store data for creators.
- You want to restrict the data that a creator can see and use from a lakehouse.
- Creators want to transform the underlying data to use in other data items.
- Creators are unable to use other approaches, like composite semantic models.

Creators must have ReadAll or Write permission to the lakehouse (either via workspace access or item access) to belong to a OneLake data access role. Creators must have

Write permission to the lakehouse to create shortcuts. You can't provide data access without first granting access to the item that contains that data.

Checklist - When sharing content with other creators, key decisions and considerations include:

- ✓ **Identify how content creators get access to content**: Decide whether you'll proactively grant access to content creators, or whether they're expected to find and request access to content themselves from the OneLake catalog.
- ✓ **Create a workspace for creators to publish content**: Ensure that creators have an available workspace to publish semantic models and reports, or to create other content in the Fabric portal.
- ✓ **Allocate licenses to creators**: Work with an appropriate administrator to ensure that creators have the appropriate per-user licenses to create and share content.
- ✓ **Decide what permissions creators should have**: Decide whether creators need Read, Build, or Write permission to existing content.
- ✓ **Share data gateways**: If necessary, set up data gateway access by adding consumers to data source connections in the User (or User with sharing) role.
- ✓ **Decide how you'll handle external users**: Identify whether you'll collaborate with any creators outside the organization. If so, decide the approach you'll take to share content with these users.
- ✓ **Add external guest users**: If you'll share content by using Microsoft Entra B2B, request that the relevant administrators add the guest users.
- ✓ **Decide whether creators need access to existing items**: Identify whether content creators need access to deployment pipelines, workspaces, individual content items, or data within lakehouses or KQL databases.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Security

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces a series of articles about Power BI security. The series of articles is targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They also support self-service users throughout the organization.
- **Content creators**: Self-service BI creators who set up permissions for the content they create and publish.

The series of articles is intended to expand upon the content in the Power BI security white paper. While the Power BI security white paper focuses on key technical topics such as authentication, data residency, and network isolation, the primary goal of the series is to provide you with considerations and decisions to help you plan for security and privacy.

It's important to plan to address challenges related to security, which include:

- Identifying and appropriately managing the volume and variety of data that's stored in many locations.
- Ensuring that sensitive data is appropriately stored and shared.
- Keeping pace with the regulatory landscape, which is ever-changing.
- Educating Power BI content creators on appropriate practices in relation to security and privacy.

> **Tip**

Also see the **information protection and data loss prevention** articles. They contain information that's complementary to this series of articles.

The focus for this series of articles is on security and privacy. It's organized into the following articles:

- **Tenant-level security planning**: Strategic decisions and actions you should consider that affect securing the content in your Power BI tenant. The focus is on strategic decisions that will impact consumers and content creators. It also includes strategies for file locations, external users, and using groups.
- **Report consumer security planning**: Tactical decisions and actions you should consider when planning how to deliver secure content to read-only consumers. The focus is primarily on report viewers and app viewers. It also describes techniques for how to enforce data security and the *Request access* workflow for consumers.
- **Content creator security planning**: Tactical decisions and actions you should consider when planning for enterprise and self-service creators, who create, manage, secure, and publish content. It also describes the data discovery experience and the *Request access* workflow for content creators.

> **Note**
>
> There are additional security planning topics not covered by these articles. We recommend that you also see Premium capacity permissions and Manage security roles of an on-premises data gateway.

Related content

In the next article in this series, learn about tenant-level security planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Tenant-level security planning

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This tenant-level security planning article is primarily targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Power BI in the organization.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with administrators of Power BI, information security teams, and other relevant teams.

This article might also be relevant for self-service Power BI creators who create, publish, and manage content in workspaces.

The series of articles is intended to expand upon the content in the Power BI security white paper. The Power BI security white paper focuses on key technical subjects such as authentication, data residency, and network isolation. The implementation planning series provides you with considerations and decisions to help you plan for security and privacy.

Because Power BI content can be used and secured in different ways, content creators make many tactical decisions. However, there are some strategic planning decisions to make at the tenant level, too. Those strategic planning decisions are the focus for this article.

We recommend that you make the tenant-level security decisions as early as possible because they affect everything else. Also, it's easier to make other security decisions once you have clarity on your overall security goals and objectives.

Power BI administration

The Fabric administrator is a high-privilege role that has significant control over Power BI. We recommend that you carefully consider who is assigned this role because a Fabric

administrator can perform many high-level functions, including:

- **Tenant settings management**: Fabric administrators can manage the tenant settings in the admin portal. They can enable or disable settings and allow or disallow specific users or groups within settings. It's important to understand that your tenant settings have a significant influence on the user experience.
- **Workspace role management**: Administrators can add themselves to the workspace admin role in the admin portal. They can potentially update workspace security to access any data or grant rights to other users to access any data in Fabric.
- **Personal workspace access**: Administrators can access contents and govern the personal workspace of any user.
- **Access to tenant metadata**: Administrators can access tenant-wide metadata, including the Power BI activity logs and activity events retrieved by the Power BI admin APIs.

> **Tip**
>
> As a best practice, you should assign between two and four users to the Fabric administrator role. That way, you can reduce risk while ensuring there's adequate coverage and cross-training.

An administrator of Power BI is assigned at least one of these built-in roles:

- Fabric Administrator
- Power Platform Administrator
- Global Administrator - this is a high privilege role and membership should be limited.

When assigning administrator roles, we recommend that you follow best practices.

> **Note**
>
> While a Power Platform administrator can manage the Power BI service, the inverse isn't true. Someone assigned to the Fabric administrator role can't manage other applications in the Power Platform.

Checklist - When planning for who will be a Fabric administrator, key decisions and actions include:

- ✓ **Identify who's currently assigned the administrator role**: Verify who's assigned to one of the roles that can administer Power BI.
- ✓ **Determine who should manage the Power BI service**: If there are too many administrators, create a plan to reduce the total number. If there are users assigned as administrators for Power BI who aren't well-suited to such a high-privilege role, create a plan to resolve the issue.
- ✓ **Clarify roles and responsibilities**: For each administrator of Power BI, ensure that their responsibilities are clear. Verify that appropriate cross-training has occurred.

Security and privacy strategies

You need to make some tenant-level decisions that relate to security and privacy. The tactics taken and the decisions you make rely on:

- Your data culture. The goal is to encourage a data culture that understands that security and protection of data is everyone's responsibility.
- Your content ownership and management strategies. The level of centralized and decentralized content management significantly affects how security is handled.
- Your content delivery scope strategies. The number of people who view content influences how security should be handled for the content.
- Your requirements to comply with global, national/regional, and industry regulations.

Here are a few examples of high-level security strategies. You might choose to make decisions that affect the entire organization.

- **Requirements for row-level security**: You can use row-level security (RLS) to restrict data access for specific users. That means different users see different data when accessing the same report. A Power BI semantic model or a data source (when using single sign-on) can enforce RLS. For more information, see the *Enforce data security based on consumer identity* section in the Report consumer security planning article.
- **Data discoverability**: Determine the extent to which data discoverability should be encouraged in Power BI. Discoverability affects who can find semantic models or datamarts in the OneLake catalog, and whether content authors are allowed to request access to those items by using the *Request access* workflow. For more information, see the customizable managed self-service BI usage scenario.
- **Data that's permitted to be stored in Power BI**: Determine whether there are certain types of data that shouldn't be stored in Power BI. For example, you might

specify that certain sensitive information types, like bank account numbers or social security numbers, can't be stored in a semantic model. For more information, see the Information protection and data loss prevention article.

- **Inbound private networking**: Determine whether there are requirements for network isolation by using private endpoints to access Power BI. When you use Azure Private Link, data traffic is sent by using the Microsoft private network backbone instead of going across the internet.
- **Outbound private networking**: Determine whether more security is required when connecting to data sources. The Virtual Network (VNet) data gateway enables secure outbound connectivity from Power BI to data sources within a VNet. You can use an Azure VNet data gateway when content is stored in a Premium workspace.

> **Important**
>
> When considering network isolation, work with your IT infrastructure and networking teams before you change any of the Fabric tenant settings. Azure Private Link allows for enhanced *inbound* security through private endpoints, while an Azure VNet gateway allows for enhanced *outbound* security when connecting to data sources. Azure VNet gateway is Microsoft-managed rather than customer-managed, so it eliminates the overhead of installing and monitoring on-premises gateways.

Some of your organizational-level decisions will result in firm governance policies, particularly when they relate to compliance. Other organizational-level decisions might result in guidance that you can provide to content creators who are responsible for managing and securing their own content. The resulting policies and guidelines should be included in your centralized portal, training materials, and communication plan.

> **Tip**
>
> See the other articles in this series for additional suggestions that relate to security planning for **report consumers** and **content creators**.

Checklist - When planning your high-level security strategies, key decisions and actions include:

- ✓ **Identify regulatory requirements related to security**: Investigate and document each requirement, including how you'll ensure compliance.
- ✓ **Identify high-level security strategies**: Determine which security requirements are important enough that they should be included in a governance policy.
- ✓ **Collaborate with other administrators**: Contact the relevant system administrator(s) to discuss how to meet security requirements and what technical prerequisites exist. Plan for doing a technical proof of concept.
- ✓ **Update Fabric tenant settings**: Set up each relevant Fabric tenant setting. Schedule follow-up reviews regularly.
- ✓ **Create and publish user guidance**: Create documentation for the high-level security strategies. Include details about the process and how a user can request an exemption from the standard process. Make this information available in your centralized portal and training materials.
- ✓ **Update training materials**: For the high-level security strategies, determine which requirements or guidelines you should include in user training materials.

Integration with Microsoft Entra ID

Power BI security is built upon the foundation of a Microsoft Entra tenant. The following Microsoft Entra concepts are relevant to the security of a Power BI tenant.

- **User access**: Access to the Power BI service requires a user account (in addition to a Power BI license: Free, Power BI Pro, or Premium Per User - PPU). You can add both internal users and guest users to Microsoft Entra ID, or they can be synchronized with an on-premises Active Directory (AD). For more information about guest users, see Strategy for external users.
- **Security groups**: Microsoft Entra security groups are required when making certain features available in the tenant settings. You might also need groups to effectively secure Power BI workspace content or for distributing content. For more information, see Strategy for using groups.
- **Conditional access policies**: You can set up conditional access to the Power BI service and the Power BI mobile app. Microsoft Entra conditional access can restrict authentication in various situations. For example, you could enforce policies that:
  - Require multifactor authentication for some or all users.
  - Allow only devices that comply with organizational policies.
  - Allow connectivity from a specific network or IP range(s).
  - Block connectivity from a non-domain-joined machine.
  - Block connectivity for a risky sign-on.
  - Allow only certain types of devices to connect.
  - Conditionally allow or deny access to Power BI for specific users.

- **Service principals**: You might need to create a Microsoft Entra app registration to provision a service principal. Service principal authentication is a recommended practice when a Power BI administrator wants to run unattended, scheduled, scripts that extract data by using the Power BI admin APIs. Service principals are also useful when embedding Power BI content in a custom application.
- **Real-time policies**: You can choose to set up real-time session control or access control policies, which involves both Microsoft Entra ID and Microsoft Defender for Cloud Apps. For example, you can prohibit the download of a report in the Power BI service when it has a specific sensitivity label. For more information, see the information protection and data loss prevention articles.

It might be difficult to find the right balance between unrestricted access and overly restrictive access (which frustrates users). The best strategy is to work with your Microsoft Entra administrator to understand what's currently set up. Try to remain responsive to the needs of the business while being mindful of necessary restrictions.

> **Tip**
>
> Many organizations have an on-premises Active Directory (AD) environment that they synchronize with Microsoft Entra ID in the cloud. This setup is known as a **hybrid identity** solution, which is out of scope for this article. The important concept to understand is that users, groups, and service principals must exist in Microsoft Entra ID for cloud-based services like Power BI to work. Having a hybrid identity solution will work for Power BI. We recommend talking to your Microsoft Entra administrators about the best solution for your organization.

Checklist - When identifying needs for Microsoft Entra integration, key decisions and actions include:

- ✓ **Work with Microsoft Entra administrators**: Collaborate with your Microsoft Entra administrators to find out what existing Microsoft Entra policies are in place. Determine whether there are any policies (current or planned) that'll affect the user experience in the Power BI service and/or in the Power BI mobile applications.
- ✓ **Decide when user access versus service principal should be used**: For automated operations, decide when to use a service principal instead of user access.
- ✓ **Create or update user guidance**: Determine whether there are security topics that you'll need to document for the Power BI user community. That way, they'll know

what to expect for using groups and conditional access policies.

Strategy for external users

Power BI supports Microsoft Entra Business-to-Business (B2B). External users, for instance from a customer or partner company, can be invited as guest users in Microsoft Entra ID for collaboration purposes. External users can work with Power BI and many other Azure and Microsoft 365 services.

> **Important**
>
> The **Microsoft Entra B2B white paper** is the best resource for learning about strategies for handling external users. This article is limited to describing the most important considerations that are relevant to planning.

There are advantages when an external user is from another organization that also has Microsoft Entra ID set up.

- **Home tenant manages the credentials**: The user's home tenant stays in control of their identity and management of credentials. You don't need to synchronize identities.
- **Home tenant manages the user's status**: When a user leaves that organization and the account is removed or disabled, with immediate effect, the user will no longer have access to your Power BI content. It's a significant advantage because you might not know when someone has left their organization.
- **Flexibility for user licensing**: There are cost-effective licensing options. An external user may already have a Power BI Pro or PPU license, in which case you don't need to assign one to them. It's also possible to grant them access to content in a Premium capacity or Fabric F64 or greater capacity workspace by assigning a Fabric (free) license to them.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Key settings

There are two aspects to enabling and managing how external user access will work:

- Microsoft Entra settings that are managed by a Microsoft Entra administrator. These Microsoft Entra settings are a prerequisite.
- Fabric tenant settings that are managed by a Power BI administrator in the admin portal. These settings will control the user experience in the Power BI service.

Guest invitation process

There are two ways to invite guest users to your tenant.

- **Planned invitations**: You can set up external users ahead of time in Microsoft Entra ID. That way, the guest account is ready whenever a Power BI user needs to use it for assigning permissions (for example, app permissions). Although it requires some up-front planning, it's the most consistent process because all Power BI security capabilities are supported. An administrator can use PowerShell to efficiently add a large number of external users.
- **Ad hoc invitations**: A guest account can be automatically generated in Microsoft Entra ID at the time that a Power BI user shares or distributes content to an external user (who wasn't previously set up). This approach is useful when you don't know ahead of time who the external users will be. However, this capability must first be enabled in Microsoft Entra ID. The ad hoc invitation approach works for ad hoc per-item permissions and app permissions.

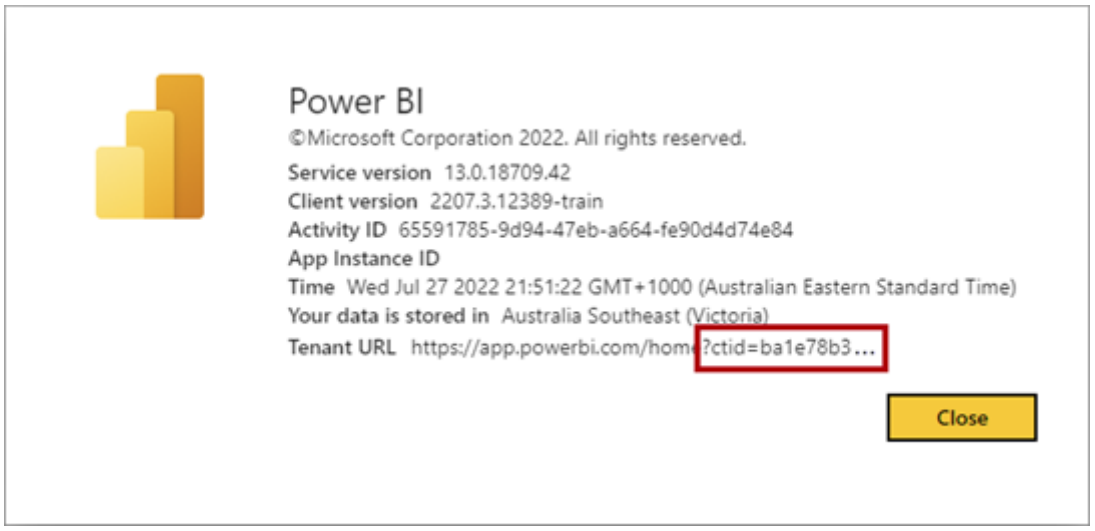
> **Tip**
>
> Not every security option in the Power BI service supports triggering an ad hoc invitation. For this reason, there's an inconsistent user experience when assigning permissions (for example workspace security versus per-item permissions versus app permissions). Whenever possible, we recommend that you use the planned invitation approach because it results in a consistent user experience.

Customer tenant ID

Every Microsoft Entra tenant has a globally unique identifier (GUID) known as the *tenant ID*. In Power BI, it's known as the *customer tenant ID (CTID)*. The CTID allows the Power BI service to locate content from the perspective of a different organizational tenant. You need to append the CTID to URLs when sharing content with an external user.

Here's an example of appending the CTID to a URL: `https://app.powerbi.com/Redirect?action=OpenApp&appId=abc123&ctid=def456`

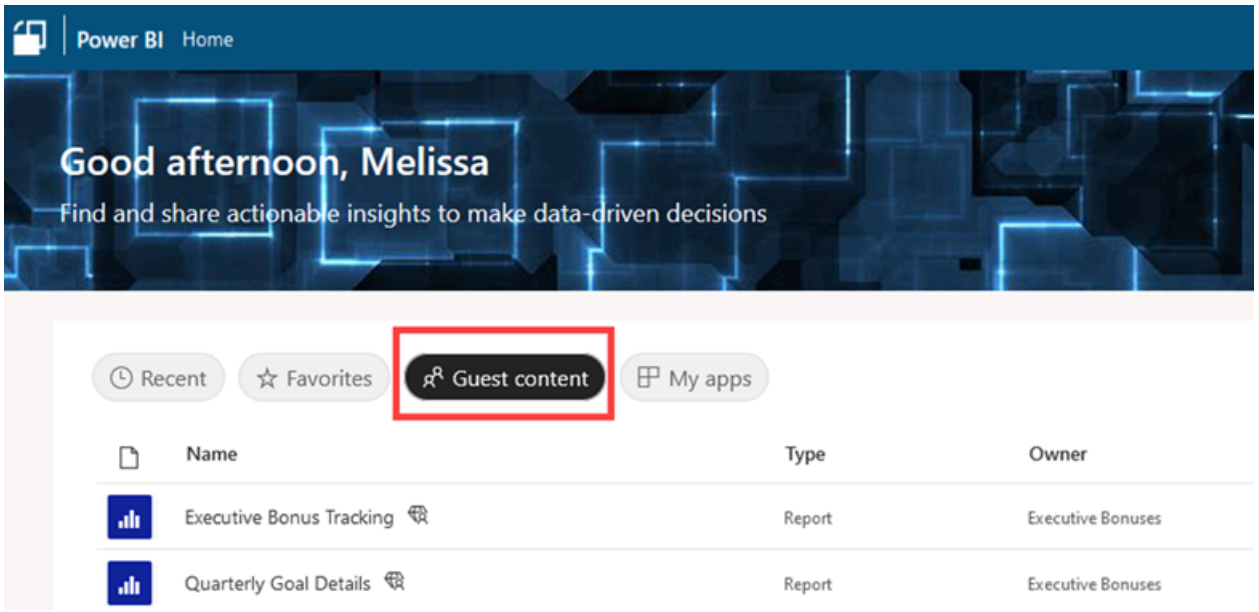
When you need to provide the CTID for your organization to an external user, you can find it in the Power BI service by opening the **About Power BI** dialog window. It's available from the **Help & Support (?)** menu located at the top-right of the Power BI service. The CTID is appended to the end of the tenant URL.



Organizational branding

When external guest access happens frequently in your organization, it's a good idea to use custom branding. Custom branding helps users identify which organizational tenant they're accessing. Branding elements can include include a logo, cover image, and theme color.

The following screenshot shows what the Power BI service looks like when accessed by a guest account. It includes a *Guest content* option, which is available when the CTID is appended to the URL.



External data sharing

Some organizations have a requirement to do more than share reports with external users. They intend to share semantic models with external users, such as partners, customers, or vendors.

The goal for in-place semantic model sharing (also known as *cross-tenant semantic model sharing*) is to allow external users to create their own customized reports and composite models by using data you create, manage, and provide. The original shared semantic model (created by you) remains in your Power BI tenant. The dependent reports and models are stored in the external user's Power BI tenant.

There are several security aspects for making in-place semantic model sharing work.

- **Tenant setting: *Allow guest users to work with shared semantic models in their own tenants***: This setting specifies whether the external data sharing feature can be used. It needs to be enabled for either of the other two settings (shown next) to take effect. It's enabled or disabled for the entire organization by the Power BI administrator.
- **Tenant setting: *Allow specific users to turn on external data sharing***: This setting specifies which groups of users can share data externally. The groups of users permitted here will be allowed to use the third setting (described next). This setting is managed by the Power BI administrator.
- **Semantic model setting: *External sharing***: This setting specifies whether that specific semantic model can be used by external users. This setting is managed by content creators and owners for each specific semantic model.
- **Semantic model permission: *Read* and *Build***: The standard semantic model permissions to support content creators are still in place.

> **Important**
>
> Typically, the term *consumer* is used to refer to view-only users who consume content that's produced by others in the organization. However, with in-place semantic model sharing, there's a *producer* of the semantic model and a *consumer* of the semantic model. In this situation, the consumer of the semantic model is usually a content creator in the other organization.

If row-level security is specified for your semantic model, it's honored for external users. For more information, see the *Enforce data security based on consumer identity* section in the Report consumer security planning article.

External user subscriptions

It's most common for external users to be managed as guest users in Microsoft Entra ID, as previously described. In addition to this common approach, Power BI provides other capabilities for distributing report subscriptions to users outside the organization.

The Power BI *Allow email subscriptions to be sent to external users* tenant setting specifies whether users are permitted to send email subscriptions to external users who aren't yet Microsoft Entra guest users. We recommend that you set this tenant setting to align with how strictly, or flexibly, your organization prefers to manage external user accounts.

> **Tip**
>
> Administrators can verify which external users are being sent subscriptions by using the **Get Report Subscriptions as Admin API**. The email address for the external user is shown. The principal type is *unresolved* because the external user isn't set up in Microsoft Entra ID.

Checklist - When planning for how to handle external guest users, key decisions and actions include:

- ✓ **Identify requirements for external users in Power BI**: Determine what use cases there are for external collaboration. Clarify the scenarios for using Power BI with

Microsoft Entra B2B. Determine whether collaboration with external users is a common or rare occurrence.

- ✓ **Determine the current Microsoft Entra settings**: Collaborate with your Microsoft Entra administrator to find out how external collaboration is currently set up. Determine what the impact will be on using B2B with Power BI.
- ✓ **Decide how to invite external users**: Collaborate with your Microsoft Entra administrators to decide how guest accounts will be created in Microsoft Entra ID. Decide whether ad hoc invitations will be allowed. Decide to what extent the planned invitation approach will be used. Ensure that the entire process is understood and documented.
- ✓ **Create and publish user guidance about external users**: Create documentation for your content creators that will guide them on how to share content with external users (particularly when the planned invitation process is required). Include information about limitations that external users will face if they intend to have external users edit and manage content. Publish this information to your centralized portal and training materials.
- ✓ **Determine how to handle external data sharing**: Decide whether external data sharing should be allowed, and whether it's limited to a specific set of approved content creators. Set the *Allow guest users to work with shared semantic models in their own tenants* tenant setting and the *Allow specific users to turn on external data sharing* tenant setting to align with your decision. Provide information about external data sharing for your semantic model creators. Publish this information to your centralized portal and training materials.
- ✓ **Determine how to handle Power BI licenses for external users**: If the guest user doesn't have an existing Power BI license, decide on the process to assign them a license. Ensure that the process is documented.
- ✓ **Include your CTID in relevant user documentation**: Record the URL that appends the tenant ID (CTID) in user documentation. Include examples for creators and consumers on how to use URLs that append the CTID.
- ✓ **Set up custom branding in Power BI**: In the admin portal, set up custom branding to help external users identify which organizational tenant they're accessing.
- ✓ **Verify or update tenant settings**: Check how the tenant settings are currently set up in the Power BI service. Update them as necessary based on the decisions made for managing external user access.

Strategy for file locations

There are different types of files that should be appropriately stored. So, it's important to help users understand expectations for where files and data should be located.

There can be risk associated with Power BI Desktop files and Excel workbooks because they can contain imported data. This data could include customer information, personally identifiable information (PII), proprietary information, or data that's subject to regulatory or compliance requirements.

> **Tip**
>
> It's easy to overlook the files that are stored outside of the Power BI service. We recommend that you consider them when you're planning for security.

Here are some of the types of files that might be involved in a Power BI implementation.

- **Source files**
  - **Power BI Desktop files**: The original files (.pbix) for content that's published to the Power BI service. When the file contains a data model, it might contain imported data.
  - **Excel workbooks**: Excel workbooks (.xlsx) might include connections to semantic models in the Power BI service. They might also contain exported data. They might even be original workbooks for content that's published to the Power BI service (as a workbook item in a workspace).
  - **Paginated report files**: The original report files (.rdl) files for content that's published to the Power BI service.
  - **Source data files**: Flat files (for example, .csv or .txt) or Excel workbooks that contain source data that's been imported into a Power BI model.
- **Exported and other files**
  - **Power BI Desktop files**: The .pbix files that have been downloaded from the Power BI service.
  - **PowerPoint and PDF files**: The PowerPoint presentations (.pptx) and PDF documents downloaded from the Power BI service.
  - **Excel and CSV files**: Data exported from reports in the Power BI service.
  - **Paginated report files**: The files exported from paginated reports in the Power BI service. Excel, PDF, and PowerPoint are supported. Other export file formats exist for paginated reports as well, including Word, XML, or web archive. When using the export files to reports API, image formats are also supported.
  - **Email files**: Email images and attachments from subscriptions.

You'll need to make some decisions about where users can or can't store files. Typically, that process involves creating a governance policy that users can refer to. The locations for source files and exported files should be secured to ensure appropriate access by authorized users.

Here are some recommendations for working with files.

- **Store files in a shared library**: Use a Teams site, a SharePoint library, or a OneDrive for work or school shared library. Avoid using personal libraries and drives. Ensure that the storage location is backed up. Also ensure that the storage location has versioning enabled so that it's possible to roll back to a previous version.
- **Use the Power BI service as much as possible**: Whenever possible, use the Power BI service for sharing and distribution of content. That way, there's full auditing of access always. Storing and sharing files on a file system should be reserved for the small number of users who are collaborating on content.
- **Don't use email**: Discourage the use of email for sharing files. When someone emails an Excel workbook or a Power BI Desktop file to 10 users, it results in 10 copies of the file. There's always the risk of including an incorrect (internal or external) email address. Also, there's a greater risk the file will be forwarded to someone else. (To minimize this risk, work with your Exchange Online administrator to implement rules to block attachments based on conditions of size or type of file extension. Other data loss prevention strategies for Power BI are described in the information protection and data loss prevention articles.)
- **Use template files**: Occasionally, there's a legitimate need to share a Power BI Desktop file with someone else. In this case, consider creating and sharing a Power BI Desktop template (.pbit) file. A template file only contains metadata, so it's smaller in size than the source file. This technique will require the recipient to input data source credentials to refresh the model data.

There are tenant settings in the admin portal that control which export formats users are permitted to use when exporting from the Power BI service. It's important to review and set these settings. It's a complementary activity to planning for the file locations that should be used for the exported files.

> **Tip**
>
> Certain export formats support end-to-end information protection by using encryption. Due to regulatory requirements, some organizations have a valid need to restrict which export formats users can use. The **Information protection for Power BI** article describes factors to consider when deciding which export formats to enable or disable in the tenant settings. In most cases, we recommend that you restrict exporting capabilities only when you must meet specific regulatory requirements. You can use the **Power BI activity log** to identify which users are performing many exports. You can then teach these users about more efficient and secure alternatives.

Checklist - When planning for file locations, the key decisions and actions include:

- ✓ **Identify where files should be located**: Decide where files should be stored. Determine whether there are specific locations that shouldn't be used.
- ✓ **Create and publish documentation about file locations**: Create user documentation that clarifies the responsibilities for managing and securing files. It should also describe any locations where files should (or shouldn't) be stored. Publish this information to your centralized portal and training materials.
- ✓ **Set the tenant settings for exports**: Review and set each tenant setting related to export formats you want to support.

Strategy for using groups

We recommend using Microsoft Entra security groups to secure Power BI content for the following reasons.

- **Reduced maintenance**: The security group membership can be modified without the need to modify the permissions for the Power BI content. New users can be added to the group, and unnecessary users can be removed from the group.
- **Improved accuracy**: Because the group membership changes are made once, it results in more accurate permission assignments. If an error is detected, it can be more easily corrected.
- **Delegation**: You can delegate the responsibility of managing group membership to the group owner.

High-level group decisions

There are some strategic decisions to be made regarding how groups will be used.

Permission to create and manage groups

There are two key decisions to make about creating and managing groups.

- **Who's allowed to create a group?** Commonly, only IT can create security groups. However, it's possible to add users to the built-in *Groups administrator* Microsoft Entra role. That way, certain trusted users, like Power BI champions or satellite members of your COE, can create groups for their business unit.

- **Who's allowed to manage members of a group?** It's common that IT manages group membership. However, it's possible to specify one or more *group owners* who are permitted to add and remove group members. Using *self-service group management* is helpful when a decentralized team or satellite members of the COE are permitted to manage the membership of Power BI-specific groups.

> **Tip**
>
> Allowing **self-service group management** and specifying decentralized **group owners** are great ways to balance efficiency and speed with governance.

Planning for Power BI groups

It's important that you create a high-level strategy for how to use groups for securing Power BI content and many other uses.

Various use cases for groups

Consider the following use cases for groups.

Expand table

Use case	Description	Example group name
Communicating with the Power BI Center of Excellence (COE)	Includes all users associated with the COE, including all core and satellite members of the COE. Depending on your needs, you might also create a separate group for only the core members. It's likely to be a Microsoft 365 group that's correlated with a Teams site.	• Power BI Center of Excellence
Communicating with the Power BI leadership team	Includes the executive sponsor and representatives from business units who collaborate on leading the Power BI initiative in the organization.	• Power BI steering committee
Communicating with the Power BI user community	Includes all users who are assigned any type of Power BI user license. It's useful for making announcements to all Power BI users in your organization. It's likely to be a Microsoft 365 group that's correlated with a Teams site.	• Power BI community
Supporting the Power BI user community	Includes help desk users who directly interact with the user community to handle Power BI support issues. This email address (and Teams site, if	• Power BI user support

Use case	Description	Example group name
	applicable) is available and visible to the user population.	
Providing escalated support	Includes specific users, usually from the Power BI COE, who provide escalated support. This email address (and Teams site, if applicable) is typically private, for use only by the user support team.	• Power BI escalated user support
Administering the Power BI service	Includes specific users who are allowed to administer the Power BI service. Optionally, members of this group can be correlated to the role in Microsoft 365 to simplify management.	• Power BI administrators
Notifying allowed features and gradual roll outs of features	Includes users allowed for a specific tenant setting in the admin portal (if the feature will be limited), or if the feature is to be rolled out gradually to groups of users. Many of the tenant settings will require you to create a new Power BI-specific group.	• Power BI workspace creators • Power BI external data sharing
Managing data gateways	Includes one or more groups of users who are allowed to administer a gateway cluster. There might be several groups of this type when there are multiple gateways or when decentralized teams manage gateways.	• Power BI gateway administrators • Power BI gateway data source creators • Power BI gateway data source owners • Power BI gateway data source users
Managing Premium capacities	Includes users allowed to manage a Premium capacity. There might be several groups of this type when there are multiple capacities or when decentralized teams manage capacities.	• Power BI capacity contributors
Securing workspaces, apps, and items	Many groups that are based on subject areas and allowed access for managing security of Power BI workspace roles, app permissions, and per-item permissions.	• Power BI workspace administrators • Power BI workspace members

Use case	Description	Example group name
		• Power BI workspace contributors • Power BI workspace viewers • Power BI app viewers
Deploying content	Includes the users that can deploy content by using a Power BI deployment pipeline. This group is used in conjunction with workspace permissions.	• Power BI deployment pipeline administrators
Automating administrative operations	Includes the service principals that are allowed to use Power BI APIs for embedding or administrative purposes.	• Power BI service principals

Groups for Fabric tenant settings

Depending on the internal processes you have in place, you'll have other groups that are necessary. Those groups are helpful when managing the tenant settings. Here are some examples.

- **Power BI workspace creators**: Useful when you need to limit who can create workspaces. It's used to set up the *Create workspaces* tenant setting.
- **Power BI certification subject matter experts**: Useful to specify who's permitted to use the certified endorsement for content. It's used to set up the *Certification* tenant setting.
- **Power BI approved content creators**: Useful when you require approval, or training, or a policy acknowledgment for installation of Power BI Desktop, or for obtaining a Power BI Pro or PPU license. It's used by tenant settings that encourage content creation capabilities, such as *Allow DirectQuery connections to Power BI semantic models*, *Push apps to end users*, *Allow XMLA endpoints*, and others.
- **Power BI external tool users**: Useful when you allow the use of external tools for a selective group of users. It's used by group policy, or when software installations or requests must be carefully controlled.
- **Power BI custom developers**: Useful when you need to control who's permitted to embed content in other applications outside of Power BI. It's used to set up the

Embed content in apps tenant setting.

- **Power BI public publishing**: Useful when you need to limit who can publish data publicly. It's used to set up the *Publish to web* tenant setting.
- **Power BI share to entire organization**: Useful when you need to restrict who can share a link with everyone in the organization. It's used to set up the *Allow shareable links to grant access to everyone in your organization* tenant setting.
- **Power BI external data sharing**: Useful when you need to allow certain users to share semantic models with external users. It's used to set up the *Allow specific users to turn on external data sharing* tenant setting.
- **Power BI guest user access licensed**: Useful when you need to group approved external users who are granted a license by your organization. It's used to set up the *Allow Microsoft Entra guest users access to Power BI* tenant setting.
- **Power BI guest user access BYOL**: Useful when you need to group approved external users who bring their own license (BYOL) from their home organization. It's used to set up the *Allow Microsoft Entra guest users access to Power BI* tenant setting.

> **Tip**
>
> For considerations about using groups when planning for workspace access, see the **Workspace-level planning** article. For information about planning for securing workspaces, apps, and items, see the **Report consumer security planning** article.

Type of group

You can create different types of groups.

- **Security group**: A security group is the best choice when your primary goal is to grant access to a resource.
- **Mail-enabled security group**: When you need to grant access to a resource and distribute messages to the entire group by email, a mail-enabled security group is a good choice.
- **Microsoft 365 group**: This type of group has a Teams site and an email address. It's the best choice when the primary goal is communication or collaboration in a Teams site. A Microsoft 365 group only has members and owners; there isn't a viewer role. For this reason, its primary purpose is collaboration. This type of group was formerly known as an Office 365 group, modern group, or unified group.
- **Distribution group**: You can use a distribution group to send a broadcast notification to a list of users. Today, it's considered to be a legacy concept that

provides backwards compatibility. For new use cases, we recommend that you create a mail-enabled security group instead.

When you request a new group, or you intend to use an existing group, it's important to be aware of its type. The type of group can determine how it's used and managed.

- **Power BI permissions**: Not every type of group is supported for every type of security operation. Security groups (including mail-enabled security groups) offer the highest coverage when it comes to setting Power BI security options. Microsoft documentation generally recommends Microsoft 365 groups. However, in the case of Power BI, they aren't as capable as security groups. For more information about Power BI permissions, see the later articles in this series on security planning.
- **Fabric tenant settings**: You can only use security groups (including mail-enabled security groups) when allowing or disallowing groups of users to work with tenant settings.
- **Advanced Microsoft Entra features**: Certain types of advanced features aren't supported for all group types. For example, Microsoft 365 groups don't support nesting of groups within a group. Although some group types support dynamic group membership based on user attributes in Microsoft Entra ID, groups that use dynamic membership aren't supported for Power BI.
- **Managed differently**: Your request to create or manage a group might be routed to a different administrator based on the type of group (mail-enabled security groups and distribution groups are managed in Exchange). Therefore, your internal process will differ depending on the type of group.

Group naming convention

It's likely that you'll end up with many groups in Microsoft Entra ID to support your Power BI implementation. Therefore, it's important to have an agreed-upon pattern for how groups are named. A good naming convention will help to determine the purpose of the group and make it simpler to manage.

Consider using the following standard naming convention: *<Prefix> <Purpose> - <Topic/Scope/Department> <[Environment]>*

The following list describes each part of the naming convention.

- **Prefix**: Used to group all Power BI groups together. When the group will be used for more than one analytical tool, your prefix might be just BI, rather than Power BI. In that case, the text that describes the purpose will be more generic so that it relates to more than one analytical tool.

- **Purpose**: The purpose will vary. It could be for a workspace role, app permissions, item-level permissions, row-level security, or other purpose. Sometimes multiple purposes can be satisfied with a single group.
- **Topic/Scope/Department**: Used to clarify who the group applies to. It will often describe the group membership. It can also refer to who manages the group. Sometimes a single group can be used for multiple purposes. For example, a collection of finance workspaces could be managed with a single group.
- **Environment**: Optional. Useful to differentiate between development, test, and production.

Here are some example group names that apply the standard naming convention.

- Power BI workspace admins - Finance [Dev]
- Power BI workspace members - Finance [Dev]
- Power BI workspace contributors - Finance [Dev]
- Power BI workspace viewers - Finance [Dev]
- Power BI app viewers - Finance
- Power BI gateway administrators - Enterprise BI
- Power BI gateway administrators - Finance

Decisions per group

When planning for which groups you'll need, several decisions must be made.

When a content creator or owner requests a new group, ideally they use a form to provide the following information.

- **Name and purpose**: A suggested group name and its intended purpose. Consider including *Power BI* (or just *BI* when you have multiple BI tools) in the group name to clearly indicate the scope of the group.
- **Email address**: An email address when communication is also required for the group members. Not all types of groups need to be mail-enabled.
- **Type of group**: Options include security group, mail-enabled security group, Microsoft 365 group, and distribution group.
- **Group owner**: Who's allowed to own and manage the members of the group.
- **Group membership**: The intended users who will be members of the group. Consider whether external users and internal users can be added, or whether there's a justification for putting external users into a different group.
- **Use of just-in-time group member assignment**: You can use Privileged Identity Management (PIM) to allow time-boxed, just-in-time, access to a group. This service can be helpful when users require temporary access. PIM is also helpful for Power BI administrators who need occasional access.

> **Tip**
>
> Existing groups that are based on the organizational chart don't always work well for Power BI purposes. Use existing groups when they meet your needs. However, be prepared to create Power BI-specific groups when the need arises.

Checklist - When creating your strategy for how to use groups, key decisions and actions include:

- ✓ **Decide on the strategy for the use of groups**: Determine the use cases and purposes you'll need to use groups. Be specific about when security should be applied by using user accounts versus when a group is required or preferred.
- ✓ **Create a naming convention for Power BI-specific groups**: Ensure that a consistent naming convention is in use for groups that will support Power BI communication, features, administration, or security.
- ✓ **Decide who is allowed to create groups**: Clarify whether all group creation is required to go through IT. Or whether certain individuals (like satellite members of the COE) can be granted permission to create groups for their business unit.
- ✓ **Create a process for how to request a new group**: Create a form for users to request the creation of a new group. Ensure that there's a process in place to respond quickly to new requests. Bear in mind that if requests are delayed, users might be tempted to start assigning permissions to individuals accounts.
- ✓ **Decide when decentralized group management is allowed**: For groups that apply to a specific team, decide when it's acceptable for a group owner (outside of IT) to manage members in the group.
- ✓ **Decide whether just-in-time group membership will be used**: Determine whether Privileged Identity Management will be useful. If so, determine which groups it can be used for (such as the Power BI administrator group).
- ✓ **Review which groups currently exist**: Determine which existing groups can be used, and which groups need to be created.
- ✓ **Review each tenant setting**: For each tenant setting, determine whether it'll be allowed or disallowed for a specific set of users. Determine whether a new group needs to be created to set up the tenant setting.
- ✓ **Create and publish guidance for users about groups**: Include documentation for content creators that includes requirements, or preferences, for using groups.

Ensure that they know what to ask for when they request a new group. Publish this information to your centralized portal and training materials.

Related content

In the next article in this series, learn about ways to securely deliver content to read-only report consumers.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Report consumer security planning

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This security planning article describes strategies for read-only consumers. The focus is on viewer permissions for reports and apps, and how to enforce data security. It's primarily targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.
- **Content creators and owners**: Self-service BI creators who need to create, publish, secure, and manage content that other users consume.

The series of articles is intended to expand upon the content in the Power BI security white paper. While the Power BI security white paper focuses on key technical topics such as authentication, data residency, and network isolation, the primary goal of the series is to provide you with considerations and decisions to help you plan for security and privacy.

In an organization, many users are classified as *consumers*. Consumers view content that other users have created and published. Consumers are the focus of this article. For security planning focused on content creators and owners, see the Content creator security planning article.

To get the most from this article, it's helpful to understand the meaning of the terms *sharing* and *distribution* in the context of Power BI.

Sharing is where one user gives another user (or group of users) access to a specific item of content. The sharing capability in the Power BI service is scoped to one item. It most commonly takes place between individuals who know each other and work closely together.

Distribution is where content is delivered to other users, who are known as recipients. It often involves a larger number of users across multiple teams. Recipients might not have explicitly requested the content, but it's recognized that they need it to perform their role. Recipients who consume distributed content might or might not know the original creator of the content. As such, distribution as a concept is more formal than sharing.

When you talk with other people, determine whether they're using the term *sharing* in a general way, or literally. Use of the term *sharing* can be interpreted in two ways.

- The term *sharing* is often used in a general way related to sharing content with colleagues. There are several techniques for delivering read-only content, which are described in this article.
- *Sharing* is also a specific feature in Power BI. It's a capability where a user or group is granted access to a single item. Sharing links and direct access sharing are described in this article.

> **Important**
>
> The **Power BI administrator** role has been renamed. The new name of the role is **Fabric administrator**.

Strategy for read-only consumers

In the Power BI service, consumers can view a report or dashboard when they have permission to both:

- View the Power BI item that contains the visualizations (such as a report or dashboard).
- Read the underlying data (semantic model or other source).

You can provide read-only access to consumers by using different techniques. The common techniques used by self-service content creators include:

- Granting users and groups access to a Power BI app.
- Adding users and groups to a Power BI workspace Viewer role.
- Providing users and groups per-item permissions by using a sharing link.
- Providing users and groups per-item permissions by using direct access.

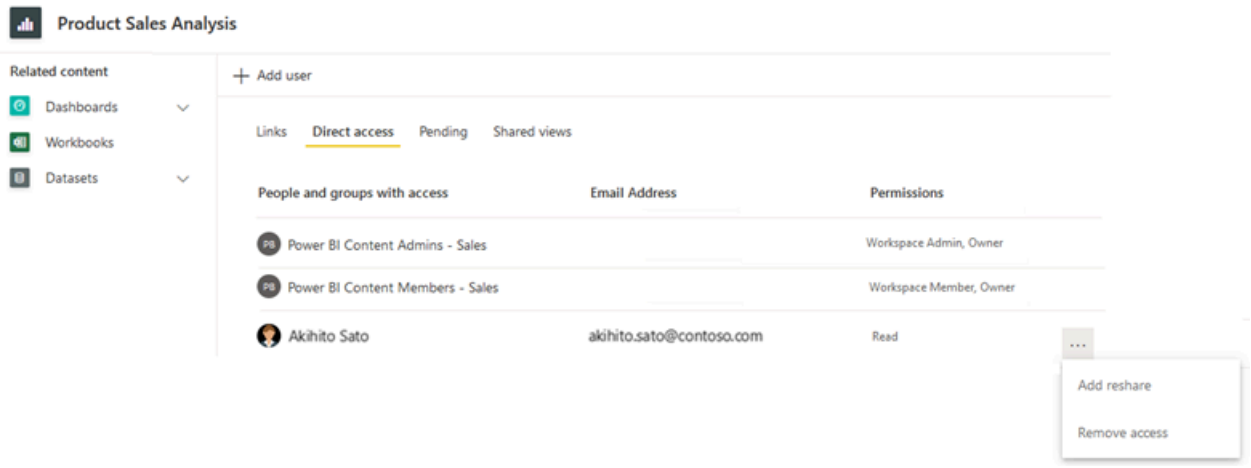
The Power BI app and Power BI workspace Viewer role options involve managing permissions for a set of items. The two per-item permissions techniques involve managing permissions for one individual item.

> **Tip**
>
> Generally, it's a best practice to use a Power BI app for most consumers. Occasionally the workspace Viewer role could also be appropriate. Both Power BI apps and the workspace Viewer role allow managing permissions for many items, and should be used whenever possible. Managing permissions for individual items can be tedious, time consuming, and error prone. In contrast, managing a set of items reduces maintenance and improves accuracy.

When reviewing security settings for an item, you might see that its permissions are either:

- Inherited from the workspace or an app.
- Applied directly to the item.

In the following screenshot, the *Direct access* permissions are shown for a report. In this instance, the workspace Admin and Member roles are each assigned to a group. These roles are shown for the report because the report-level access is inherited from the workspace. There's also one user who has Read permissions applied directly to the report.



The strategy you choose for read-only consumers can be different. It should be based on the individual solution, the preferences of who manages the solution, or the needs of the consumer. The remainder of this section describes when to consider using each of the available techniques.

Checklist - When creating your strategy for how to provide content to read-only consumers, key decisions and actions include:

- ✓ **Assess your existing strategy for read-only consumers**: Verify how content is currently distributed and shared to consumers. Identify whether there are opportunities for improvement.
- ✓ **Decide on your strategy for read-only consumers**: Consider what your preferences are for using app permissions, workspace roles, or per-item permissions. If changes are necessary to meet these preferences, create a plan for making improvements.

Power BI app permissions

A Power BI app delivers a collection of reports, dashboards, and workbooks to consumers. An app provides the best user experience for consumers because:

- The app's navigation pane provides a simple and intuitive user experience. It's a nicer experience than accessing content directly in a workspace.
- Content can be logically organized into sections (which are like folders) in the app's navigation pane.
- Consumers only have access to specific items that have been explicitly included in the app for their audience.
- Links to additional information, documentation, or other content can be added to the navigation pane for their audience.
- There's a built-in *Request access* workflow.

> **Note**
>
> All references to an app in this article refer to a Power BI app. It's a different concept from **Power Apps**. It's also a different concept than the Power BI mobile applications. In this section, the focus is on organizational apps rather than template apps.

You can create one app for each workspace as a formal way to distribute some, or all, workspace content. Apps are a good way to distribute content broadly within an organization, especially to users that you don't know or don't collaborate with closely.

> **Tip**
>
> For more information about using a Power BI app for broad content distribution, see the **enterprise BI** usage scenario. We recommend that content creators who need to distribute content consider creating an app as their first choice.

You manage app permissions separately from workspace roles. The separation of permissions has two advantages. It encourages:

- Granting workspace access to content creators. It includes users that are actively collaborating on the content, like semantic model creators, report creators, and testers.
- Granting app permissions to consumers. Unlike workspace permissions, app permissions are always read-only (or none).

All users with workspace access can automatically view the app (when a Power BI app has been published for the workspace). Due to this behavior, you can conceptually think of workspace roles as being *inherited* by each app audience. Some users with workspace access can also update the Power BI app, depending on their assigned workspace role.

> **Tip**
>
> For more information about workspace access, see the **Content creator security planning** article.

Using an app to distribute content to read-only consumers is the best choice when:

- You want users to be able to view only specific items that are visible for that audience (rather than all items within the underlying workspace).
- You want to manage read-only permissions for the app separately from the workspace.
- You want simpler permission management for read-only users than per-item permissions.
- You want to ensure that row-level security is enforced for consumers (when they have read-only permission on the underlying semantic model).
- You want to ensure that consumers can't view new and changed reports until the app is republished.

While it's true that changes to reports and dashboards aren't visible to users of the app until the app is republished, there are two considerations that require caution.

- **Immediate semantic model changes**: Semantic model changes always take effect immediately. For example, if you introduce breaking changes to a semantic model in the workspace, it could inadvertently result in reports becoming unstable (even though they haven't been republished in the app). There are two ways to mitigate this risk: First, do all development work in Power BI Desktop (separate from the workspace). Second, insulate the production app by using separate workspaces for development and test. (Optionally, you can achieve a higher level of control over

deploying workspace content from development to test and production by using deployment pipelines.)

- **Content and permissions are published together**: When you publish an app, its permissions are published at the same time as the content. For example, you might have report changes in a workspace that aren't yet complete, fully tested, or approved. So, you can't republish the app merely to update permissions. To mitigate this risk, assign app permissions to security group(s), and use security group memberships (instead of individual users) when granting app permissions. Avoid republishing an app merely to apply permission changes.

App audience

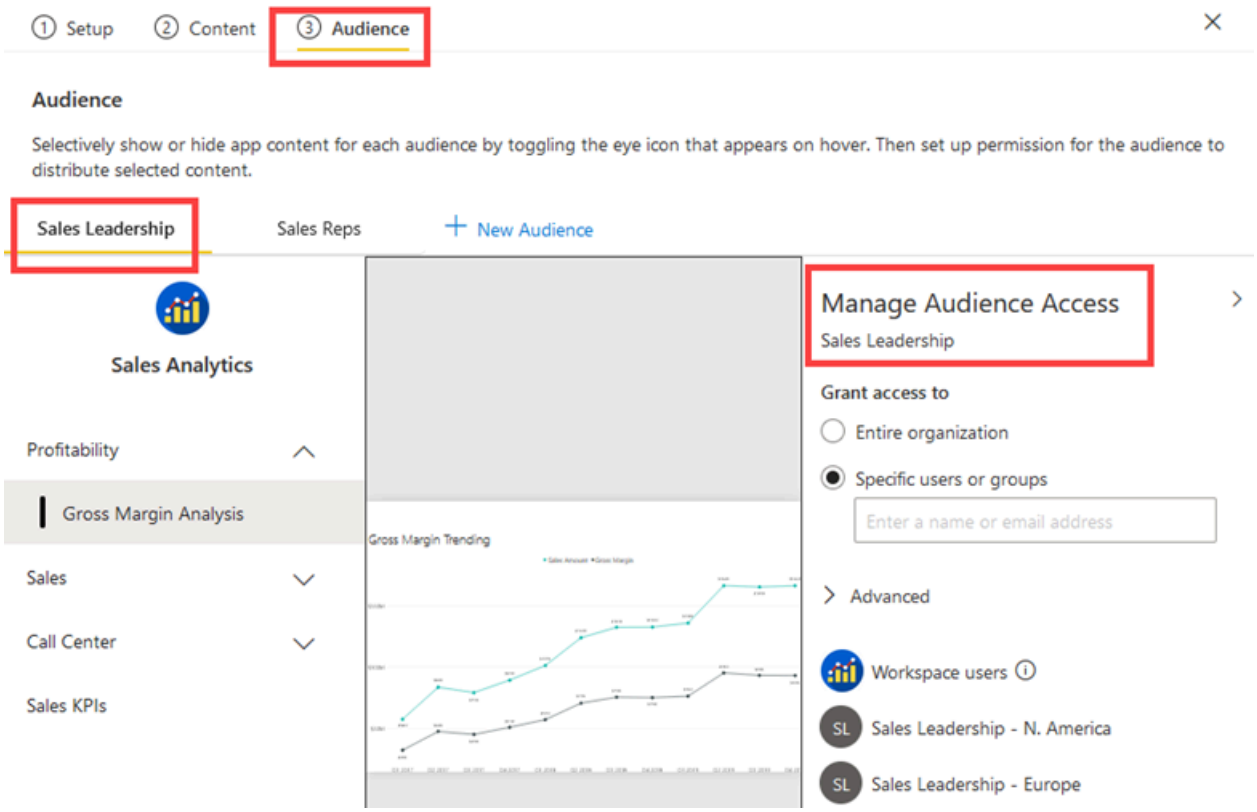
Each workspace in the Power BI service can have only one Power BI app. However, within the app you can create one or more *audiences*. Consider the following scenario.

- You have five sales reports that are distributed to many users throughout your global sales organization.
- One audience is defined in the app for the sales representatives. This audience can view three of the five reports.
- Another audience is defined in the app for the sales leadership team. This audience can view all five reports, including the two reports that aren't available to sales representatives.

This capability to *mix and match* content and audiences has the following advantages.

- Certain reports can be available for viewing by multiple audiences. So, creating multiple audiences removes the need to duplicate content across different workspaces.
- Certain reports should be available to only one audience. So, content for that one audience can reside in the same workspace as other related content.

The following screenshot shows an app with two audiences: **Sales Leadership** and **Sales Reps**. The *Manage Audience Access* pane provides access to the **Sales Leadership** audience group for two security groups: *Sales Leadership-North America* and *Sales Leadership-Europe*. The *Gross Margin Analysis* report that's shown in the screenshot for the **Sales Leadership** audience group isn't available to the **Sales Reps** audience group.



> **Note**
>
> The term *audience group* is sometimes used. It isn't a direct reference to the use of security groups. It includes members of the target audience who will see the collection of content within a Power BI app. While you can assign individual users to an audience, it's a best practice to assign security groups, Microsoft 365 groups, or distribution groups whenever practical. For more information, see the strategy for using groups in the Tenant-level security planning article.

When you manage permissions for an app, on the *Direct Access* page you can view the members of each audience. You can also see users with a workspace role listed under the *All* audience. You can't update the app permissions from the *Direct Access* page. Instead, you must republish the app. You can, however, update app permissions from the *Pending* page when there are open access requests for the app.

> **Tip**
>
> The primary use case for using app audiences is to define specific permissions for different sets of users. However, you can get a little creative when using audiences. A user can be a member of multiple audiences, and each audience is shown to viewers of the app as a secondary set of menus. For example, you can create an audience named *Start Here* that contains information about how to use the app, who to contact, how to provide feedback, and how to get help. Or, you can create

an audience named *KPI Definitions* that includes a data dictionary. Providing this type of information helps new users and improves **solution adoption** efforts.

App permission options

When you create (or republish) an app, each audience has a *Manage Audience Access* pane. In that pane, the following permissions are available.

- **Grant access to**: For each audience, you can grant access to individual users and groups. It's possible to publish the app to the entire organization when it's enabled by the *Publish apps to the entire organization* tenant setting, and the app isn't installed automatically. Whenever possible, we recommend that you assign groups to audiences because adding or removing users involves republishing the app. Everyone with workspace access automatically has permission to view or update the app depending on their workspace role.
- **Semantic model permissions**: Two types of semantic model permissions can be granted while publishing an app:
  - **Semantic model Reshare**: When enabled, app users are granted the Reshare permission to the underlying semantic model(s) with others. It makes sense to enable this option when the underlying semantic model(s) can be readily reshared with anyone. We recommend that you get approval from the semantic model owner(s) before granting the Reshare permission to an app audience.
  - **Semantic model Build**: When enabled, app users are granted the Build permission for the semantic models. Build permission allows users to create new reports, export underlying data from reports, and more. We recommend that you get approval from the semantic model owner(s) before granting Build permission to an app audience.

The capability to add the semantic model Reshare or Build permissions while publishing an app is convenient. However, we recommend that you consider managing app permissions and semantic model permissions separately. Here are the reasons why.

- **Shared semantic model might be in a separate workspace**: If the semantic model is published to a separate workspace from the app, you'll need to manage its permissions directly. The ability to add Read, Build or Reshare permissions while publishing an app only works for semantic models that are in the same workspace as the app. For this reason, we recommend that you get into the habit of managing semantic model permissions independently.
- **Semantic model permissions are managed separately**: If you remove or change permissions for an app, that action only affects the app. It doesn't automatically remove any semantic model permissions that were previously assigned. In this way,

you can think of the app permissions and semantic model permissions as being *decoupled*. You'll need to manage the semantic model directly, separately from the app, when semantic model permissions change or need to be removed.

- **Semantic model permissions should be controlled**: Granting semantic model permissions through an app removes control from the semantic model owner. Granting the Reshare permission relies on good judgment by users who are choosing to reshare the semantic model(s). Your internal governance or security guidelines can become more difficult to manage when resharing is allowed.
- **Consumers and creators have different goals**: Typically, there are many more content consumers than creators in an organization. In line with the principle of least privilege, consumers only need Read permission for the underlying semantic model. They don't need Build permission unless they intend to create new reports.

> **Tip**
>
> For more information about when to use separate data workspaces and reporting workspaces, see the **Workspace-level planning** article.

App pre-installation rights

After you publish a Power BI app, a user typically needs to install it so they can open it. A user can install an app from the Apps page in the Power BI service, or by using a link they've received from another user. They'll be able to find (and install) an app when they're included in at least one audience of the app.

An alternative approach to install an app is to *push* it to app consumers. It results in the pre-installation of the app so that it automatically shows up in the Apps page in the Power BI service. This approach is a convenience for consumers because they don't need to find and install the app. However, pre-installed apps can become an annoyance for users because they might become overwhelmed by too many apps that aren't relevant to them.

The *Push apps to end users* tenant setting controls who's allowed to automatically install apps. We recommend that you use this feature because it's convenient for users. However, we also recommend that you educate your content creators on when to use it so that it isn't overused.

> **Tip**
>
> When publishing an app, if you select the option to install the app automatically, you can't set the audience to be the entire organization (if enabled by the *Push*

apps to end users tenant setting).

Checklist - When creating your strategy for using apps for content viewers, key decisions and actions include:

- ✓ **Decide on the strategy for use of apps**: Define your preferences for how to use apps. Ensure that it aligns with your overall strategy for read-only consumers.
- ✓ **Decide who can publish apps to the entire organization**: Decide which report creators are able to publish to the entire organization. Set the *Publish apps to the entire organization* tenant setting to align with this decision.
- ✓ **Decide who can push apps to end users**: Decide which Power BI report creators can pre-install apps. Set the *Push apps to end users* tenant setting to align with this decision.
- ✓ **Create and publish guidance for content creators**: Provide documentation and training for content creators. Include requirements and preferences for how to use apps most effectively.
- ✓ **Determine how to handle app access requests**: Ensure that a process is in place to assign contacts and handle app access requests in a timely manner.

Workspace Viewer role

As described in the Workspace planning articles, the primary purpose of a workspace is collaboration. Workspace collaborators, such as semantic model creators, report creators, and testers, should be assigned to one of three roles: Contributor, Member, or Admin. These roles are described in the Content creator security planning article.

You can assign the workspace Viewer role to consumers. Allowing consumers to access content directly in a workspace can make sense for small teams and informal teams who work closely together.

Allowing consumers to access workspace content directly is a good choice when:

- The formality of an app, with its separate permissions, isn't necessary.
- Viewers are permitted to view all items stored within the workspace.
- You want simpler permissions management than per-item permissions.
- Workspace users might also view an app (when an app is published for the workspace).
- The intention is for viewers to review content before it's published in an app.

Here are some suggestions to support workspace viewers.

- Organize content in each workspace so that the items are easily located by report consumers and so they align well with security. Workspace organization by subject area or project usually works well.
- Separate development and test content from production content so that work-in-progress items can't be accessed by viewers.
- Use apps (or per-item permissions when appropriate) when you expect to have many access requests to process. There isn't a *Request access* workflow for workspaces.

Checklist - When creating your strategy for using workspaces for content viewers, key decisions and actions include:

- ✓ **Decide on a strategy for using the workspace Viewer role**: Define what your preferences are for how to use workspaces for consumers. Ensure that it aligns with your overall strategy for read-only consumers.
- ✓ **Create and publish guidance for content creators**: Provide documentation and training for content creators. Include requirements and preferences for how to use workspace permissions most effectively.

Per-item permissions

Individual item sharing grants permission to a single item. Less experienced content creators commonly use this technique as the primary sharing technique because the sharing commands are prominently displayed in the Power BI service. For this reason, it's important to educate your content creators on the different sharing options, including when to use app permissions instead of workspace roles.

Per-item permissions are a good choice when:

- You want to provide read-only access to one item (report or dashboard).
- You don't want the consumer to view all content published to a workspace.
- You don't want the consumer to view all content published to an app audience.

Use per-item permissions sparingly because sharing grants Read permission to a single item. In a way, you can think of per-item permissions as an override of workspace roles or app permissions.

> **Tip**
>
> We recommend that you use app permissions whenever possible. Next, consider using workspace roles to enable direct workspace access. Lastly, use per-item permissions when they meet the above criteria. App permissions and workspace roles both specify security for a collection of content (rather than individual items), which is a better security practice.

Sharing many items by using per-item permissions can be tedious and error prone, especially when sharing to individual users instead of groups. Consider this scenario: You have 40 reports that you've shared to colleagues by using their individual user accounts. When one colleague transfers to a different department, you'll need to revoke their access, which will involve editing permissions for all 40 reports.

> **Important**
>
> Sharing content from a personal workspace should be done infrequently. Personal workspaces are best suited to non-critical, informal, or temporary content. If you have a situation where content creators frequently share important or critical content from their personal workspaces, you should take appropriate action to move that content to a standard workspace. For more information, see the personal BI usage scenario.

When you share an individual item, you have several permission options.

- **Reshare permission**: When enabled, users can share the item with other users, including its underlying semantic models. It makes sense to grant this permission when the item can be readily shared with anyone. It removes control from the person or team that manages the item. So, it relies on good judgment by users who are granted the Reshare permission. However, your internal governance or security guidelines can become more difficult to manage when resharing is allowed.
- **Build permission**: When enabled, users are granted Build permission for the underlying semantic model. Build permission allows users to create new content that's based on the semantic model. It also allows them to export underlying data from reports, and more. Considerations for granting Build permission are described in the Content creator security planning article.

Per-item permissions for reports and dashboards can make sense for informal scenarios when content is shared with a few users. It's a good idea to educate users on managing

permissions with apps and workspaces instead, especially when they're sharing content to large numbers of users or users outside their team. It's important to emphasize the following points.

- It becomes more difficult to determine which content has been shared with which users, because the permissions on each report and dashboard must be reviewed individually.
- In many instances, Reshare permission is set because the user experience enables this option by default. So, there's a risk that content is shared to a wider set of users than intended. This outcome can be prevented by unchecking the *Allow recipients to share this report* option when sharing. Minimizing oversharing in this way is a user training issue. The content creator that's setting the sharing permissions should consider this choice every time.
- All changes to reports and dashboards are viewable by others immediately, which could confuse users when content modifications are a work in progress. This concern can be mitigated by distributing content in an app, or by using separate workspaces to segregate development, test, and production content. For more information, see the self-service content publishing usage scenario.
- When a user shares content from their personal workspace and they leave the organization, IT usually disables their user account. In this case, all recipients of the shared content will immediately lose access to the content.

There are three specific types of sharing: sharing links, direct access sharing, and shared views.

Per-item permission links

When you share an individual item, the default experience results in a sharing link. There are three types of sharing links.

- **People in your organization**: When enabled in your Fabric tenant settings, this type of sharing link is a straightforward way to provide read-only access to everyone within the organization. However, the sharing link won't work for external users. This option is best suited to when anyone can view the content, and the link can be freely shared throughout the organization. Unless it's disabled by the *Allow shareable links to grant access to everyone in your organization* tenant setting, this type of sharing is the default.
- **People with existing access**: This option doesn't create a new sharing link. Rather, it allows you to retrieve the URL so you can send it to someone who already has access.
- **Specific people**: This option produces a sharing link for specific users or groups. We recommend that you use this option most of the time because it provides

specific access. If you commonly work with external users, you can use this type of link for guest users who already exist in Microsoft Entra ID. For more information about the planned invitation process to create guest users, see the Tenant-level security planning article.

> **Important**
>
> We recommend that you consider restricting the ***Allow shareable links to grant access to everyone in your organization*** tenant setting to members of a group. You can create a group name like *Power BI Share to Entire Organization*, and then add a small number of users who understand the implications of organization-wide sharing. If you're concerned about existing organization-wide links, you can use the **admin API** to find all items that have been shared with the entire organization.

A sharing link adds Read permission to the item. The Reshare permission is selected by default. It's also possible to add Build permission to the underlying semantic model at the same time that the sharing link is created.

> **Tip**
>
> We recommend that you teach content creators to enable the Build permission option *only* when the consumer of the report is also a content creator who might need to create reports, export data, or create a composite model from the underlying semantic model.

Sharing links are easier to maintain than direct access sharing, particularly when you need to do bulk changes. It's a significant advantage when individual users are granted sharing permissions, more so than groups (which commonly occurs when self-service users are responsible for managing permissions). Consider the following comparisons.

- **Sharing link**: 20 individual users are granted access with a sharing link. With a single change to the link, it affects all 20 users.
- **Direct access**: 20 individuals are granted direct access to an item. To make a change, all 20 user permissions must be modified.

Per-item direct access permissions

You can also achieve per-item permissions by using *direct access*. Direct access involves setting up the permissions for a single item. You can also determine any inherited permissions derived from workspace roles.

When you grant a user direct access, they're granted Read permission for the item. The Reshare permission is selected by default, as is the Build permission for the underlying semantic model. We recommend that you teach content creators to enable Build permission only when the consumer of this report is also a content creator who might need to create reports, export data, or create composite models from the underlying semantic model.

> **Tip**
>
> The user experience makes granting Reshare and Build permissions very straightforward, but the user doing the sharing should always verify whether those permissions are necessary.

Shared views

Use a shared view to share a filtered perspective of a report with another user. You can publish a shared view by using a sharing link or by direct access.

Shared views are a temporary concept. They automatically expire after 180 days. For this reason, shared views are best suited to informal and temporary sharing scenarios. Be sure your users are aware of this limitation.

Checklist - When creating your strategy for using per-item permissions, key decisions and actions include:

- ✓ **Decide on the strategy for use of the sharing feature**: Define what your preferences are for how to use per-item permissions. Ensure that it aligns with your overall strategy for read-only consumers.
- ✓ **Decide who can publish links to the entire organization**: Decide which report creators are able to publish links for the entire organization. Set the *Allow shareable links to grant access to everyone in your organization* tenant setting to align with this decision.
- ✓ **Create and publish guidance for content creators**: Provide documentation and training for content creators that includes requirements and preferences for how to use per-item permissions most effectively. Ensure they're clear on the advantages and disadvantages of per-item permissions. Include guidance for when to use sharing links and when to use direct access sharing.

Other consumer query techniques

The most common ways for consumers to interact with Power BI are with apps, workspaces, and per-item permissions (previously described in this article).

There are other techniques that consumers can use to query Power BI data. Each of the following query techniques requires semantic model or datamart Build permission.

- **Analyze in Excel**: Consumers who prefer to use Excel can query a Power BI semantic model by using Analyze in Excel. This capability is a great alternative to exporting data to Excel because the data isn't duplicated. With a live connection to the semantic model, users can create PivotTables, charts, and slicers. They can then publish the workbook to a workspace in the Power BI service which allows consumers to open it and interact with it.
- **XMLA endpoint**: Consumers can query a semantic model by connecting to the XMLA endpoint. An application that's XMLA-compliant can connect to, query, and consume a semantic model that's stored in a Premium workspace. This capability is helpful when consumers want to use a Power BI semantic model as their data source for a data visualization tool outside of the Microsoft ecosystem.
- **Datamart editor**: Consumers can query a Power BI datamart by using the datamart editor. It's a web-based visual query editor for creating no-code queries. There's also a web-based SQL editor for when consumers prefer to write SQL queries. Both editors query the managed Azure SQL Database that underlies the Power BI datamart (rather than the built-in semantic model).
- **SQL endpoint**: Consumers can query a Power BI datamart by using the SQL endpoint. They can use tools like Azure Data Studio or SQL Server Management Studio (SSMS) to run SQL queries. The SQL endpoint directs queries to the managed Azure SQL Database that underlies the Power BI datamart (rather than the built-in semantic model).

For more information about the Build permission, see the Content creator security planning article.

Checklist - When planning which query techniques consumers will use, key decisions and actions include:

- ✓ **Create guidance for users on using Analyze in Excel**: Provide documentation and training for consumers on the best way to reuse existing semantic models with

Excel.

- ✓ **Create guidance for users on using the XMLA endpoint**: Provide documentation and training for consumers on the best way to reuse existing semantic models with the XMLA endpoint.
- ✓ **Create guidance for users on datamart queries**: Provide documentation and training for consumers on the available techniques for querying Power BI datamarts.

Request access workflow for consumers

When sharing content, it's common that one user forwards a link (URL) to another user. When the recipient tries to view the content, and discovers that they don't have access to it, they can select the *Request access* button. This action initiates the Request access workflow. The user is then asked to provide a message to explain why they want access.

A *Request access* workflow exists for:

- Access to a Power BI app.
- Access to an item, like a report or dashboard.
- Access to a semantic model. For more information about the *Request access* workflow when a semantic model is discoverable, see the Content creator security planning article.

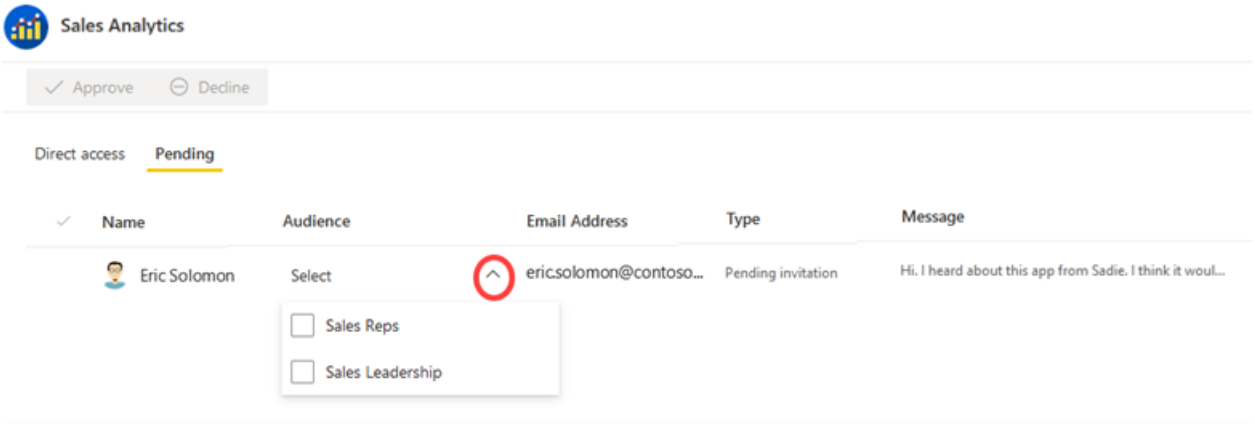
App access requests

There are two ways to learn about pending access requests that have been submitted for an app.

- **Email**: The contact(s) for the app receive an email notification. By default, this contact is the app publisher. To provide better support for critical apps, we recommend that you set the contact to a group that's able to respond quickly to access requests.
- **Manage permissions menu**: Workspace administrators and members can view, approve, or decline access requests. The *Manage permissions* page is available on the Apps page, and can be opened for each app. This capability is also available to workspace contributors when the *Allow contributors to update the app for this workspace* setting is enabled.

Pending access requests for an app show the message provided by the user. Each pending request can be approved or declined. When choosing to approve a request, an app audience must be selected.

The following screenshot shows a pending access request from a user. To approve it, one of the two app audiences, *Sales Reps* or *Sales Leadership*, must be selected.



When you publish an app, the content and the permissions are published at the same time. As previously described, it's not possible to publish only the app permissions without content changes too. However, there's one exception: When you approve a pending access request (such as the one shown in the previous screenshot), the permission change occurs without publishing the latest content in the workspace.

Workspace access requests

Workspace access is granted by users who belong to the Admin role or Member role.

A user who is attempting to view a workspace receives an *access denied* message when they aren't a member of a workspace role. Since there isn't a built-in *Request access* workflow for workspaces, they're best used for small teams and informal teams who work closely together. That's one reason why a Power BI app is better suited to larger teams and broader content distribution scenarios.

Per-item access requests

There are two ways to learn about pending access requests that have been submitted for an individual item, like a report.

- **Email**: The contact(s) for the item receive an email notification. To provide better support for critical reports, we recommend that you set the contact to a group that's able to respond quickly to access requests.
- **Manage permissions menu**: Workspace administrators and members can access the *Manage permissions* page for each item. They can view, approve, or decline access pending requests.

Manage access requests with groups

When a user submits the *Request access* form for a Power BI item (like a report or semantic model) or a Power BI app, the request is submitted for an individual user. However, many large organizations need to use groups to comply with their internal security policies.

We recommend that you use groups, rather than individuals, for securing content whenever practical. For more information about planning for groups, see the Tenant-level security planning article.

If you intend to provide access to groups instead of individual users, the content owner or administrator that's processing the request for access will need to complete the request in multiple steps:

1. Decline the pending request in Power BI (because it's associated with an individual user).
2. Add the requestor to the correct group according to your current process.
3. Notify the requestor that they now have access.

> **Tip**
>
> See **Request access workflow for creators** for information about responding to requests for Build access from content creators. It also includes recommendations about using a form for access requests.

Checklist - When planning the *Request access* workflow, key decisions and actions include:

- ✓ **Determine who should handle app access requests**: Ensure that a process is in place to handle app access requests in a timely manner. Ensure that app contacts are assigned to support the process.
- ✓ **Determine who should handle per-item requests**: Ensure that a process is in place to handle access requests in a timely manner. Ensure that contacts are assigned to each item to support the process.
- ✓ **Include in documentation and training for content creators**: Ensure that content creators understand how to handle access requests in a timely manner. Make them aware of how to handle requests when a group should be used instead of an individual user.

✓ **Include in documentation and training**: Include guidance for your content creators on how to manage access requests effectively. Also include guidance for consumers on what information to include in their access request message.

Enforce data security based on consumer identity

You can plan to create fewer semantic models and reports by enforcing data security. The objective is to enforce data security based on the identity of the user who's viewing the content.

For example, consider that you can share a single sales report with all salespeople (consumers), knowing that each salesperson will only see sales results for their region. This approach allows you to avoid the complexity of creating separate reports *per region* that would need to be shared with the salespeople from that sales region.

Some organizations have specific requirements for endorsed (certified or promoted) semantic models or datamarts. For data that will be widely used, there might be a requirement to use data security.

You can accomplish data security in multiple ways.

- **Power BI semantic model**: As a Power BI data creator, you can enforce row-level security (RLS) and object-level security (OLS). RLS involves defining roles and rules that filter data model rows, while OLS restricts access to specific tables or columns. These defined RLS and OLS rules do not apply to references stored outside the semantic model such as slicer and filter selections. Both RLS and OLS techniques are further described later in this section.
- **Analysis Services**: A live connection semantic model can connect to a remote data model, which is hosted by either Azure Analysis Services (AAS) or SQL Server Analysis Services (SSAS). The remote model can enforce RLS or OLS based on the consumer identity.
- **Data source**: Some data sources, like Azure SQL Database, can enforce RLS. In this case, the Power BI model can take advantage of the existing security rather than redefining it. That approach can be a significant advantage when RLS defined in the source is complex. You can develop and publish a DirectQuery model and set the data source credentials of the semantic model in the Power BI service to enable single sign-on (SSO). When a report consumer opens a report, Power BI passes their identity to the data source. The data source then enforces RLS based on the identity of the report consumer. For more information about Azure SQL Database RLS, see this article.

> **Note**
>
> Source systems, like Azure SQL Database, can also use techniques like views to narrow down what the user can see. While that's a valid technique, it's not relevant to the focus of this section.

Row-level security

Row-level security (RLS) allows a data modeler to restrict access to a subset of data. It's typically used to ensure that some report consumers can't see specific data, like sales results of other sales regions.

> **Tip**
>
> If you've noticed someone creating multiple data models to support different groups of consumers, check whether RLS will satisfy their requirements. It's typically better to create, test, and maintain one data model rather than multiple data models.

Take care, because if a Power BI report references a row with RLS configured then the same message will be displayed as for a deleted or non-existing field. To these users, it looks like the report is broken.

There are two steps for setting up RLS: rules and role mappings.

RLS rules

For semantic models, a data modeler can set up RLS in Power BI Desktop by creating one or more roles. A role has a unique name in the model, and it usually includes one or more rules. Rules enforce filters on model tables by using Data Analysis Expressions (DAX) filter expressions. By default, a model has no roles.

> **Important**
>
> A model without roles means that users (who have permission to query the data model) have access to all model data.

Rule expressions are evaluated within row context. Row context means the expression is evaluated for each row by using the column values of that row. When the expression

returns `TRUE`, the user can see the row. You can define rules that are either *static* or *dynamic*.

- **Static rules**: Use DAX expressions that refer to constants, like `[Region] = "Midwest"`.
- **Dynamic rules**: Use specific DAX functions that return environmental values (as opposed to constants). Environmental values are returned from three specific DAX functions: USERNAME, USERPRINCIPALNAME, and CUSTOMDATA. Defining dynamic rules is simple and effective when a model table stores username values. They allow you to enforce a *data-driven* RLS design.

RLS role mappings

After you publish the model to the Power BI service, you must set up role mappings in advance of users accessing related reports. Role mapping involves assigning Microsoft Entra security objects to roles. Security objects can be user accounts or security groups.

Whenever possible, it's a best practice to map roles to security groups. That way, there will be fewer mappings, and group membership management can be handled by the owner of the group.

We recommend that you make security account information from Microsoft Entra available to your content creators. One option is to create a dataflow with data that's kept in sync with Microsoft Entra ID. That way, content creators can integrate the dataflow data to produce a data-driven semantic model.

> **Tip**
>
> It's possible to define a role that has no rules. In this case, the role provides access to all rows of all model tables. Setting up this type of role is suitable when an administrator or user is allowed to view all data in the model.

RLS user experience

Some organizations choose to purposefully use RLS as a secondary layer of security, in addition to standard Power BI permissions. Consider the following scenario: You share a link to a report with the entire organization. Any user who views the report must be mapped to an RLS role to be able to see data in the report. If they aren't mapped to an RLS role, they won't see any data.

The presence of RLS changes the default experience for consumers.

- **When RLS isn't defined for the semantic model**: Creators and consumers with *at least* Read permission on the semantic model can view *all data* in the semantic model.
- **When RLS is defined on the semantic model**: Creators and consumers with *only* Read permission on the semantic model will only be able to view the data they're allowed to see (based on their RLS role mapping).

> **Note**
>
> Some organizations enforce RLS as an additional layer of security, especially when sensitive data is involved. For this reason, you might choose to require RLS for semantic models that are **certified**. That requirement can be accomplished with an internal review and approval process prior to certifying the semantic model.

When a user views a report in either a workspace or an app, RLS might or might not be enforced depending on their semantic model permissions. For this reason, it's critical that content consumers and creators *only* possess Read permission on the underlying semantic model when RLS must be enforced.

Here are the permission rules that determine whether RLS is enforced.

- **User has Read permission on the semantic model**: RLS is enforced for the user.
- **User has Read and Build permissions on the semantic model**: RLS is enforced for the user.
- **User has Write permission on the semantic model**: RLS isn't enforced for the user, meaning that they can see all data in the semantic model. The Write permission provides the ability to edit a semantic model. It can be granted in one of two ways:
  - With the Contributor, Member, or Admin workspace roles (for the workspace where the semantic model is stored).
  - With the per-item Write semantic model permission.

> **Tip**
>
> For more information about how to use separate workspaces so that RLS works for content creators, see the **managed self-service BI** usage scenario.

For more information about RLS, see Restrict access to Power BI model data.

RLS for datamarts

Power BI datamarts can also enforce RLS. However, the implementation is different.

The main difference is that RLS for datamarts is set up in the Power BI service, rather than in Power BI Desktop.

Another difference is that datamarts enforce RLS on both the semantic model and the managed Azure SQL Database that's associated with the datamart. Enforcing RLS at both layers provides consistency and flexibility. The same RLS filters are applied regardless of how the user queries the data, whether it's by connecting to the semantic model or to the managed Azure SQL Database.

For more information, see RLS for datamarts.

Object-level security

Object-level security (OLS) allows a data modeler to restrict access to specific tables and columns, and their metadata. You typically use OLS to ensure sensitive columns, like employee salary, aren't visible to certain users. While it isn't possible to restrict access to measures, any measure that references a restricted column will itself be restricted.

Consider an example of an employee table. It contains columns that store the employee name and phone number, and also salary. You can use OLS to ensure that only certain users, like senior Human Resources staff, can see salary values. For those users that can't see salary values, it's as if that column doesn't exist.

Take care, because if a Power BI report visual includes salary, users that don't have access to that field will receive an error message. The message will inform them that the object doesn't exist. To these users, it looks like the report is broken.

> **Note**
>
> You can also define **perspectives** in a data model. A perspective defines viewable subsets of model objects to help provide a specific focus for report creators. Perspectives aren't intended to restrict access to model objects. A user can still query a table or column even when it's not visible to them. Therefore, consider perspectives as a user convenience rather than a security feature.

There isn't currently an interface in Power BI Desktop to set up OLS. You can use Tabular Editor, which is a third-party tool for creating, maintaining, and managing models. For more information, see the advanced data model management usage scenario.

For more information about OLS, see Restrict access to Power BI model objects.

Checklist - When planning for RLS and OLS, key decisions and actions include:

- ✓ **Decide on the strategy for use of RLS**: Consider for which use cases and purposes you intend to use row-level security.
- ✓ **Decide on the strategy for use of OLS**: Consider for which use cases and purposes you intend to use object-level security.
- ✓ **Consider requirements for certified content**: If you have a process for what's required to certify a semantic model, decide whether to include any specific requirements for using RLS or OLS.
- ✓ **Create and publish user guidance**: Create documentation for users that includes requirements and preferences for using RLS and OLS. Describe how to obtain user mapping information if it exists in a centralized location.
- ✓ **Update training materials**: Include key information about requirements and preferences for RLS and OLS in user training materials. Provide examples for users to understand when it's appropriate to use either data security technique.

Related content

In the next article in this series, learn about security planning for content creators who are responsible for creating semantic models, dataflows, datamarts, reports, or dashboards.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Content creator security planning

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This security planning article describes strategies for content creators who are responsible for creating semantic models, dataflows, datamarts, reports, or dashboards. It's primarily targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.
- **Content creators and owners**: Self-service BI creators who need to create, publish, secure, and manage content that others consume.

The series of articles is intended to expand upon the content in the Power BI security white paper. While the Power BI security white paper focuses on key technical topics such as authentication, data residency, and network isolation, the primary goal of the series is to provide you with considerations and decisions to help you plan for security and privacy.

In an organization, many users are *content creators*. Content creators produce and publish content that's viewed by others. Content creators are the focus of this article.

> **Tip**
>
> We recommend that you review the Report consumer security planning article first. It describes strategies for securely providing content to read-only consumers including how to enforce data security.

Strategy for content creators

The foundation of a well-governed self-service BI system begins with content creators and owners. They create and validate semantic models and reports. In many cases, content creators also set up permissions to manage security for their content.

> **Tip**
>
> We recommend that you foster a **data culture** that makes security and protection of data a normal part of everyone's role. To achieve that objective, user education, support, and training is essential.

For purposes of security and permissions, consider that there are two types of content creators: data creators and report creators. They can be responsible for creating and managing enterprise BI or self-service BI content.

Data creators

A data creator is any Power BI user who creates semantic models, dataflows, or datamarts.

Here are some common data creator scenarios.

- **Create a new semantic model**: Create and test a new data model in Power BI Desktop. It's then published to the Power BI service so that it can be used as a shared semantic model for many reports. For more information about reusing shared semantic models, see the managed self-service BI usage scenario.
- **Extend and customize a semantic model**: Create a live connection to an existing shared semantic model in Power BI Desktop. Convert the live connection to a local model, which allows extending the model design with new tables or columns. For more information about extending and customizing shared semantic models, see the customizable managed self-service BI usage scenario.
- **Create a new dataflow**: In the Power BI service, create a new dataflow so that it can be used as a source by many semantic models. For more information about reusing data preparation activities, see the self-service data preparation usage scenario.
- **Create a new datamart**. In the Power BI service, create a new datamart.

Data creators are often found in enterprise BI teams and in the Center of Excellence (COE). They also have a key role to play in decentralized business units and departments that maintain and manage their own data.

For other considerations about business-led BI, managed self-service BI, and enterprise BI, see the Content ownership and management article.

Report creators

Report creators create reports and dashboards to visualize data that's sourced from existing semantic models.

Here are some common report creator scenarios.

- **Create a new report including a data model**: Create and test a new report and data model in Power BI Desktop. The Power BI Desktop file that contains one or more report pages and includes a data model is published to the Power BI service. New content creators commonly use this method before they're aware of using shared semantic models. It's also appropriate for narrow use cases that have no need for data reuse.
- **Create a live connection report**: Create a new Power BI report that connects to a shared semantic model in the Power BI service. For more information about reusing shared semantic models, see the managed self-service BI usage scenario.
- **Create a connected Excel workbook**: Create a new Excel report that connects to a shared semantic model in the Power BI service. Connected Excel experiences, rather than downloads of data, are highly encouraged.
- **Create a DirectQuery report**: Create a new Power BI report that connects to a supported data source in DirectQuery mode. One situation when this method is useful is when you want to take advantage of user security that's implemented by the source system.

Report creators are found throughout every business unit in the organization. There are usually many more report creators in an organization than data creators.

> **Tip**
>
> While not every semantic model is a shared semantic model, it's still worth adopting a **managed self-service BI** strategy. This strategy reuses shared semantic models whenever possible. In that way, report creation and data creation are *decoupled*. Any content creator from any business unit can effectively use this strategy.

Permissions for creators

This section describes the most common permissions for data creators and report creators.

This section isn't intended to be an all-inclusive list of every possible permission. Rather, it's intended to help with planning your strategy for supporting different types of content creators. Your goal should be to follow the principle of least privilege. This principle allows enough permissions for users to be productive, without over-provisioning permissions.

Creating new content

The following permissions are commonly required for creating new content.

Expand table

Permission	Report creator	Semantic model creator	Dataflow creator	Datamart creator
Access to the underlying data source		✓	✓	✓
Semantic model Read and Build permissions	✓			
Dataflow Read permission (when a dataflow is used as a source, via the workspace Viewer role)		✓	✓	✓
Access where original Power BI Desktop file is stored	✓	✓		
Permission to use custom visuals	✓			

Publishing content permissions

The following permissions are commonly required for publishing content.

Expand table

Permission	Report creator	Semantic model creator	Dataflow creator	Datamart creator
Workspace role: Contributor, Member, or Admin	✓	✓	✓	✓
Semantic model Write permission (when the user doesn't belong to a workspace role)		✓		

Permission	Report creator	Semantic model creator	Dataflow creator	Datamart creator
Deployment pipeline role to publish items (optional)	✅	✅	✅	✅

Refreshing data

The following permissions are commonly required for refreshing data.

Expand table

Permission	Report creator	Semantic model creator	Dataflow creator	Datamart creator
Owner assigned (who has set up settings or taken over the item)		✅	✅	✅
Access to the underlying data source (when a gateway isn't used)		✅	✅	✅
Access to the data source in a gateway (when the source is on-premises or in a virtual network)		✅	✅	✅

The remainder of this article describes considerations for content creator permissions.

> **Tip**
>
> For permissions related to viewing content, see the Report consumer security planning article.

Checklist - When planning your security strategy for content creators, key decisions and actions include:

- ✓ **Determine who your data creators are**: Ensure that you're familiar with who's creating semantic models, dataflows, and datamarts. Verify that you understand what their needs are before starting your security planning activities.

- ✓ **Determine who your report creators are**: Ensure that you're familiar with who's creating reports, dashboards, workbooks, and scorecards. Verify that you understand what their needs are before starting your security planning activities.

Discover content for creators

Users can rely on *data discovery* to find semantic models and datamarts. Data discovery is a Power BI feature that allows content creators to locate existing data assets—even when they don't have any permissions for that content.

Discovery of existing data is useful for:

- Report creators who want to use an existing semantic model for a new report.
- Report creators who want to query data from an existing datamart.
- Semantic model creators who want to use an existing semantic model for a new composite model.

> ⓘ **Note**
>
> Data discovery in Power BI isn't a data security permission. It's a setting that allows report creators to read metadata, helping them to discover data and request access to it.

You can set up a semantic model or datamart as discoverable when it's been endorsed (certified or promoted). When it's discoverable, content creators can find it in the OneLake catalog.

A content creator can also request access to the semantic model or datamart. Essentially, an access request asks for Build permission, which is required to create new content based on it. When responding to requests for access, consider using groups instead of individual users. For more information about how to use groups for this purpose, see Request access workflow for consumers.

Consider the following three examples.

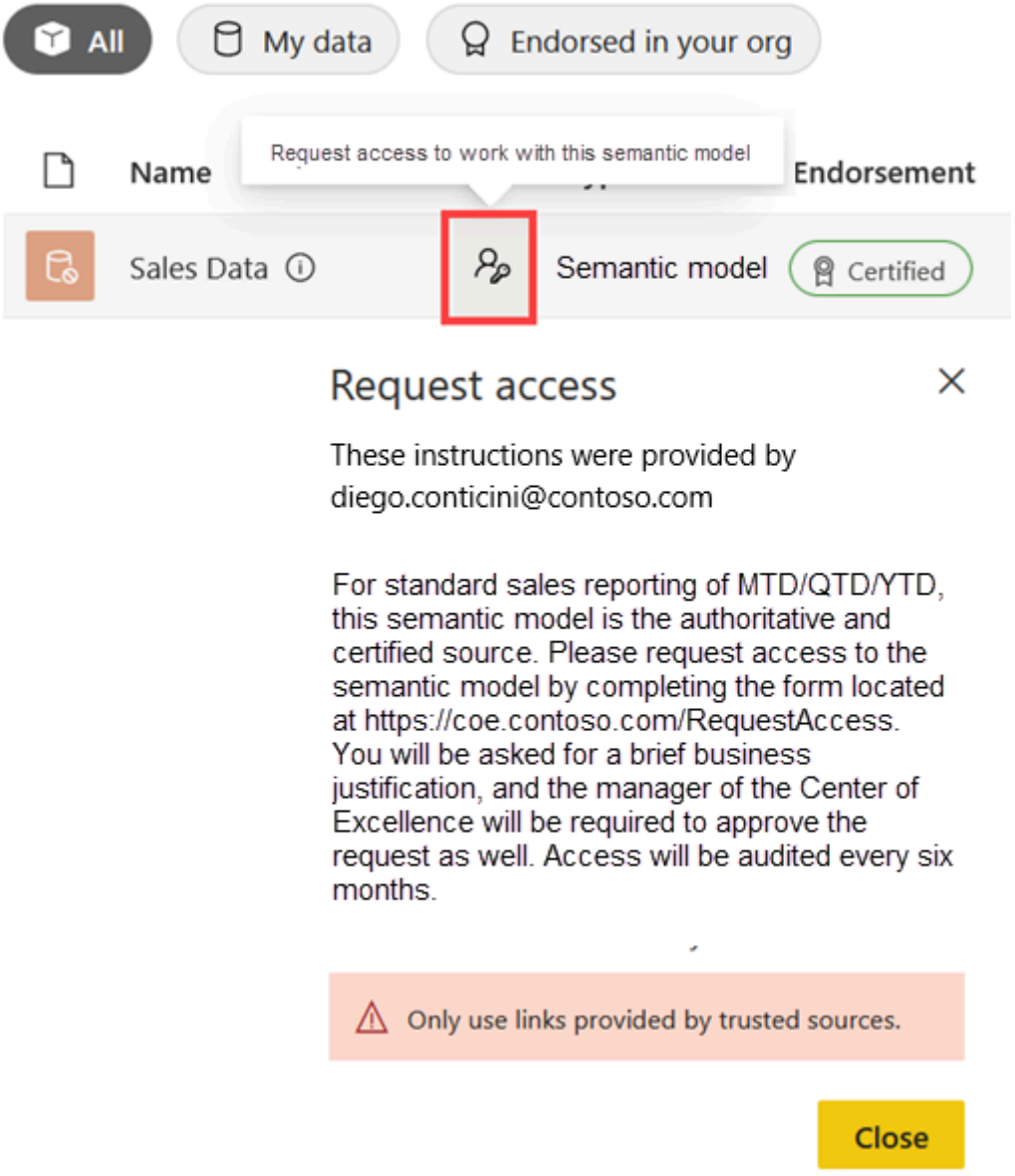
- The **Sales Summary** semantic model is certified. It's the trusted and authoritative source for sales tracking. Many self-service report creators throughout the organization use this semantic model. So, there are many existing reports and composite models based on the semantic model. To encourage other creators to find and use the semantic model, it's set as discoverable.
- The **Inventory Stats** semantic model is certified. It's a trusted and authoritative source for inventory analysis. The semantic model and related reports are

maintained and distributed by the enterprise BI team. Due to the complex design of the semantic model, only the enterprise BI team is allowed to create and maintain inventory content. Since the goal is to discourage report creators from using the semantic model, it isn't set as discoverable.

- The **Executive Bonuses** semantic model contains highly confidential information. Permissions to view or update this semantic model are restricted to a few users. This semantic model isn't set as discoverable.

The following screenshot shows a semantic model in the OneLake catalog in the Fabric portal. Specifically, it shows an example of a *Request access* message for a discoverable semantic model. This message is shown when the user doesn't currently have access. The *Request access* message has been customized in the semantic model settings.

The *Request access* message reads: *For standard sales reporting of MTD/QTD/YTD, this semantic model is the authoritative and certified source. Please request access to the semantic model by completing the form located at* `https://COE.contoso.com/RequestAccess`. *You will be asked for a brief business justification, and the manager of the Center of Excellence will be required to approve the request as well. Access will be audited every six months.*



> **Note**
>
> Your data culture and your stance on data democratization should strongly influence whether you enable data discovery. For more information about data discovery, see the customizable managed self-service BI usage scenario.

There are three tenant settings related to discovery.

- The *Discover content* tenant setting allows Power BI administrators to set which groups of users are allowed to discover data. It's primarily targeted at report creators who might need to locate existing semantic models when creating reports. It's also useful for semantic model creators who might look for existing data that they can use in their composite model development. While it's possible to set it for specific security groups, it's a good idea to enable the setting for the entire organization. The discovery setting on individual semantic models and

dataflows will control what's discoverable. Less commonly, you might consider restricting this capability only to approved content creators.

- The *Make certified content discoverable* tenant setting allows Power BI administrators to set which groups can set content to be discoverable (when they also have permission to edit the item as well as permission to certify content, which is granted by the *Certification* tenant setting). The ability to certify content should be tightly controlled. In most cases, the same users who are allowed to certify content should be allowed to set it as discoverable. In some situations, you might want to restrict this capability only to approved data creators.
- The *Make promoted content discoverable* tenant setting allows Power BI administrators to set which groups can set the content as discoverable (when they also have permissions to edit the data). Because the ability to promote content is open to all content creators, in most cases, this capability should be available to all users. Less commonly, you might consider restricting this capability to only approved content creators.

Checklist - When planning data discovery for your content creators, key decisions and actions include:

- ✓ **Clarify needs for data discovery**: Consider what your organization's position is on encouraging content creators to find existing semantic models and datamarts. When appropriate, create a governance policy about how data discovery should be used.
- ✓ **Decide who can discover content**: Decide whether any Power BI user is allowed to discover content, or whether discovery should be limited to certain groups of users (for example, a group of approved content creators). Set the *Discover content* tenant setting to align with this decision.
- ✓ **Decide who can set certified content to be discoverable**: Decide whether any Power BI user (who has permission to edit the semantic model or datamart, as well as permission to certify it) can set it as discoverable. Set the *Make certified content discoverable* tenant setting to align with this decision.
- ✓ **Decide who can set promoted content to be discoverable**: Decide whether any Power BI user (who has permission to edit the semantic model or datamart) can set it as discoverable. Set the *Make promoted content discoverable* tenant setting to align with this decision.
- ✓ **Include in documentation and training for semantic model creators**: Include guidance for your semantic model creators about when it's appropriate to use data

discovery for the semantic models and datamarts they own and manage.

✓ **Include in documentation and training for report creators**: Include guidance for your content creators about how data discovery works, and what they can expect.

Request access workflow for creators

A user can request access to content in two ways.

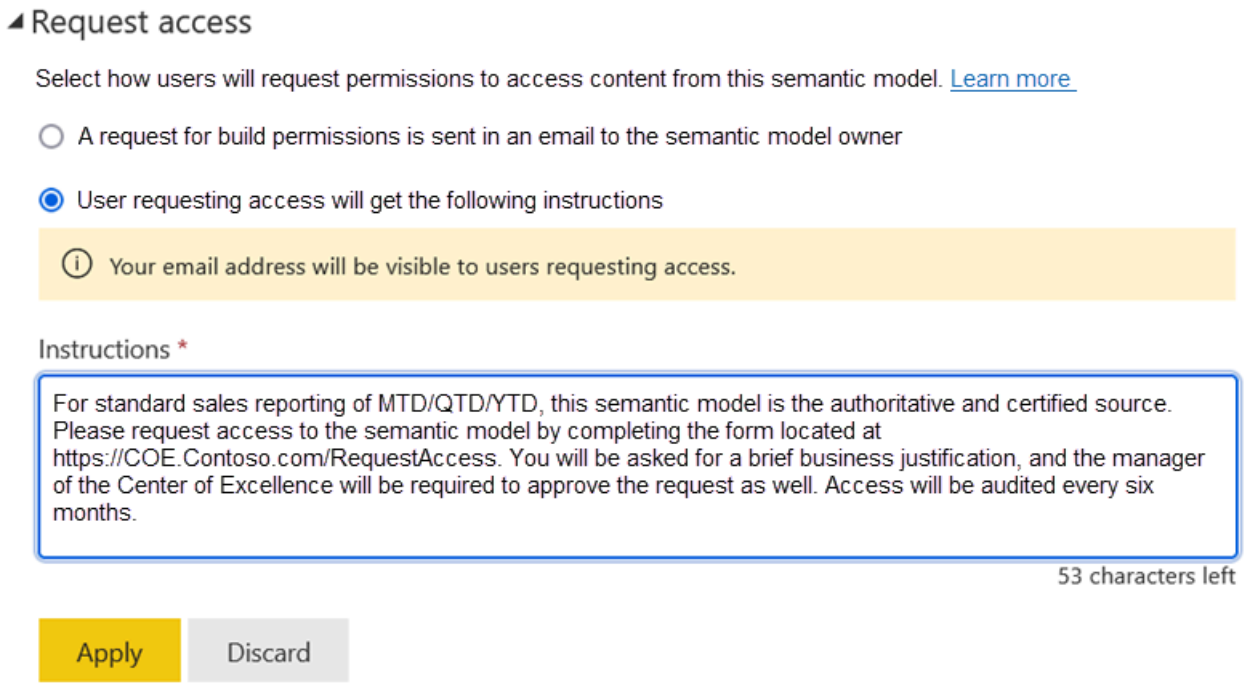
- **For content consumers**: A user receives a link to an existing report or app in the Power BI service. To view the item, the consumer can select the *Request access* button. For more information, see the Report consumer security planning article.
- **For content creators**: The user discovers a semantic model or datamart in the OneLake catalog. To create a new report or composite model based on the existing data, the content creator can select the *Request access* button. This experience is the focus of this section.

By default, a request for access to a semantic model or a datamart goes to the owner. The owner is the user who last scheduled data refresh or input credentials. Relying on one user to process access requests might be acceptable for team semantic models. However, that might not be practical or reliable.

Instead of relying on one owner, you can define custom instructions that are presented to users when they request access to a semantic model or datamart. Custom instructions are helpful when:

- The semantic model is set as discoverable.
- Approval of the access request will be done by someone other than the data owner.
- There's an existing process in place that needs to be followed for access requests.
- Tracking of who requested access, when, and why is necessary for auditing or compliance reasons.
- Explanation is necessary for how to request access, and to set expectations.

The following screenshot shows an example of setting up custom instructions that a user sees when they request the Build permission. The custom instructions read: *For standard sales reporting of MTD/QTD/YTD, this semantic model is the authoritative and certified source. Please request access to the semantic model by completing the form located at* `https://COE.contoso.com/RequestAccess`*. You will be asked for a brief business justification, and the manager of the Center of Excellence will be required to approve the request as well. Access will be audited every six months.*



There are many options to create a form. Power Apps and Microsoft Forms are both low-code, easy-to-use options. We recommend that you create a form in a way that's independent of a single user. It's crucial that your form is created, managed, and monitored by the proper team.

We recommend that you create helpful information for:

- Content creators so they know what to expect when they request access.
- Content owners and administrators so they know how to manage requests that are submitted.

> **Tip**
>
> For more information about responding to requests for read access from consumers, see **Request access workflow for consumers**. It also includes information about using groups (instead of individual users).

Checklist - When planning the *Request access* workflow, key decisions and actions include:

- ✓ **Clarify preferences for how to handle access requests**: Determine for which situations it's acceptable for owner approval, and when a different process should

be used. When appropriate, create a governance policy about how access requests should be handled.

- ✓ **Include in documentation and training for semantic model and datamart creators**: Include guidance for your semantic model and datamart creators about how and when to set custom instructions for access requests.
- ✓ **Include in documentation and training for report creators**: Include guidance for your report creators about what they can expect when requesting Build permissions for semantic models and datamarts.

Create and publish content

This section includes security aspects that apply to content creators.

> ⓘ Note
>
> For consumers who view reports, dashboards, and scorecards, see the Report consumer security planning article. Considerations related to app permissions are covered in that article, too.

Workspace roles

You grant workspace access by adding users or groups (including security groups, Microsoft 365 groups, and distribution lists) to workspace roles. Assigning users to workspace roles allows you to specify what they can do with the workspace and its content.

> ⓘ Note
>
> For more information about workspace planning considerations, see the workspace planning articles. For more information about groups, see the Tenant-level security planning article.

Because the primary purpose of a workspace is collaboration, workspace access is mostly relevant for the users who own and manage its content. When starting to plan for workspace roles, it's helpful to ask yourself the following questions.

- What are the expectations for how collaboration will occur in the workspace?
- Who will be responsible for managing the content in the workspace?
- Is the intention to assign individual users or groups to workspace roles?

There are four Power BI workspace roles: Admin, Member, Contributor, and Viewer. The first three roles are relevant to content creators, who create and publish content. The Viewer role is relevant to read-only consumers.

The four workspace role permissions are nested. That means that workspace administrators have all the capabilities available to members, contributors, and viewers. Likewise, members have all the capabilities available to contributors and viewers. Contributors have all the capabilities available to viewers.

> **Tip**
>
> See the **workspace roles** documentation for the authoritative reference for each of the four roles.

Workspace administrator

Users assigned to the Admin role become the workspace administrators. They can manage all settings and perform all actions, including adding or removing users (including other workspace administrators).

Workspace administrators can update or delete the Power BI app (if one exists). They can, optionally, allow contributors to update the app for the workspace. For more information, see Variations to workspace roles later in this article.

> **Tip**
>
> When referring to an administrator, be sure to clarify whether you're speaking about a workspace administrator or a Power BI tenant-level administrator.

Take care to ensure that only trusted and reliable individuals are workspace administrators. A workspace administrator has high privileges. They have access to view and manage all content in the workspace. They can add and remove users (including other administrators) to any workspace role. They can also delete the workspace.

We recommend that there are at least two administrators so that one serves as a backup should the primary administrator be unavailable. A workspace that doesn't have an administrator is known as an *orphaned workspace*. The orphaned status occurs when a user leaves the organization and there's no alternative administrator assigned to the workspace. For more information about how to detect and rectify orphaned workspaces, see Manage workspaces.

Ideally, you should be able to determine who's responsible for the workspace content by who the workspace administrators and members are (and the contacts specified for the workspace). However, some organizations adopt a content ownership and management strategy that restricts workspace creation to specific users or groups. They typically have an established workspace creation process that is managed by the IT department. In this case, the workspace administrators would be the IT department rather than the users who directly create and publish the content.

Workspace member

Users assigned to the Member role can add other workspace users (but not administrators). They can also manage permissions for all content in the workspace.

Workspace members can publish or unpublish the app for the workspace, share a workspace item or the app, and allow other users to share workspace items of the app.

Workspace members should be limited to the users who need to manage the workspace content creation and publish the app. In some cases, the workspace administrators fulfill that purpose, so you might not need to assign any users or groups to the Member role. When the workspace administrators aren't directly related to the workspace content (for example, because IT manages the workspace creation process), the workspace members might be the true owners responsible for the workspace content.

Workspace contributor

Users assigned to the Contributor role can create, edit, or delete workspace content.

Contributors can't update the Power BI app (when one exists for the workspace) unless that's been allowed by the workspace setting. For more information, see Variations to workspace roles later in this article.

Most content creators in the organization are workspace contributors.

Workspace viewer

Users assigned to the Viewer role can view and interact with all workspace content.

The Viewer role is relevant to read-only consumers for small teams and informal scenarios. It's fully described in the Report consumer security planning article.

Workspace ownership considerations

Consider an example where the following actions are taken to set up a new workspace.

1. Specific Power BI champions and satellite members of the Center of Excellence (COE) have been granted permission in the tenant settings to create new workspaces. They've been trained in content organization strategies and naming standards.
2. You (a content creator) submit a request to create a workspace for a new project that you'll manage. The workspace will include reports that track the progress of your project.
3. A Power BI champion for your business unit receives the request. They determine that a new workspace is justified. They then create a workspace and assign the Power BI champions security group (for their business unit) to the workspace Admin role.
4. The Power BI champion assigns you (the content creator) to the workspace Member role.
5. You assign a trusted colleague to the workspace Member role to ensure there's a backup should you be away.
6. You assign other colleagues to the workspace Contributor role because they'll be responsible for creating the workspace content, including semantic models and reports.
7. You assign your manager to the workspace Viewer role because they've requested access to monitor the progress of the project. Your manager would like to review content in the workspace before you publish an app.
8. You take responsibility for managing other workspace properties such as description and contacts. You also take responsibility for managing workspace access on an ongoing basis.

The previous example shows an effective way to allow a decentralized business unit the ability to act independently. It also shows the principle of least privilege.

For governed content, or critical content that's more tightly managed, it's a best practice to assign groups rather than individual user accounts to workspace roles. That way, you can manage the group membership separately from the workspace. However, when you assign groups to roles, it's possible that users become assigned to multiple workspace roles (because the user belongs to multiple groups). In that case, their effective permissions are based on the highest role that they're assigned to. For more considerations, see Strategy for using groups.

When a workspace is co-owned by multiple individuals or teams, it can make management of the content complicated. Try to avoid multi-team ownership scenarios by separating out workspaces. That way, responsibilities are clear and role assignments are straightforward to set up.

Variations to workspace roles

There are two variations to the four workspace roles (described previously).

- By default, only workspace administrators and members can create, publish, and update the app for the workspace. The *Allow contributors to update the app option for this workspace* setting is a workspace-level setting, which lets workspace administrators delegate the ability to update the app for the workspace to contributors. However, contributors can't publish a new app or change who has permission to edit it. This setting is useful when you want contributors to be able to update the app (when one exists for the workspace), yet not grant the other permissions available to members.
- The *Block republish and disable package refresh* tenant setting only allows semantic model owners to publish updates. When enabled, workspace administrators, members, and contributors can't publish changes unless they first take over the semantic model as its owner. Because this setting applies to the entire organization, enable it with a measure of caution because it affects all semantic models for the tenant. Be sure to communicate to your semantic model creators what to expect because it changes the normal behavior of workspace roles.

> **Important**
>
> Per-item permissions can also be thought of as an override of the standard workspace roles. For more information about per-item permissions, see the **Report consumer security planning** article.

Checklist - When planning for workspace roles, key decisions and actions include:

- ✓ **Create a responsibility matrix**: Map out who is expected to handle each function when creating, maintaining, publishing, securing, and supporting content. Use this information when planning your workspace roles.
- ✓ **Decide on your strategy for assigning workspace roles for content creators**: Determine which users should be an administrator, member, or contributor, and in what circumstances (such as job role or subject area). If there are mismatches that cause a security concern, reconsider how your workspaces could be better organized.

- ✓ **Determine how security groups versus individuals should be used for workspace roles**: Determine the use cases and purposes you'll need to use groups. Be specific about when security should be applied by using user accounts versus when a group is required or preferred.
- ✓ **Provide guidance for content creators about managing workspace roles**: Include documentation for content creators about how to manage workspace roles. Publish this information to your centralized portal and training materials.
- ✓ **Set up and test workspace role assignments**: Verify that content creators have the functionality they need for editing and publishing content.

App creator permissions

Content creators who are workspace administrators or members can create and publish a Power BI app.

A workspace administrator can also specify a setting in the workspace which allows workspace contributors to update the app. It's a variation to workspace role security because it grants contributors one other permission they wouldn't normally have. This setting is set on a per-workspace basis.

> **Tip**
>
> For more information about delivering content to read-only consumers, see the Report consumer security planning article. This article includes information about app permissions for app consumers, including audiences for the app.

Checklist - When planning for app creator permissions, key decisions and actions include:

- ✓ **Decide on your strategy for who can create and publish Power BI apps**: Clarify who should be permitted to create and publish Power BI apps.
- ✓ **Determine when contributors can update Power BI apps**: Clarify the situations when a contributor should be permitted to update Power BI apps. Update the workspace setting when this capability is required.

Data source permissions

When a data creator starts a new project, permissions required to access external data sources are one of their first security-related considerations. They might also need guidance on other data source related matters, including privacy levels, native database queries, and custom connectors.

Access to data source

When a data creator creates a semantic model, dataflow, or datamart, they must authenticate with data sources to retrieve data. Usually, authentication involves user credentials (account and password), which could be for a service account.

Sometimes it's useful to create specific service accounts for accessing data sources. Check with your IT department for guidance on how service accounts should be used in your organization. When they're permitted, the use of service accounts can:

- Centralize permissions needed for data sources.
- Reduce the number of individual users that need permissions to a data source.
- Avoid data refresh failures when a user leaves the organization.

> **Tip**
>
> If you choose to use service accounts, we recommend that you tightly control who has access to the credentials. Rotate passwords on a regular basis (such as every three months) or when someone that has access leaves the organization.

When accessing data sources, apply the principle of least privilege to ensure that users (or service accounts) have permission to read *only* the data they need. They should never have permission to perform data modifications. Database administrators who create these service accounts should inquire about expected queries and workloads and take steps to ensure adequate optimizations (like indexes) and resources are in place.

> **Tip**
>
> If it's difficult to provide direct data source access to self-service data creators, consider using an indirect approach. You can create dataflows in the Power BI service and allow self-service data creators to source data from them. This approach has the added benefits of reducing the query load on the data source and delivering a consistent snapshot of data. For more information, see the **self-service data preparation** and **advanced data preparation** usage scenarios.

The credentials (account and password) can be applied in one of two ways.

- **Power BI Desktop**: Credentials are encrypted and stored locally on the user machine.
- **Power BI service**: Credentials are encrypted and securely stored for either:
  - The semantic model (when a data gateway isn't in use to reach the data source).
  - The gateway data source (when a standard gateway or a virtual network gateway service is in use to reach the data source).

> **Tip**
>
> When you've already entered credentials for a semantic model data source, the Power BI service will automatically bind those credentials to other semantic model data sources when there's an exact match of connection string and database name. Both the Power BI service and Power BI Desktop make it look like you're entering credentials for each data source. However, it can apply the same credentials to matching data sources that have the same owner. In that respect, semantic model credentials are *scoped* to the owner.

Credentials are encrypted and stored separately from the data model in both Power BI Desktop and the Power BI service. This data separation has the following security advantages.

- It facilitates the reuse of credentials for multiple semantic models, dataflows, and datamarts.
- When someone parses the metadata of a semantic model, they can't extract the credentials.
- In Power BI Desktop, another user can't connect to the original data source to refresh data without first applying the credentials.

Some data sources support single-sign on (SSO), which can be set when entering credentials in the Power BI service (for semantic model or gateway data sources). When you enable SSO, Power BI sends the authenticated user's credentials to the data source. This option enables Power BI to honor the security settings that are set up in the data source, such as row-level security. SSO is especially useful when tables in the data model use DirectQuery storage mode.

Privacy levels

Data privacy levels specify isolation levels that define the degree that one data source is isolated from other data sources. When appropriately set, they ensure that Power Query only transmits compatible data between sources. When Power Query can transmit data between data sources, it can result in more efficient queries that reduce the volume of

data that's sent to Power BI. When it can't transmit data between data sources, it can result in slower performance.

There are three privacy levels.

- **Private**: Includes sensitive or confidential data that must be isolated from all other data sources. This level is the most restrictive. Private data source data can't be shared with any other data sources. For example, a human resources database that contains employee salary values should be set to the Private privacy level.
- **Organizational**: Isolated from public data sources but is visible to other organizational data sources. This level is the most common. Organizational data source data can be shared with private data sources or other organizational data sources. Most internal operational databases can be set with the Organizational privacy level.
- **Public**: Non-sensitive data that could be made visible to any data source. This level is the least restrictive. Public data source data can be shared with any other data source. For example, a census report obtained from a government website can be set to the Public privacy level.

When combining queries from different data sources, it's important that you set the correct privacy levels. When privacy levels are set correctly, there's the potential for data from one data source to be transmitted to another data source to efficiently query data.

Consider a scenario where a semantic model creator has two data sources: an Excel workbook and a table in an Azure SQL Database. They want to filter the data in the Azure SQL Database table by using a value sourced from the Excel workbook. The most efficient way for Power Query to generate a SQL statement for the Azure SQL Database is to apply a `WHERE` clause to perform the necessary filtering. However, that SQL Statement will contain a `WHERE` clause predicate with a value sourced from the Excel workbook. If the Excel workbook contains sensitive data, it could represent a security breach because the database administrator could view the SQL statement by using a tracing tool. While less efficient, the alternative is for the Power Query mashup engine to download the entire result set of the database table and perform the filtering itself in the Power BI service. This approach will be less efficient and slow, but secure.

Privacy levels can be set for each data source:

- By data modelers in Power BI Desktop.
- By semantic model owners in the Power BI service (for cloud data sources, which don't require a gateway).
- By gateway data source creators and owners in the Power BI service (for gateway data sources).

> **Important**
>
> The privacy levels that you set in Power BI Desktop aren't transferred to the Power BI service.

There's a Power BI Desktop security option that allows ignoring privacy levels to improve performance. You might use this option to improve query performance while developing a data model when there's no risk of breaching data security (because you're working with development or test data that isn't sensitive). However, this setting isn't honored by the Power BI service.

For more information, see Power BI Desktop privacy levels.

Native database queries

To create efficient Power Query queries, you can use a *native query* to access data. A native query is a statement written in a language supported by the data source. Native queries are only supported by specific data sources, which are typically relational databases like Azure SQL Database.

Native queries can pose a security risk because they could run a malicious SQL statement. A malicious statement could perform data modifications or delete database records (when the user has the required permissions in the data source). For this reason, by default, native queries require user approval to run in Power BI Desktop.

There's a Power BI Desktop security option that allows you to disable the requirement for pre-approval. We recommend that you leave the default setting that requires user approval, especially when you anticipate that the Power BI Desktop file could be refreshed by other users.

Custom connectors

Developers can use the Power Query SDK to create *custom connectors*. Custom connectors allow access to proprietary data sources or implement specific authentication with custom data extensions. Some custom connectors are certified and distributed by Microsoft as *certified connectors*. Certified connectors have been audited and reviewed to ensure that they meet certain specified code requirements that Microsoft has tested and approved.

There's a Power BI Desktop data extension security option that restricts the use of non-certified connectors. By default, an error is raised when an attempt is made to load a

non-certified connector. By setting this option to allow non-certified connectors, custom connectors will load without validation or warning.

We recommend that you keep your data extension security level at the higher level, which prevents loading of non-certified code. However, there might be cases where you want to load specific connectors, perhaps connectors that you've developed, or connectors provided to you by a trusted consultant or vendor outside the Microsoft certification path.

> **Note**
>
> Developers of in-house-developed connectors can take steps to sign a connector with a certificate, allowing you to use the connector without the need to change your security settings. For more information, see Trusted third-party connectors.

Checklist - When planning data source permissions, key decisions and actions include:

- ✓ **Decide who can directly access each data source**: Determine which data creators are permitted to access a data source directly. If there's a strategy to reduce the number of people with direct access, clarify what the preferred alternative is (perhaps by using dataflows).
- ✓ **Decide how data sources should be accessed**: Determine whether individual user credentials will be used for accessing a data source, or whether a service account should be created for that purpose. Determine when single sign-on is appropriate.
- ✓ **Provide guidance for semantic model creators about accessing data sources**: Include documentation for content creators about how to access organizational data sources. Publish the information to your centralized portal and training materials.
- ✓ **Provide guidance for semantic model creators about privacy levels**: Provide guidance to semantic model creators to make them aware of privacy levels, and their implications when working with sensitive or confidential data. Publish this information to your centralized portal and training materials.
- ✓ **Provide guidance for gateway connection creators about privacy levels**: Provide guidance to semantic model creators to make them aware of privacy levels and their implications when working with sensitive or confidential data. Publish this information to your centralized portal and training materials.

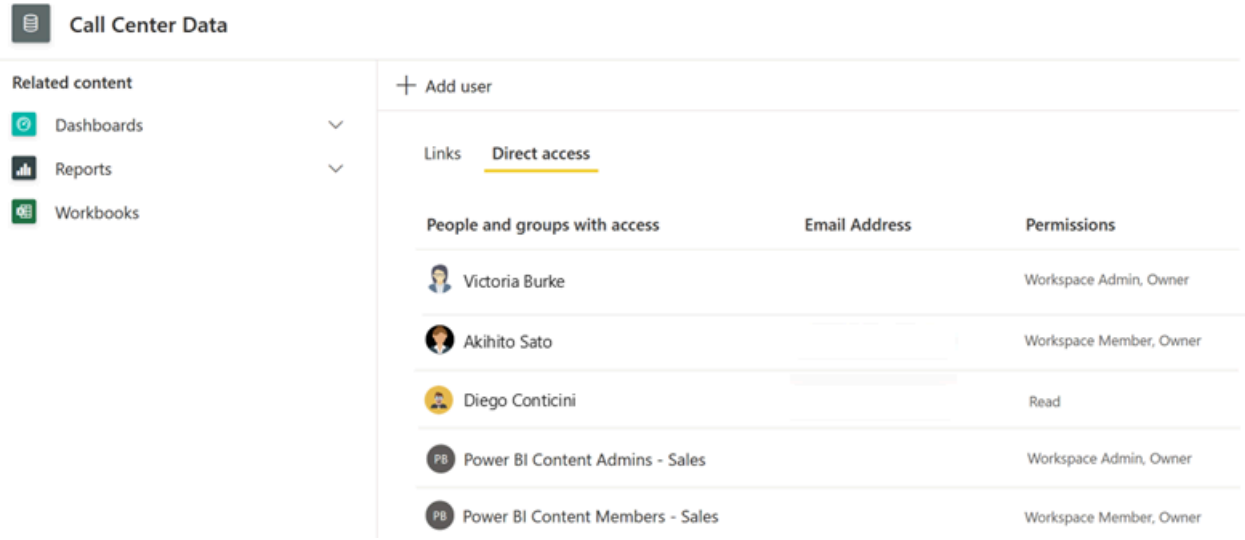
- ✓ **Decide on the strategy for using native database queries**: Consider your strategy for using native database queries. Educate semantic model creators on how and when to set the Power BI Desktop native database queries option to disable pre-approval when Power Query runs native queries.
- ✓ **Decide on the strategy for using custom connectors**: Consider your strategy for using custom connectors. Determine whether the use of non-certified connectors is justified, in which case educate semantic model creators on how and when to set the Power BI Desktop data extension option.

Semantic model creator permissions

You can assign permission to edit a semantic model to a user or group in different ways.

- **Workspace role**: Assignment to any of the workspace roles provides access to all semantic models in the workspace. The ability to view or edit an existing semantic model depends on the workspace role that you assign. Administrators, members, and contributors can publish or edit content within a workspace.
- **Per-item permission links**: If a sharing link was created for a report, permission to read the semantic model (and optionally, build, write, and/or reshare) is also indirectly granted by the link.
- **Per-item direct access permissions**: You can assign the direct access permission to a specific semantic model.

In the following screenshot, notice the permissions assigned to the **Call Center Data** semantic model. One user has Read permission, which was granted by using per-item direct access permissions. The remaining users and groups have permissions because they're assigned to workspace roles.



> **Tip**

Using per-item permissions (links or direct access) works best when the intention is for a user or group to view or edit one specific item in the workspace. It's best suited when the user isn't permitted to access all items in the workspace. In most cases, we recommend that you design your workspaces so that security is simpler to manage with workspace roles. Avoid setting per-item permissions whenever possible.

Semantic model permissions

You can assign the following semantic model permissions.

- **Read**: Targeted primarily at report consumers, this permission allows a report to query data in the semantic model. For more information about permissions for viewing read-only content, see the Report consumer security planning article.
- **Build**: Targeted at report creators, this permission allows users to create new reports based on the shared semantic model. For more information, see the Report creator permissions section later in this article.
- **Write**: Targeted at semantic model creators who create, publish, and manage semantic models, this permission allows users to edit the semantic model. It's described later in this section.
- **Reshare**: Targeted at anyone with existing permission to the semantic model, this permission allows users to share the semantic model with another user. It's described later in this section.

A workspace administrator or member can edit the permissions for a semantic model.

Semantic model Read permission

The semantic model Read permission is primarily targeted at consumers. This permission is required for users to be able to view data that's displayed in reports. Be aware that reports based on the semantic model must also have Read permission; otherwise, the report will fail to load. For more information about setting report Read permissions, see the Report consumer security planning article.

Semantic model Build permission

In addition to semantic model Read permission, content creators also need the semantic model Build permission. Specifically, the Build permission allows report creators to:

- Create new Power BI reports based on the semantic model.
- Connect to the semantic model by using Analyze in Excel.

- Query the semantic model by using the XMLA endpoint.
- Export Power BI report visual underlying data (instead of the summarized data retrieved by the visual).
- Create a DirectQuery connection to a Power BI semantic model. In this case, the new semantic model connects to one or more existing Power BI semantic models (known as *chaining*). To query chained semantic models, the semantic model creator will need Build permission for all upstream semantic models. For more information, see Chained semantic models later in this article.

You can grant Build permission to a user or group, directly or indirectly, in different ways.

- Grant Build directly by:
  - Setting semantic model permissions on the semantic model settings page in the Power BI service.
  - Setting semantic model permissions by using the Power BI REST API.
- Grant Build indirectly by:
  - Sharing a report or dashboard and setting the option to grant semantic model Build permission.
  - Publishing a Power BI app and setting the advanced option (for an audience) to grant Build permission on the related semantic models.
  - Assigning users to the Admin, Member, or Contributor workspace roles.

Setting Build permission directly for a semantic model is appropriate when you want to manage security on a granular, per-item basis. Setting Build permission indirectly is appropriate when the users who will view or use the content through one of the indirect methods will also create new content.

> **Tip**
>
> Frequently, the users who view a report or a Power BI app are different from the users who create new content by using the underlying semantic model(s). Most consumers are viewers only, so they don't need to create new content. We recommend that you educate your content creators to grant the least number of permissions that are required.

Semantic model Write permission

Usually, setting permissions for who can edit and manage semantic models will be done by assigning users to either the Admin, Member, or Contributor workspace role. However, it's also possible to set Write permission for a specific semantic model.

We recommend that you use workspace roles whenever possible because it's the simplest way to manage and audit permissions. Use the semantic model Write permissions on a per-item basis when you've chosen to create fewer workspaces, and a workspace contains semantic models for different subject areas that require different permissions management.

> **Tip**
>
> For guidance on how to organize workspaces, see the **workspace planning** articles.

Semantic model Reshare permission

The semantic model Reshare permission allows a user with existing permission to share the semantic model with other users. You can grant this permission when content in the semantic model can be freely shared, based on user discretion.

In many cases, we recommend limiting the use of the Reshare permission to ensure that semantic model permissions are carefully controlled. Get approval from the semantic model owner(s) before granting the Reshare permission.

Semantic model data security

You can plan to create fewer semantic models and reports by enforcing data security. The objective is to enforce data security based on the identity of the user who's viewing the content.

A semantic model creator can enforce data security in two ways.

- Row-level security (RLS) allows a data modeler to restrict access to a subset of data.
- Object-level security (OLS) allows a data modeler to restrict access to specific tables and columns, and their metadata.

The implementation of RLS and OLS are targeted at report consumers. For more information, see Report consumer security planning article. It describes how and when RLS and OLS are enforced for consumers who have view-only permission to the semantic model.

For RLS and OLS targeted at other report creators, see data security in the Report creator permissions section later in this article.

Chained semantic models

Power BI semantic models can connect to other semantic models in a process known as *chaining*, which are connections to upstream semantic models. For more information, see Use composite models in Power BI Desktop.

The *Allow DirectQuery connections to Power BI semantic models* tenant setting allows Power BI administrators to set up which groups of content creators can create chained semantic models. If you don't want to restrict semantic model creators from chaining semantic models, you can leave this setting enabled for the entire organization and rely on workspace access and semantic model permissions. In some cases, you might consider restricting this capability to approved content creators.

> **Note**
>
> As a semantic model creator, you can restrict chaining to your semantic model. It's done by enabling the *Discourage DirectQuery connection to this semantic model* option in Power BI Desktop. For more information, see Manage DirectQuery connections to a published semantic model.

Semantic model API queries

In some situations, you might want to execute a DAX query by using the Power BI REST API. For example, you might want to perform data quality validations. For more information, see Datasets - Execute Queries.

The *Dataset Execute Queries REST API* tenant setting allows Power BI administrators to set which groups of users can send DAX queries by using the Power BI REST API. In most cases, you can leave this setting enabled for the entire organization and rely on workspace access and semantic model permissions. In some cases, you might consider restricting this capability to approved content creators.

Checklist - When planning for semantic model creator permissions, key decisions and actions include:

- ✓ **Decide on the strategy for semantic model creator permissions**: Determine what preferences and requirements exist for managing security for semantic model creators. Consider the subject area and level of data sensitivity. Also consider who's

allowed to take responsibility for managing data and permissions in centralized and decentralized business units.

- ✓ **Review how workspace roles are handled for semantic model creators**: Determine what the impact is on your workspace design process. Create separate data workspaces for each subject area so that you can more easily manage the workspace roles and semantic model security for each subject area.
- ✓ **Provide guidance for semantic model creators about managing permissions**: Include documentation for semantic model creators about how to manage semantic model permissions. Publish this information to your centralized portal and training materials.
- ✓ **Decide who can use DirectQuery connections for Power BI semantic models**: Decide whether there should be any limitations for which Power BI semantic model creators (with existing Build permission for a semantic model) can create a connection to a Power BI semantic model. Set the *Allow DirectQuery connections to Power BI semantic models* tenant setting to align with this decision. If you decide to limit this capability, consider using a group such as *Power BI approved semantic model creators*.
- ✓ **Decide who can query Power BI semantic models by using the REST API**: Decide whether to restrict Power BI content creators from querying Power BI semantic models by using the Power BI REST API. Set the *Dataset Execute Queries REST API* tenant setting to align with this decision. If you decide to limit this capability, consider using a group such as *Power BI approved report creators*.
- ✓ **Decide on the strategy for the use of RLS or OLS for semantic model creators**: Consider which use cases and purposes you intend to use RLS or OLS. Factor in the workspace design strategy, and who has read versus edit permissions, when you want to enforce RLS or OLS for semantic model creators.

Report creator permissions

Report creators need workspace access to create reports in the Power BI service or publish them from Power BI Desktop. They must be either an administrator, member, or contributor in the target workspace.

Whenever possible, report creators should use an existing shared semantic model (via a live connection or DirectQuery). That way, the report creation process is *decoupled* from the semantic model creation process. This type of separation provides many benefits for security and team development scenarios.

A report creator needs to be a workspace administrator, member, or contributor.

Unlike semantic models, there isn't a Write permission for reports. To support report creators, workspace roles must be used. For this reason, optimal workspace design is

important to balance content organization and security needs.

> **Tip**
>
> For permissions to support report consumers (including the Read and Reshare per-item permissions), see the **Report consumer security planning** article.

Read and Build permissions for underlying semantic model

Report creators must have Read and Build permissions on the semantic models that their reports will use, which includes chained semantic models. That permission can be granted explicitly on the individual semantic models, or it can be granted implicitly for workspace semantic models when the report creator is a workspace administrator, member, or contributor.

The *Use semantic models across workspaces* tenant setting allows Power BI administrators to set up which groups of users can create reports that use semantic models located in other workspaces. This setting is targeted at semantic model and report creators. Usually, we recommend that you leave this setting enabled for the entire organization and relying on workspace access settings and semantic model permissions. That way, you can encourage the use of existing semantic models. In some cases, you might consider restricting this capability only to approved content creators.

There's also the *Allow live connections* tenant setting, which allows Power BI administrators to set up which groups of users can create live connections to semantic models in Power BI Desktop or Excel. It's targeted specifically at report creators, and it also requires that they're granted Read and Build permission on the semantic model that the report will use. We recommend that you leave this setting enabled for the entire organization and rely on workspace access and semantic model permissions. That way, you can encourage the use of existing semantic models. In some cases, you might consider restricting this capability only to approved content creators.

Data security for underlying semantic model

RLS and OLS (described previously in this article) are targeted at report consumers. However, sometimes it also needs to be enforced for report creators. Creating separate workspaces is justified when RLS needs to be enforced for report creators and also report consumers.

Consider the following scenario.

- **Centralized shared semantic models with RLS**: The enterprise BI team published sales semantic models to the *Sales Data* workspace. These semantic models enforce RLS to show sales data for the assigned sales region of the report consumer.
- **Decentralized self-service report creators**: The sales and marketing business unit has many capable analysts who create their own reports. They publish their reports to the *Sales Analytics* workspace.
- **Read and Build permissions for semantic models**: Whenever possible, the analysts use the semantic models from the *Sales Data* workspace to avoid the unnecessary duplication of data. Because the analysts only have Read and Build permissions on these semantic models (with no write or edit permissions), RLS is enforced for the report creators (and also the report consumers).
- **Edit permissions for reporting workspace**: The analysts have more rights in the *Sales Analytics* workspace. The administrator, member, or contributor workspace roles allow them to publish and manage their reports.

For more information about RLS and OLS, see the Report consumer security planning article. It describes how and when RLS and OLS are enforced for consumers who have view-only permission to the semantic model.

Connecting to external semantic models

When a report creator connects to a shared semantic model for their report, they usually connect to a shared semantic model that's been published in their own Power BI tenant. When permission has been granted, it's also possible to connect to a shared semantic model in another tenant. The other tenant could be a partner, customer, or vendor.

This functionality is known as in-place semantic model sharing (also known as *cross-tenant semantic model sharing*). The reports created by the report creator (or new composite models created by a semantic model creator) are stored and secured in your Power BI tenant by using your normal process. The original shared semantic model remains in its original Power BI tenant, and all permissions are managed there.

For more information, see the Tenant-level security planning article. It includes information about the tenant settings and semantic model settings that make external sharing work.

Featured tables

In Power BI Desktop, semantic model creators can set a model table to become a featured table. When the semantic model is published to the Power BI service, report

creators can use the Data Types Gallery in Excel to find the featured table, allowing them to add featured table data to augment their Excel worksheets.

The *Allow connections to featured tables* tenant setting allows Fabric administrators to set up which groups of users can access featured tables. It's targeted at Excel users who want to access Power BI featured tables in Excel organization data types. We recommend that you leave this setting enabled for the entire organization and rely on workspace access and semantic model permissions. That way, you can encourage the use of featured tables.

Custom visual permissions

In addition to core visuals, Power BI report creators can use *custom visuals*. In Power BI Desktop, custom visuals can be download from Microsoft AppSource. They can also be developed in-house by using the Power BI SDK, and installed by opening the visual file (.pbviz).

Some visuals available for download from AppSource are *certified visuals*. Certified visuals meet certain specified code requirements that the Power BI team has tested and approved. The tests check that visuals don't access external services or resources.

The *Allow visuals created by the Power BI SDK* tenant setting allows Power BI administrators to control which groups of users can use custom visuals.

There's also the *Add and use certified visuals only* tenant setting, which allows Power BI administrators to block the use of non-certified visuals in the Power BI service. This setting can be enabled or disabled for the entire organization.

> **Note**
>
> If you block the use of non-certified visuals, it only applies to the Power BI service. If you want to restrict their use in Power BI Desktop, ask your Fabric administrator to use a **group policy setting** to block their use in Power BI Desktop. Taking this step will ensure that report creators don't waste time and effort creating a report that won't work when published to the Power BI service. We highly recommend that you set up your users to have consistent experiences in the Power BI service (with the tenant setting) and Power BI Desktop (with group policy).

Power BI Desktop has an option to show a security warning when a report creator adds a custom visual to the report. Report creators can disable this option. This option doesn't test whether the visual is certified.

Power BI administrators can approve and deploy custom visuals for their organization. Report creators can then easily discover, update, and use these visuals. Administrators can then manage these visuals by updating versions or disabling and enabling specific custom visuals. This approach is useful when you want to make an in-house-developed visual available to your report creators, or when you acquire a custom visual from a vendor that isn't in AppSource. For more information, see Power BI organizational visuals.

Consider a balanced strategy of enabling only certified custom visuals in your organization (with the tenant setting and group policy previously described), while deploying organizational visuals to handle any exceptions.

Checklist - When planning for report creator permissions, key decisions and actions include:

- ✓ **Decide on the strategy for report creator permissions**: Determine what preferences and requirements exist for managing security for report creators. Consider the subject area and level of data sensitivity. Also, consider who's allowed to take responsibility for creating and managing reports in centralized and decentralized business units.
- ✓ **Review how workspace roles are handled for report creators**: Determine what the impact is on your workspace design process. Create separate data workspaces and reporting workspaces for each subject area, so the workspace roles (and underlying semantic model security) are simplified for the subject area.
- ✓ **Provide guidance for report creators about managing permissions**: Include documentation for report creators about how to manage permissions for report consumers. Publish this information to your centralized portal and training materials.
- ✓ **Decide who can use shared semantic models**: Decide whether there should be any limitations for which Power BI report creators (who already have Read and Build permissions for a semantic model) can use semantic models across workspaces. Set the *Use semantic models across workspaces* tenant setting to align with this decision. If you decide to limit this capability, consider using a group such as *Power BI approved report creators*.
- ✓ **Decide who can use live connections**: Decide whether there should be any limitations for which Power BI report creators (who already have Read and Build permission for a semantic model) can use live connections. Set the *Allow live*

connections tenant setting to align with this decision. If you decide to limit this capability, consider using a group such as *Power BI approved report creators*.

- ✓ **Decide on the strategy for use of RLS for report creators**: Consider which use cases and purposes you intend to use row-level security. Factor in the workspace design strategy to ensure that RLS is enforced for report creators.
- ✓ **Decide on the strategy for use of custom visuals**: Consider your strategy for which report creators can use custom visuals. Set the *Allow visuals created by the Power BI SDK* tenant setting to align with this decision. Create a process for using organizational visuals, when appropriate.

Dataflow creator permissions

Dataflows are helpful for centralizing data preparation so that the work done in Power Query isn't repeated across many semantic models. They're a building block for achieving a *single source of truth*, preventing analysts from requiring direct access to sources, and for helping to perform extract, transform, and load (ETL) operations at scale.

A dataflow creator needs to be a workspace administrator, member, or contributor.

To consume a dataflow (for instance, from a new data model created in Power BI Desktop or in another workspace), a semantic model creator can belong to any workspace role, including the Viewer role. There's no concept of RLS for dataflows.

In addition to workspace roles, the *Create and use dataflows* tenant setting must be enabled. This tenant setting applies to the entire organization.

Consider the following scenario.

- Many semantic models in the organization need to enforce dynamic RLS. It requires that user principal names (UPNs) be stored in the semantic model (to filter by the identity of the report consumer).
- A dataflow creator, who belongs to the Human Resources department, creates a dataflow of current employee details including their UPNs. They set up the dataflow to refresh daily.
- Semantic model creators then consume the dataflow in their model designs to set up RLS.

For more information about using dataflows, see the self-service data preparation and advanced data preparation usage scenarios.

Checklist - When planning for dataflow creator permissions, key decisions and actions include:

- ✓ **Decide on the strategy for dataflow creator permissions**: Determine what preferences and requirements exist for managing security for dataflow creators. Consider who's allowed, or encouraged, to take responsibility for managing data preparation activities in centralized and decentralized business units.
- ✓ **Decide who can create dataflows**: Decide whether there should be any limitations for which Power BI data creators can create dataflows. Set the *Create and use dataflows* tenant setting to align with this decision.
- ✓ **Review how workspace roles are handled for dataflow creators**: Determine what the impact is on your workspace design process. Create separate dataflow workspaces per subject area so that you can handle workspace roles and permissions separately for each the subject area, when appropriate.

Datamart creator permissions

A datamart is a self-service analytics solution that enables users to store and explore data that's loaded in a fully managed relational database. It also comprises an auto-generated semantic model.

Datamarts provide a simple low-code experience to ingest data from different data sources, and to extract, transform, and load (ETL) the data by using Power Query Online. The data is loaded into an Azure SQL Database that's fully managed and requires no tuning or optimization. The auto-generated semantic model is always synchronized with the managed database because it's in DirectQuery mode.

You can create a datamart when you're either a workspace administrator, member, or contributor. Workspace roles also get mapped to database-level roles in the Azure SQL Database (however, because the database is fully managed, user permissions can't be edited or managed in the relational database).

The *Create datamarts* tenant setting allows Power BI administrators to set up which groups of users can create datamarts.

Datamart sharing

For datamarts, the term *sharing* takes on a meaning that's different to other Power BI content types. Usually, a sharing operation is targeted at a consumer because it provides read-only permission to one item, like a report.

Sharing a datamart is targeted at content creators (rather than consumers). It grants the Read and Build permissions, which allows users to query either the semantic model or the relational database, whichever they prefer.

Sharing a datamart allows content creators to:

- **Build content by using the auto-generated semantic model**: The semantic model is the semantic layer on which Power BI reports can be built. Most report creators should use the semantic model.
- **Connect to and query the Azure SQL Database**: The relational database is useful for content creators who want to create new semantic models or paginated reports. They can write structured query language (SQL) queries to retrieve data by using the SQL endpoint.

Datamart row-level security

You can define RLS for datamarts to restrict data access for specified users. RLS is set up in the datamart editor in the Power BI service, and it's automatically applied to the auto-generated semantic model and the Azure SQL Database (as security rules).

Regardless of how a user chooses to connect to the datamart (to the semantic model or the database), identical RLS permissions are enforced.

Checklist - When planning for datamart creator permissions, key decisions and actions include:

- ✓ **Decide on the strategy for datamart creator permissions**: Determine what preferences and requirements exist for managing security for datamart creators. Consider who's allowed, or encouraged, to take responsibility for managing data in centralized and decentralized business units.
- ✓ **Decide who can create datamarts**: Decide whether there should be any limitations for which Power BI data creators can create a datamart. Set the *Create datamarts* tenant setting to align with this decision. If you decide to limit who can create datamarts, consider using a group such as *Power BI approved datamart creators*.

- ✓ **Review how workspace roles are handled for datamart creators**: Determine what the impact is on your workspace design process. Create separate data workspaces per subject area so the workspace roles and semantic model security can be simplified for the subject area.
- ✓ **Provide guidance for datamart creators about managing permissions**: Include documentation for datamart creators about how to manage datamart permissions. Publish this information to your centralized portal and training materials.
- ✓ **Decide on the strategy for using RLS in datamarts**: Consider which use cases and purposes you intend to use RLS within a datamart.

Scorecard creator permissions

Metrics in Power BI let you curate specific metrics and track them against key business objectives. Metrics are added to scorecards, which can be shared with other users and viewed in a single pane.

Scorecards can be secured with three levels of permissions:

- Workspace.
- Scorecard (per-item) permissions.
- Metrics (within the scorecard).

A user who creates, or fully manages, a scorecard needs to be a workspace administrator, member, or contributor.

Because metrics often span multiple subject areas, we recommend that you create a separate workspace so that you can independently manage permissions for creators and consumers.

The *Create and use metrics* tenant setting allows Power BI administrators to set up which groups of users can create scorecard metrics.

Scorecard permissions

You can assign the following scorecard permissions.

- **Read**: This permission allows a user to view the scorecard.
- **Reshare**: Targeted at anyone with existing permission to the scorecard, this permission allows users to share the scorecard with another user.

Consistent with other content types in the Power BI service, the per-item permissions are useful when the intention is to share one item with another user. We recommend using workspace roles and app permissions whenever possible.

Metric-level permissions

Each scorecard has a set of metric-level permissions that you can set up in the scorecard settings. The metric-level permissions (within a scorecard) can be granted differently from the workspace or the scorecard (per-item) permissions.

The metric-level roles allow you to set:

- Who can view individual metrics on a scorecard.
- Who can update individual metrics by:
  - Updating the status during a check-in.
  - Adding notes during a check-in.
  - Updating the current value during a check-in.

To reduce the level of future maintenance, it's possible to set default permissions that will be inherited by submetrics you create in the future.

Checklist - When planning for metric creator permissions, key decisions and actions include:

- ✓ **Decide on the strategy for scorecard creator permissions**: Determine what preferences and requirements exist for managing security for scorecard creators. Consider who's allowed, or encouraged, to take responsibility for managing data in centralized and decentralized business units.
- ✓ **Decide who can create scorecards**: Decide whether there should be any limitations for which Power BI data creators can create scorecards. Set the *Create and use Metrics* tenant setting to align with this decision. If you decide to limit who can create scorecards, consider using a group such as *Power BI approved scorecard creators*.
- ✓ **Review how workspace roles are handled for scorecard creators**: Determine what the impact is on your workspace design process. Consider creating separate workspaces for scorecards when the content spans subject areas.
- ✓ **Provide guidance for scorecard creators about managing permissions**: Include documentation for scorecard creators about how to manage metric-level permissions. Publish this information to your centralized portal and training materials.

Publishing content

This section includes topics related to publishing content that are relevant to content creators.

Workspaces

Content creators will need administrator, member, or contributor role access to publish content to a workspace. For more information, see the workspace roles described earlier in this article.

Except for personal BI, content creators should be encouraged to publish content to standard workspaces, instead of their personal workspace.

The *Block republish and disable package refresh* tenant setting changes the behavior for publishing semantic models. When enabled, workspace administrators, members, or contributors can't publish changes to a semantic model. Only the semantic model owner is permitted to publish an update (forcing the takeover of a semantic model before publishing an updated semantic model). Because this tenant setting applies to the entire organization, enable it with a measure of caution because it affects all semantic models for the entire tenant. Be sure to communicate to your semantic model creators what to expect because it changes the normal behavior of workspace roles.

Power Apps synchronization

It's possible to create a Power Apps solution that includes embedded Power BI reports. The Power Apps process will automatically create a dedicated Power BI workspace for storing and securing the Power BI reports and semantic models. To manage items that exist in both Power Apps and Power BI, there's a synchronization process.

The process synchronizes security roles to ensure that Power BI inherits the same roles that were initially set up in Power Apps. It also allows the content creator to manage permissions for who can view the Power BI reports (and related semantic models) that are embedded in a Power App.

For more information about synchronizing Power Apps roles with Power BI workspace roles, see Permission sync between Power Apps environment and Power BI workspace.

Deployment pipeline access

Content creators and owners can use Power BI deployment pipelines for self-service content publishing. Deployment pipelines simplify the publication process and improve the level of control when releasing new content.

You manage pipeline permissions (for users who can deploy content with a deployment pipeline) separately from the workspace roles. Access to both the workspace and the deployment pipeline are required for the users conducting a deployment.

Content creators might also need:

- Workspace creation permissions (when workspaces need to be created by the pipeline).
- Premium or Fabric capacity permissions (when workspaces are assigned by the pipeline).

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

For more information, see Deployment pipeline access.

XMLA endpoint

The XMLA endpoint uses the XMLA protocol to expose all features of a tabular data model, including some data modeling operations that aren't supported by Power BI Desktop. You can use the Tabular Object Model (TOM) API to make programmatic changes to a data model.

The XMLA endpoint also provides connectivity. You can only connect to a semantic model when the workspace has its license mode set to **Premium per user**, **Premium per capacity**, or **Embedded**. Once a connection is made, an XMLA-compliant tool can operate on the data model to read or write data. For more information about how you can use the XMLA endpoint for managing a semantic model, see the advanced data model management usage scenario.

Access through the XMLA endpoint will honor existing permissions. Workspace administrators, members, and contributors implicitly have semantic model Write permission, which means they can deploy new semantic models from Visual Studio and execute Tabular Modeling Scripting Language (TMSL) scripts in SQL Server Management Studio (SSMS).

Users with the semantic model Build permission can use the XMLA endpoint to connect to and browse semantic models for data consumption and visualization. RLS rules are honored, and users can't see internal semantic model metadata.

The *Allow XMLA endpoints and Analyze in Excel with on-premises semantic models* tenant setting refers to two capabilities: It controls which groups of users can use the XMLA endpoint to query and/or maintain semantic models in the Power BI service. It also determines whether Analyze in Excel can be used with on-premises SQL Server Analysis Services (SSAS) models.

> **Note**
>
> The Analyze in Excel aspect of that tenant setting only applies to on-premises SQL Server Analysis Services (SSAS) models. The standard Analyze in Excel functionality, which connects to a Power BI semantic model in the Power BI service, is controlled by the *Allow Live Connections* tenant setting.

Publish to web

Publish to web is a feature that provides access to Power BI reports to anyone on the internet. It doesn't require authentication and access isn't logged for auditing purposes. Because report consumers don't need to belong to the organization or have a Power BI license, this technique is well suited to data journalism, a process where reports are embedded in blog posts, websites, emails, or social media.

> **Caution**
>
> Publish to web has the potential to expose sensitive or confidential data, whether accidentally or intentionally. For this reason, this feature is disabled by default. Publish to web should only be used for reports that contain data that can be viewed by the public.

The *Publish to web* tenant setting allows Power BI administrators to control which groups of users can publish reports to the web. To maintain a higher level of control, we recommend that you don't include other groups in this tenant setting (like Power BI administrators or other types of content creators). Instead, enforce specific user access by using a security group such as *Power BI public publishing*. Ensure that the security group is well managed.

Embedding in custom apps

The *Embed content in apps* tenant setting allows Power BI administrators to control which users can embed Power BI content outside of the Power BI service. When you plan to embed Power BI content in custom applications, enable this setting for specific groups, such as app developers.

Embedding in PowerPoint

The *Enable Power BI add-in for PowerPoint* tenant setting allows Power BI administrators to control which users can embed Power BI report pages in PowerPoint presentations. When appropriate, enable this setting for specific groups, such as report creators.

> **Note**
>
> For this capability to work, users must install the Power BI add-in for PowerPoint. To use the add-in, users must either have access to the Office add-in store, or the add-in must be made available to them as an **admin managed add-in**. For more information, see **Power BI add-in for PowerPoint**.

Educate report creators to be cautious about where they save their PowerPoint presentations and who they share them with. That's because an image of the Power BI report visuals is shown to users when they open the presentation. That image is captured from the last time the PowerPoint file was connected. However, the image could inadvertently reveal data that the receiving user doesn't have permission to see.

> **Note**
>
> The image can be useful when the receiving user doesn't yet have the add-in, or until the add-in connects to the Power BI service to retrieve data. Once the user connects, only data the user can see (enforcing any RLS) is retrieved from Power BI.

Template apps

Template apps enable Power BI partners and software vendors to build Power BI apps with little or no coding, and deploy them to any Power BI customer.

The *Publish template apps* tenant setting allows Power BI administrators to control which users can publish template apps outside of the organization, such as through Microsoft AppSource. For most organizations, this tenant setting should be disabled or tightly controlled. Consider using a security group such as *Power BI external template app creators*.

Email subscriptions

You can subscribe yourself and others to Power BI reports, dashboards, and paginated reports. Power BI will then send an email on a schedule you set. The email will contain a snapshot and link to the report or dashboard.

You can create a subscription that includes other users when you're a workspace administrator, member, or contributor. If the report is in a Premium workspace, you can subscribe groups (whether they're in your domain or not) and external users. When setting up the subscription, there's also the option to grant permissions to the item. Per-item direct access permissions are used for this purpose, which are described in the Report consumer security planning article.

> **⊗ Caution**
>
> The email subscription feature has the potential to share content to internal and external audiences. Also, when RLS is enforced on the underlying semantic model, attachments and images are generated by using the security context of the subscribing user.

The *Email subscriptions* tenant setting allows Power BI administrators to control whether this feature is enabled or disabled for the entire organization.

There are some limitations concerning attachments related to licensing and tenant setting restrictions. For more information, see Email subscriptions for reports and dashboards in the Power BI service.

Checklist - When planning for publishing content, key decisions and actions include:

- ✓ **Decide on the strategy for where content should be published, how, and by whom**: Determine what preferences and requirements exist for where content gets published.
- ✓ **Verify workspace access**: Confirm the workspace design approach. Verify how to use the workspace access roles to support your strategy for where content should be published.
- ✓ **Determine how to handle deployment pipeline permissions**: Decide which users are permitted to publish content by using a deployment pipeline. Set the

deployment pipeline permissions accordingly. Ensure that workspace access is also provided as well.

- ✓ **Decide who can connect to semantic models by using the XMLA endpoint**: Decide which users are permitted to query or manage semantic models by using the XMLA endpoint. Set the *Allow XMLA endpoints and Analyze in Excel with on-premises semantic models* tenant setting to align with this decision. When you decide to limit this capability, consider using a group such as *Power BI approved content creators*.
- ✓ **Decide who can publish reports publicly**: Decide which users are permitted to publish Power BI reports publicly, if any. Set the *Publish to web* tenant setting to align with this decision. Use a group such as *Power BI public publishing*.
- ✓ **Decide who can embed content in custom apps**: Determine who should be allowed to embed content outside of the Power BI service. Set the *Embed content in apps* tenant setting to align with this decision.
- ✓ **Decide who can embed content in PowerPoint**: Determine who should be allowed to embed content in PowerPoint. Set the *Enable Power BI add-in for PowerPoint* tenant setting to align with this decision.
- ✓ **Decide who can publish template apps**: Determine what your strategy is for using template apps outside of the organization. Set the *Publish template apps* tenant setting to align with this decision.
- ✓ **Decide whether to enable subscriptions**: Confirm what your strategy is for using subscriptions. Set the *Email Subscriptions* tenant setting to align with this decision.

Refresh data

Once published, data creators should ensure that their semantic models and dataflows (that contain imported data) are periodically refreshed. You should also decide on appropriate strategies for the semantic model and dataflow owners.

Semantic model owner

Each semantic model has an owner, which is a single user account. The semantic model owner is required to set up semantic model refresh and set semantic model parameters.

By default, the semantic model owner also receives access requests from report creators who want Build permissions (unless the *Request access settings* for the semantic model are set to provide custom instructions). For more information, see the Request access workflow for creators section in this article.

There are two ways that Power BI can obtain credentials to refresh a semantic model.

- The semantic model owner stores credentials in the semantic model settings.

- The semantic model owner references a gateway in the semantic model settings (that contains a data source with stored credentials).

If a different user needs to set up refresh or set semantic model parameters, they must take ownership of the semantic model. semantic model ownership can be taken over by a workspace administrator, member, or contributor.

> ⊗ **Caution**
>
> Taking semantic model ownership permanently removes any stored credentials for the semantic model. Credentials must be re-entered to allow data refresh operations to resume.

Ideally, the semantic model owner is the user who's responsible for the semantic model. You should update the semantic model owner when they leave the organization or change roles. Also, be aware that when the semantic model owner's user account is disabled in Microsoft Entra ID, data refresh is automatically disabled. In this case, another user must take ownership of the semantic model to allow data refresh operations to resume.

Dataflow owner

Like semantic models, dataflows also have an owner, which is a single user account. The information and guidance provided in the previous topic about semantic model owners also applies to dataflow owners.

Checklist - When planning for security related to data refresh processes, key decisions and actions include:

- ✓ **Decide on the strategy for semantic model owners**: Determine what preferences and requirements exist for managing semantic model owners.
- ✓ **Decide on the strategy for dataflow owners**: Determine what preferences and requirements exist for managing dataflow owners.
- ✓ **Include in documentation and training for semantic model creators**: Include guidance for your data creators about how to manage owners for each type of item.

Related content

For other considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see the subject areas to consider.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Information protection and data loss prevention

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces the Power BI information protection and data loss prevention (DLP) articles. These articles are targeted at multiple audiences:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators need to collaborate with information security teams and other relevant teams.
- **Center of Excellence, IT, and BI teams**: The teams that are responsible for overseeing Power BI in the organization. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.

> **Important**
>
> Information protection and DLP is a significant organization-wide undertaking. Its scope and impact are far greater than Power BI alone. This type of initiative requires funding, prioritization, and planning. Expect to involve several cross-functional teams in your planning, usage, and oversight efforts.

As someone who manages Power BI for your organization, you won't usually be directly responsible for most aspects of information protection and DLP. It's likely those responsibilities will fall to the information security team and other system administrators.

The focus for this set of articles includes:

- **Why**: Why these capabilities are important for compliance and auditing.
- **What**: An overview of what the end-to-end process involves.
- **Who**: Which teams participate in the end-to-end process.

- **Prerequisites**: The things that need to be in place before the information protection and DLP capabilities can be enabled for Power BI.

> **Important**
>
> The **Power BI administrator** role has been renamed. The new name of the role is **Fabric administrator**.

Protect organizational data

Data exists in many applications and services. It's stored in source databases and files. It's published to the Power BI service. It also exists outside the Power BI service as original files, downloaded files, and exported data. When data becomes more accessible and across more resources, the way you go about protecting data becomes increasingly important.

In short, protecting data is about:

- Safeguarding organizational data.
- Reducing the risk of unauthorized or unintentional sharing of sensitive information.
- Strengthening compliance status for regulatory requirements.

Protecting data is a complex subject. At a high level, topics relevant to Power BI include:

- **Responsible actions taken by users**: Users who have received guidance and training, and clearly understand what's expected of them, can act ethically. They can enact a culture that values security, privacy, and compliance during the normal course of their work.
- **Right-sized user security permissions**: In Power BI, securing data and reports is separate and distinct from the information protection and DLP activities described in these articles. Security methods in Power BI include techniques such as workspace roles, sharing, app permissions, and row-level security (RLS). Security techniques, such as workspace roles, app permissions, per-item sharing, and RLS, are covered in the security planning articles.
- **Data lifecycle management**: Processes such as backups and version control are important for protecting data. The setup of encryption keys and geographic locations for data storage also are considerations.
- **Information protection**: Labeling and classifying content by using sensitivity labels is the first step towards being able to protect it. Information protection is covered in this series of articles.

- **Data loss prevention policies**: DLP refers to controls and policies that reduce the risk of data leakage. Data loss prevention is covered in this series of articles.

The information protection and DLP series of articles focus on the final two bullet points: information protection and DLP, and specifically how they relate to Power BI.

We recommend that you also become familiar with the full Microsoft Purview Information Protection framework: know your data, protect your data, prevent data loss, and govern your data.

> **Tip**
>
> Your organization's IT department will have existing processes in place that are considered information protection, but they're out of scope for this series of articles. Processes could include high availability and disaster recovery efforts related to source database systems. They could also include protecting mobile devices. Be sure to identify and involve relevant technology and governance teams in all your planning efforts.

Common use cases

Power BI compliance challenges and regulatory reporting requirements are frequently a driving factor for getting started with information protection and DLP.

> **Tip**
>
> Data leakage refers to the risk of data being viewed by unauthorized users. The term is often used when referring to external users. However, it can apply to internal users too. Reducing the risk of data leakage is usually a top priority for information protection and DLP efforts. All the use cases listed in this section can help reduce data leakage.

This section includes common use cases that would compel an organization to implement information protection and DLP. The use cases focus primarily on Power BI, although the advantages to the organization are much broader.

Classify and label data

Organizations commonly have external or internal requirements for classifying and labeling content. The use of sensitivity labels in Power BI (and in other organizational

applications and services as well) is a key factor in meeting compliance requirements.

Once you assign a sensitivity label to content in Power BI, you're able to gain knowledge and insight about:

- Whether sensitive data is contained in a Power BI workspace.
- Whether a particular Power BI item, like a semantic model, is considered confidential.
- Who can access Power BI items that are considered sensitive.
- Who has accessed sensitive data in the Power BI service.

With end-to-end protection, sensitivity labels can (optionally) be automatically inherited from data sources. Label inheritance reduces the risk of users accessing and sharing sensitive data with unauthorized users because it wasn't labeled.

When exported from the Power BI service, sensitivity labels are retained when content is exported to supported file types. The retention of the label when content is exported is another key factor in reducing data leakage.

For more information about labeling and classifying Power BI content, see Information protection for Power BI.

Educate users

As stated previously, one aspect of protecting data involves responsible actions taken by users.

Because sensitivity labels are clearly displayed in plain text, they serve as helpful reminders to users. During the normal course of their work, labels raise awareness about how users should interact with data according to organizational guidelines and policies.

For example, when a user sees a *Highly Confidential* sensitivity label, it should prompt them to take extra care with their decisions about downloading, saving, or sharing the content with others. In this way, sensitivity labels help users responsibly handle sensitive data and reduce risk that it's shared by mistake with unauthorized users.

For more information, see Information protection for Power BI.

Detect sensitive data

The ability to detect where sensitive data is stored is another important aspect of data leakage.

When a dataset has been published to the Power BI service and it's in a Premium workspace, you can use DLP for Power BI to detect the existence of certain sensitive information types within it. This capability is helpful to find sensitive data (such as financial data or personal data) that are stored in Power BI semantic models.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

This type of DLP policy for Power BI allows security administrators to monitor and detect when unauthorized sensitive data is uploaded to the Power BI service. They can depend on alerts to act quickly. Policy tips are also used to guide content creators and owners on how to properly handle sensitive data. For more information about DLP for Power BI, see Data loss prevention for Power BI.

> **Tip**
>
> Having properly classified data allows you to correlate, analyze, and report on it. In most cases, you'll need to correlate data from multiple sources to form a complete understanding. You can capture data by using tools like the **Power BI scanner APIs** and the **Power BI activity log**. For more information about these topics, as well as audit logs in the Microsoft Purview compliance portal, see **Auditing of information protection and data loss prevention for Power BI**.

Use data encryption

Files that are classified with a sensitivity label can (optionally) include protection. When a file is protected with encryption, it reduces the risk of data leakage and oversharing. The encryption setting follows the file, regardless of device or user. Unauthorized users (internal and external to the organization) are unable to open, decrypt, or view the file contents.

> **Important**

There are trade-offs you should understand when implementing encryption. For more information, including encryption considerations, see **Information protection for Power BI**.

For more information about the types of controls you can implement to reduce data leakage, see Defender for Cloud Apps for Power BI.

Control activity in real time

To augment existing security settings in Power BI, you can implement real-time controls to reduce the risk of data leakage.

For example, you can restrict users from downloading highly sensitive data and reports from the Power BI service. This type of real-time control is helpful when someone is allowed to view content themselves, but they should be prevented from downloading and distributing it to others.

For more information about the types of controls you can implement, see Defender for Cloud Apps for Power BI.

Tip

For additional considerations related to strengthening Power BI compliance, see the **security planning** articles.

Information protection and DLP services

Many features and services related to information protection and DLP have been rebranded and now form part of Microsoft Purview. The Microsoft 365 security and compliance functionality has also become part of Microsoft Purview.

The features and services that are most pertinent for this series of articles include:

- **Microsoft Purview Information Protection** (formerly known as Microsoft Information Protection): Microsoft Purview Information Protection includes capabilities for discovering, classifying, and protecting data. A key principle is that data can be better protected once it's classified. The key building blocks for classifying data are sensitivity labels, which are described in the Information protection for Power BI article.
- **Microsoft Purview compliance portal** (formerly known as the Microsoft 365 compliance center): The portal is where you set up sensitivity labels. It's also where

you set up Power BI for DLP, which is described in the Data loss prevention for Power BI article.

- **Microsoft Purview Data Loss Prevention** (formerly known as Office 365 Data Loss Prevention): DLP activities focus primarily on reducing data leakage. By using sensitivity labels or sensitive information types, Microsoft Purview Data Loss Prevention policies help an organization locate sensitive data and protect it. Capabilities relevant to Power BI are described in the Data loss prevention for Power BI article.
- **Microsoft Defender for Cloud Apps** (formerly known as Microsoft Cloud App Security): Policies in Microsoft Defender for Cloud Apps (which are defined in a separate application) also help protect data, including real-time controls. Capabilities relevant to Power BI are described in the Defender for Cloud Apps for Power BI article.

The above list isn't exhaustive. Microsoft Purview includes a broad set of capabilities that far exceeds the scope of this series of articles. For example, the Microsoft Purview data cataloging and governance features are important; however, they're not directly in scope for this series of articles.

> **Tip**
>
> If you have questions about services, features, or licensing, contact your Microsoft account team. They're in the best position to clarify what's available for your organization.

The remainder of the information protection and DLP content is organized into the following articles:

- Organization-level information protection
- Information protection for Power BI
- Data loss prevention for Power BI
- Defender for Cloud Apps for Power BI
- Auditing of information protection and data loss prevention for Power BI

Related content

In the next article in this series, learn about getting started with information protection with organization-level planning activities for Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Organization-level information protection

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article describes the initial assessment and preparatory activities for information protection in Power BI. It's targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators need to collaborate with information security and other relevant teams.
- **Center of Excellence, IT, and BI teams**: The teams that are responsible for overseeing Power BI in the organization. They might need to collaborate with Power BI administrator, information security teams, and other relevant teams.

> **Important**
>
> Information protection and data loss prevention (DLP) is a significant organization-wide undertaking. Its scope and impact are far greater than Power BI alone. This type of initiative requires funding, prioritization, and planning. Expect to involve several cross-functional teams in your planning, usage, and oversight efforts.

Current state assessment

Before you get started with any setup activities, assess what's currently happening within your organization. It's critical to understand the extent to which information protection is currently implemented (or being planned).

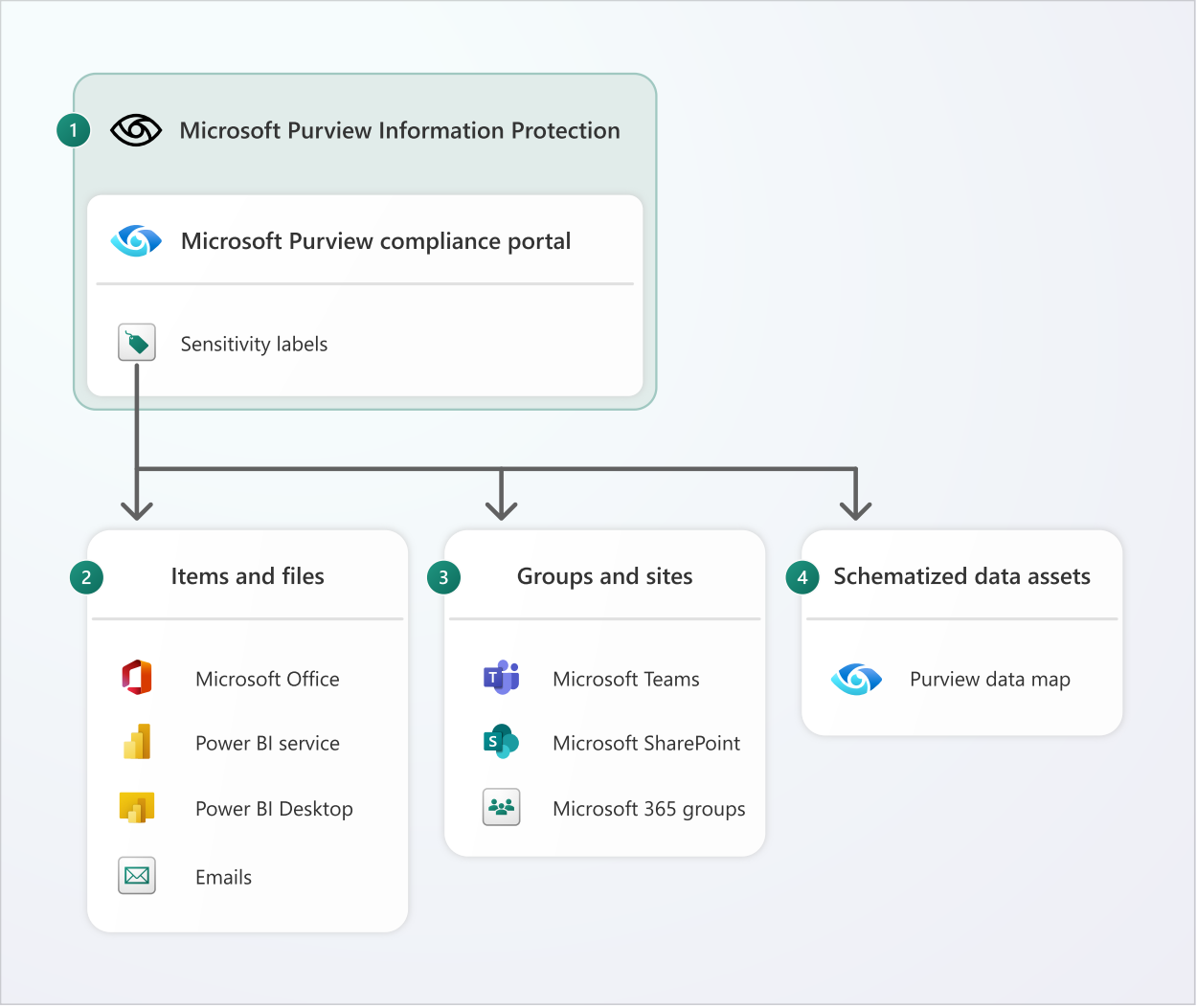
Generally, there are two cases of sensitivity label use.

- **Sensitivity labels are currently in use**: In this case, sensitivity labels are set up and used to classify Microsoft Office files. In this situation, the amount of work that will

be required to use sensitivity labels for Power BI will be significantly lower. The timeline will be shorter, and it will be more straightforward to set up quickly.

- **Sensitivity labels not yet in use**: In this case, sensitivity labels aren't used for Microsoft Office files. In this situation, an organization-wide project to implement sensitivity labels will be required. For some organizations, this project can represent a significant amount of work and a considerable time investment. That's because labels are intended to be used across the organization by various applications (rather than one application, such as Power BI).

The following diagram shows how sensitivity labels are used broadly across the organization.



The above diagram depicts the following items:

Expand table

Item	Description
1	Sensitivity labels are set up in the Microsoft Purview compliance portal.

Item	Description
2	Sensitivity labels can be applied to many types of items and files, such as Microsoft Office files, items in the Power BI service, Power BI Desktop files, and emails.
3	Sensitivity labels can be applied for Teams sites, SharePoint sites, and Microsoft 365 groups.
4	Sensitivity labels can be applied to schematized data assets that are registered in the Microsoft Purview Data Map.

In the diagram, notice that items in the Power BI service, and Power BI Desktop files, are just some of many resources that allow assigning sensitivity labels. Sensitivity labels are defined centrally in Microsoft Purview Information Protection. Once defined, the same labels are used by all supported applications throughout the organization. It's not possible to define labels for use in only one application, such as Power BI. Therefore, your planning process needs to consider a broader set of usage scenarios to define labels that can be used in multiple contexts. Because information protection is intended to be used consistently across applications and services, it's critical to begin with assessing what sensitivity labels are currently in place.

The activities for implementing sensitivity labels are described in the Information protection for Power BI article.

> **Note**
>
> Sensitivity labels are the first building block towards implementing information protection. DLP occurs after information protection is set up.

Checklist - When assessing the current state of information protection and DLP in your organization, key decisions and actions include:

- ✓ **Determine whether information protection is currently in use**: Find out what capabilities are currently enabled, how they're being used, by which applications, and by whom.
- ✓ **Identify who is currently responsible for information protection**: While assessing the current capabilities, determine who is currently responsible. Involve that team in all activities going forward.

- ✓ **Consolidate information protection projects**: If applicable, combine the information protection methods currently in use. If possible, consolidate projects and teams to gain efficiency and consistency.

Team staffing

As previously stated, many of the information protection and DLP capabilities that will be set up will have an impact across the entire organization (well beyond Power BI). That's why it's critical to assemble a team that includes all relevant people. The team will be crucial in defining the goals (described in the next section) and guiding the overall effort.

As you define roles and responsibilities for your team, we recommend that you include people who can capably translate requirements and communicate well with stakeholders.

Your team should include pertinent stakeholders involving different individuals and groups in the organization, including:

- Chief information security officer / data protection officer
- Information security / cyber security team
- Legal
- Compliance
- Risk management
- Enterprise data governance committee
- Chief data officer / chief analytics officer
- Internal audit team
- Analytics Center of Excellence (COE)
- Enterprise analytics / business intelligence (BI) team
- Data stewards and domain data owners from key business units

Your team should also include the following system administrators:

- Microsoft Purview administrator
- Microsoft 365 administrator
- Microsoft Entra ID administrator
- Defender for Cloud Apps administrator
- Power BI administrator

Tip

Expect the planning and implementation of information protection to be a collaborative effort that will take time to get right.

The task of planning and implementing information protection is usually a part-time responsibility for most people. It's typically one of many pressing priorities. Therefore, having an executive sponsor will help to clarify priorities, set deadlines, and provide strategic guidance.

Clarity on roles and responsibilities is necessary to avoid misunderstandings and delays when working with cross-functional teams across organizational boundaries.

Checklist - When putting together your information protection team, key decisions and actions include:

- ✓ **Assemble the team**: Involve all pertinent technical and non-technical stakeholders.
- ✓ **Determine who the executive sponsor is**: Ensure you're clear on who is the leader of the planning and implementation effort. Involve this person (or group) for prioritization, funding, reaching consensus, and decision-making.
- ✓ **Clarify roles and responsibilities**: Make sure everyone involved is clear on their role and responsibilities.
- ✓ **Create a communication plan**: Consider how and when you'll communicate with users throughout the organization.

Goals and requirements

It's important to consider what your goals are for implementing information protection and DLP. Different stakeholders from the team you've assembled are likely to have different viewpoints and areas of concern.

At this point, we recommend that you focus on the strategic goals. If your team has started by defining implementation level details, we suggest that you step back and define the strategic goals. Well-defined strategic goals will help you to deliver a smoother implementation.

Your information protection and DLP requirements might include the following goals.

- **Self-service user enablement**: Allow self-service BI content creators and owners to collaborate, share, and be as productive as possible–all within the guardrails

established by the governance team. The goal is to balance self-service BI with centralized BI and make it easy for self-service users to do the right thing, without negatively impacting their productivity.

- **Data culture that values protecting trusted data**: Implement information protection in a way that's low friction and doesn't get in the way of user productivity. When implemented in a balanced way, users are far more likely to work within your systems than around them. User education and user support are essential.
- **Risk reduction**: Protect the organization by reducing its risks. Risk reduction goals often include minimizing the possibility of data leakage outside of the organization and protecting data against unauthorized access.
- **Compliance**: Support compliance efforts for industry, regional, and governmental regulations. Additionally, your organization might also have internal governance and security requirements that are deemed critical.
- **Auditability and awareness**: Understand where the sensitive data is located throughout the organization and who's using it.

Be aware that an initiative to introduce information protection is complementary to other related approaches that involve security and privacy. Coordinate information protection initiatives with other efforts, such as:

- Access roles, permissions, sharing, and row-level security (RLS) for Power BI content
- Data residency requirements
- Network security requirements
- Data encryption requirements
- Data cataloging initiatives

For more information about securing content in Power BI, see the security planning articles.

Checklist - When considering your information protection goals, key decisions and actions include:

- ✓ **Identify applicable data privacy regulations and risks**: Ensure that your team is aware of the data privacy regulations that your organization is subject to for your industry or geographic region. If necessary, conduct a data privacy risk assessment.

- ✓ **Discuss and clarify your goals**: Have initial discussions with relevant stakeholders and interested people. Make sure you're clear on your information protection strategic goals. Ensure that you can translate these goals into business requirements.
- ✓ **Approve, document, and prioritize your goals**: Ensure that your strategic goals are documented and prioritized. When you need to make complex decisions, prioritize, or make trade-offs, refer to these goals.
- ✓ **Verify and document regulatory and business requirements**: Ensure that all regulatory and business requirements for data privacy are documented. Refer to them for prioritization and compliance needs.
- ✓ **Begin creating a plan**: Start building a project plan by using the prioritized strategic goals and documented requirements.

Related content

In the next article in this series, learn about labeling and classification of data assets for use with Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Information protection for Power BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article describes the planning activities related to implementing information protection in Power BI. It's targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators need to collaborate with information security and other relevant teams.
- **Center of Excellence, IT, and BI teams**: Others who are responsible for overseeing Power BI in the organization. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.

> **Important**
>
> Information protection and data loss prevention (DLP) is a significant organization-wide undertaking. Its scope and impact are far greater than Power BI alone. This type of initiative requires funding, prioritization, and planning. Expect to involve several cross-functional teams in your planning, usage, and oversight efforts.

Labeling and classification activities extend beyond Power BI and even data assets. The decisions discussed in this article apply to assets for the entire organization, including files and emails, and not just to Power BI. This article introduces topics that apply to labeling and classification in general, because making the right organizational decisions is critical for the success of data loss prevention in Power BI.

This article also includes introductory guidance about defining a sensitivity label structure. Technically, the sensitivity label structure is a prerequisite for implementation of sensitivity labels in Power BI. The purpose of including some basic information in this article is to help you understand what's involved. It's crucial that you collaborate with IT for planning and implementing information protection in the organization.

Purpose of sensitivity labels

The use of Microsoft Purview Information Protection sensitivity labels is about classifying content. Think of a sensitivity label like a tag that you apply to an item, file, site, or data asset.

There are many advantages to using information protection. Classifying and labeling content helps organizations to:

- Understand where sensitive data resides.
- Track external and internal compliance requirements.
- Protect content from unauthorized users.
- Educate users on how to responsibly handle data.
- Implement real-time controls to reduce the risk of data leakage.

For more use cases for information protection, see Information protection and DLP (Common use cases).

> **Tip**
>
> It helps to remember that Microsoft Purview Information Protection is the product. Sensitivity labels are a specific feature of that product.

A sensitivity label is a brief description in clear text. Conceptually, you can think of a sensitivity label like a tag. Only one label can be assigned to each item (like a Power BI semantic model in the Power BI service) or each file (like a Power BI Desktop file).

A label has the following purposes.

- **Classification**: It provides a classification for describing the sensitivity level.
- **User education and awareness**: It helps users understand how to appropriately work with the content.
- **Policies**: It forms the basis for applying and enforcing policies and DLP.

Prerequisites for Power BI information protection

By now, you should have completed the organization-level planning steps that are described in the Organization-level information protection planning article. Before proceeding, you should have clarity on:

- **Current state**: The current state of information protection in your organization. You should have an understanding whether sensitivity labels are already in use for Microsoft Office files. In this case, the scope of work to add Power BI is much smaller than if you're bringing information protection to the organization for the first time.
- **Goals and requirements** : The strategic goals for implementing information protection in your organization. Understanding the goals and requirements will serve as a guide for your implementation efforts.

If information protection isn't in use by your organization, the remainder of this section provides information to help you collaborate with others to introduce information protection to your organization.

If information protection is actively in use within your organization, we recommend that you use this article to verify that the prerequisites are met. If sensitivity labels are actively in use, then most (or all) activities in rollout phases 1-4 (in the next section) will already be complete.

Rollout phases

We recommend that you plan to enact a gradual rollout plan for implementing and testing information protection. The objective for a gradual rollout plan is to set yourself up to learn, adjust, and iterate as you go. The advantage is that fewer users are impacted during the early stages (when changes are more likely), until information protection is eventually rolled out to all users in the organization.

Introducing information protection is a significant undertaking. As described in the Organization-level information protection planning article, if your organization has already implemented information protection for Microsoft Office documents, many of these tasks will already be complete.

This section provides an overview of the phases that we recommend you include in your gradual rollout plan. It should give you a sense for what to expect. The remainder of this article describes other decision-making criteria for the key aspects that affect Power BI most directly.

Phase 1: Plan, decide, prepare

In the first phase, focus on planning, decision-making, and preparatory activities. Most of the remainder of this article focuses this first phase.

As early as possible, clarify where the initial testing will occur. That choice will impact where you'll initially set up, publish, and test. For the initial testing, you can use a non-production tenant (if you have access to one).

> **Tip**
>
> Most organizations have access to one tenant, so it can be challenging to explore new features in an isolated way. For those organizations that have a separate development or test tenant, we recommend that you use it for the initial testing phase. For more information about managing tenants and how to create a trial tenant to test out new features, see **Tenant setup**.

Phase 2: Set up supporting user resources

The second phase includes steps to set up the resources for supporting users. Resources include your data classification and protection policy and the custom help page.

It's important to have some of the user documentation published early. It's also important to have the user support team prepared early.

Phase 3: Set up labels and publish

The third phase focuses on defining sensitivity labels. When all the decisions have been made, the setup isn't difficult or time-consuming. Sensitivity labels are set up in the Microsoft Purview compliance portal in the Microsoft 365 admin center.

Phase 4: Publish label policy

Before a label can be used, you must publish it as part of a label policy. Label policies allow certain users to use a label. Label policies are published in the Microsoft Purview compliance portal in the Microsoft 365 admin center.

> **Note**
>
> Everything up until this point is a prerequisite for implementing information protection for Power BI.

Phase 5: Enable Fabric tenant settings

There are several information protection tenant settings in the Fabric admin portal. They're required to enable information protection in the Power BI service.

> **ⓘ Important**
>
> You should set the tenant settings *after* you've set up and published the labels in the Microsoft Purview compliance portal.

Phase 6: Initial testing

In the sixth phase, you perform initial tests to verify that everything behaves as expected. For initial testing purposes, you should publish the label policy only for the implementation team.

During this phase, be certain to test:

- Microsoft Office files
- Power BI items in the Power BI service
- Power BI Desktop files
- Exporting files from the Power BI service
- Other scopes included in the configuration, such as Teams sites or SharePoint

Be sure to check the functionality and user experience by using a web browser and also mobile devices that are commonly used. Update your user documentation accordingly.

> **ⓘ Important**
>
> Even if only a few members of the team are authorized to set a sensitivity label, all users will be able to see labels that are assigned to content. If you're using your production tenant, users might wonder why they see labels assigned to items in a workspace in the Power BI service. Be ready to support and respond to user questions.

Phase 7: Gather user feedback

The goal for this phase is to obtain feedback from a small group of key users. The feedback should identify areas of confusion, gaps in the label structure, or technical issues. You might also find reasons to improve the user documentation.

To this end, you should publish (or republish) the label policy to a small subset of users who are willing to provide feedback.

> **Tip**
>
> Be sure to factor sufficient time into your project plan. For labels and label policy settings, the product documentation recommends **allowing 24 hours** for the changes to take effect. This time is required to ensure all changes propagate through to related services.

Phase 8: Release iteratively

The implementation phase is usually an iterative process.

Often, the initial objective is to get to a state where all Power BI content has a sensitivity label assigned. To achieve this objective, you might introduce a mandatory label policy or a default label policy. You might also use the information protection admin APIs to programmatically set or remove sensitivity labels.

You can gradually include more groups of users until the entire organization is included. This process involves republishing each label policy to increasingly larger groups of users.

Throughout this process, be sure to prioritize providing guidance, communications, and training to your users so they'll understand the process and what's expected of them.

Phase 9: Monitor, audit, adjust, integrate

There are other steps to do after the initial rollout. You should have a primary team to monitor information protection activities and tune them over time. As labels are applied, you'll be able to assess their usefulness and identify areas for adjustments.

There are many aspects to auditing information protection. For more information, see Auditing of information protection and data loss prevention for Power BI.

The investments you make in setting up information protection can be used in DLP policies for Power BI, which are set up in the Microsoft Purview compliance portal. For more information, including a description of DLP capabilities, see Data loss prevention for Power BI.

Information protection can also be used to create policies in Microsoft Defender for Cloud Apps. For more information, including a description of capabilities that you might find helpful, see Defender for Cloud Apps for Power BI.

Checklist - When preparing your information protection rollout phases, the key decisions and actions include:

- ✓ **Create a gradual rollout plan**: Define the phases for your rollout plan. Clarify what the specific objectives are for each phase.
- ✓ **Identify where to do testing**: Determine where the initial testing can be done. To minimize the impact on users, use a non-production tenant, if possible.
- ✓ **Create a project plan**: Build a project plan that includes all the key activities, estimated timeline, and who will be responsible.

Sensitivity label structure

The sensitivity label structure is a prerequisite for implementing sensitivity labels in Power BI. This section includes some basic information to help you understand what's involved if you're involved in creating the label structure.

This section isn't an exhaustive list of all possible sensitivity label considerations for all possible applications. Instead, its focus is on considerations and activities that directly affect the classification of Power BI content. Ensure that you work with other stakeholders and system administrators to make decisions that work well for all applications and use cases.

The foundation for implementing information protection begins with a set of sensitivity labels. The end goal is to create a set of sensitivity labels that are clear and straightforward for users to work with.

The sensitivity label structure that's used in an organization represents a *label taxonomy*. It's also commonly referred to as a *data classification taxonomy* because the goal is to classify data. Sometimes it's referred to as a *schema definition*.

There isn't a standard, or built-in, set of labels. Each organization must define and customize a set of labels to suit their needs. The process of arriving at the right set of labels involves extensive collaboration. It requires thoughtful planning to ensure that the labels will meet goals and requirements. Remember, labels will be applied to more than just Power BI content.

> **Tip**

Most organizations start out by assigning labels to Microsoft Office files. They then evolve to classifying other content, such as Power BI items and files.

A label structure includes:

- **Labels**: Labels form a hierarchy. Each label indicates the level of sensitivity for an item, file, or data asset. We recommend that you create between three and seven labels. The labels should rarely change.
- **Sub-labels**: Sub-labels indicate variations in protection or scope within a specific label. By including them in different label policies, you can scope sub-labels to a certain set of users or to users involved with a specific project.

> **Tip**
>
> While sub-labels offer flexibility, they should be used in moderation only to satisfy critical requirements. Creating too many sub-labels results in increased management. They can also overwhelm users with too many options.

Labels form a hierarchy, beginning with the least sensitive classification to the most sensitive classification.

Sometimes Power BI content contains data that spans multiple labels. For example, a semantic model could contain product inventory information (*General Internal Use*) and the current quarter sales figures *(Restricted*). When choosing which label to assign to the Power BI semantic model, users should be taught to apply the most restrictive label.

> **Tip**
>
> The next section describes the *Data classification and protection policy* that can provide users with guidance on when to use each label.

> **Important**
>
> Assigning a label or a sub-label doesn't directly affect access to Power BI content in the Power BI service. Instead, the label provides a useful category that can guide user behavior. Data loss prevention policies can also be based on the assigned label. However, nothing changes for how access to Power BI content is managed, except when a file is encrypted. For more information, see **Use of encryption protection**.

Be deliberate with the labels you create because it's challenging to remove or delete a label once you've progressed beyond the initial testing phase. Because sub-labels can (optionally) be used for a particular set of users, they can change more often than labels.

Here are some best practices for defining a label structure.

- **Use intuitive, unambiguous, terms**: Clarity is important to ensure that users know what to choose when classifying their data. For example, having a *Top Secret* label and a *Highly Confidential* label is ambiguous.
- **Create a logical hierarchical order**: The order of the labels is crucial to making everything work well. Remember that the last label in the list is the most sensitive. The hierarchical order, in combination with well-selected terms, should be logical and intuitive for users to work with. A clear hierarchy will also make policies easier to create and maintain.
- **Create just a few labels that apply across the organization**: Having too many labels for users to choose from will be confusing. It will also lead to less accurate label selection. We recommend that you create just a few labels for the initial set.
- **Use meaningful, generic names**: Avoid using industry jargon or acronyms in your label names. For example, rather than creating a label named *Personally Identifiable Information*, use names like *Highly Restricted* or *Highly Confidential* instead.
- **Use terms that are easily localized into other languages**: For global organizations with operations in multiple countries/regions, it's important to choose label terms that won't be confusing or ambiguous when they're translated into other languages.

> **Tip**
>
> If you find yourself planning for many labels that are highly specific, step back and reassess your approach. Complexity can lead to user confusion, reduced adoption, and less effective information protection. We recommend that you begin with an initial set of labels (or use what you already have). After you've gained more experience, cautiously expand the set of labels by adding more specific ones when needed.

Checklist - When planning your sensitivity label structure, key decisions and actions include:

- ✓ **Define an initial set of sensitivity labels**: Create an initial set of between three and seven sensitivity labels. Ensure that they have broad use for a wide range of content. Plan to iterate on the initial list as you finalize the data classification and protection policy.
- ✓ **Determine whether you need sub-labels**: Decide whether there's a need to use sub-labels for any of the labels.
- ✓ **Verify localization of label terms**: If labels will be translated into other languages, have native speakers confirm that localized labels convey the intended meaning.

Sensitivity label scope

A sensitivity label scope limits the use of a label. While you can't specify Power BI directly, you can apply labels to various scopes. Possible scopes include:

- Items (such as items published to the Power BI service, and files and emails)
- Groups and sites (such as a Teams channel or a SharePoint site)
- Schematized data assets (supported sources that are registered in the Purview Data Map)

> ⓘ **Important**
>
> It's not possible to define a sensitivity label with a scope of Power BI only. While there are some settings that apply specifically to Power BI, scope isn't one of them. The *items* scope is used for the Power BI service. Sensitivity labels are handled differently from DLP policies, which are described in the **Data loss prevention for Power BI planning** article, in that some types of DLP policies can be defined specifically for Power BI. If you intend to use sensitivity label inheritance from data sources in Power BI, there are **specific requirements** for the label scope.

Events related to sensitivity labels are recorded in the activity explorer. Logged details of these events will be significantly richer when the scope is broader. You'll also be better prepared to protect data across a broader spectrum of applications and services.

When defining the initial set of sensitivity labels, consider making the initial set of labels available to all scopes. That's because it can become confusing for users when they see different labels in different applications and services. Over time, however, you might discover use cases for more specific sub-labels. However, it's safer to start with a consistent and simple set of initial labels.

The label scope is set in the Microsoft Purview compliance portal when the label is set up.

Checklist - When planning for the label scope, key decisions and actions include:

- ✓ **Decide the label scope**: Discuss and decide whether each of your initial labels will be applied to all scopes.
- ✓ **Review all prerequisites**: Investigate prerequisites and necessary setup steps that will be required for each scope that you intend to use.

Use of encryption protection

There are multiple options for protection with a sensitivity label.

- **Encryption**: Encryption settings pertain to files or emails. For example, a Power BI Desktop file can be encrypted.
- **Markings**: Refers to headers, footers, and watermarks. Markings are useful for Microsoft Office files, but they aren't shown on Power BI content.

> **Tip**
>
> Usually when someone refers to a label as *protected*, they're referring to encryption. It might be sufficient that only higher-level labels, like *Restricted* and *Highly Restricted*, are encrypted.

Encryption is a way to cryptographically encode information. Encryption has several key advantages.

- Only authorized users (for example, internal users within your organization) can open, decrypt, and read a protected file.
- Encryption remains with a protected file, even when the file is sent outside the organization or is renamed.
- The encryption setting is obtained from the original labeled content. Consider a report in the Power BI service has a sensitivity label of *Highly Restricted*. If it's exported to a supported export path, the label remains intact, and encryption is applied on the exported file.

The Azure Rights Management Service (Azure RMS) is used for file protection with encryption. There are some important prerequisites that must be met in order to use Azure RMS encryption.

> **ⓘ Important**
>
> There's a limitation to consider: An offline user (without an internet connection) can't open an encrypted Power BI Desktop file (or other type of Azure RMS-protected file). That's because Azure RMS must synchronously verify that a user is authorized to open, decrypt, and view the file contents.

Encrypted labels are handled differently depending on where the user is working.

- **In the Power BI service**: The encryption setting doesn't have a direct effect on user access in the Power BI service. Standard Power BI permissions (such as workspace roles, app permissions, or sharing permissions) control user access in the Power BI service. Sensitivity labels don't affect access to content within the Power BI service.
- **Power BI Desktop files**: An encrypted label can be assigned to a Power BI Desktop file. The label is also retained when it's exported from the Power BI service. Only authorized users will be able to open, decrypt, and view the file.
- **Exported files**: Microsoft Excel, Microsoft PowerPoint, and PDF files exported from the Power BI service retain their sensitivity label including encryption protection. For supported file formats, only authorized users will be able to open, decrypt, and view the file.

> **ⓘ Important**
>
> It's critical that users understand the distinctions between the Power BI service and files, which are easily confused. We recommend that you provide a FAQs document, together with examples, to help users understand the differences.

To open a protected Power BI Desktop file, or an exported file, a user must meet the following criteria.

- **Internet connectivity**: The user must be connected to the internet. An active internet connection is required to communicate with Azure RMS.
- **RMS permissions**: The user must have RMS permissions, which are defined within the label (rather than within the label policy). RMS permissions allow authorized users to decrypt, open, and view supported file formats.
- **Allowed user**: Users or groups must be specified in the label policy. Typically, assigning authorized users is only required for content creators and owners so they can apply labels. However, when using encryption protection there's another requirement. Each user that needs to open a protected file must be specified in the

label policy. This requirement means that information protection licensing might be required for more users.

> **Tip**
>
> The **Allow workspace administrators to override automatically applied sensitivity labels** tenant setting enables workspace administrators to change a label that was automatically applied, even if protection (encryption) is enabled for the label. This capability is particularly helpful when a label was automatically assigned or inherited but the workspace administrator isn't an authorized user.

Label protection is set in the Microsoft Purview compliance portal when you set up the label.

Checklist - When planning for the use of label encryption, key decisions and actions include:

- ✓ **Decide which labels should be encrypted**: For each sensitivity label, decide whether it should be encrypted (protected). Carefully consider the limitations that are involved.
- ✓ **Identify the RMS permissions for each label**: Determine what the user permissions will be for accessing and interacting with encrypted files. Create a mapping of users and groups for each sensitivity label to help with the planning process.
- ✓ **Review and address RMS encryption prerequisites**: Ensure that technical prerequisites for using Azure RMS encryption are met.
- ✓ **Plan to conduct thorough testing of encryption**: Due to the differences between Office files and Power BI files, ensure that you commit to a thorough testing phase.
- ✓ **Include in user documentation and training**: Ensure that you include guidance in your documentation and training about what users should expect for files that are assigned a sensitivity label that's encrypted.
- ✓ **Conduct knowledge transfer with support**: Make specific plans to conduct knowledge transfer sessions with the support team. Due to the complexity of encryption, they'll likely get questions from users.

Inheritance of labels from data sources

When importing data from supported data sources (such as Azure Synapse Analytics, Azure SQL Database, or an Excel file), a Power BI semantic model can, optionally, inherit the sensitivity label applied to the source data. Inheritance helps to:

- Promote consistency of labeling.
- Reduce user effort when assigning labels.
- Reduce the risk of users accessing and sharing sensitive data with unauthorized users because it wasn't labeled.

> **Tip**
>
> There are two types of inheritance for sensitivity labels. ***Downstream inheritance*** refers to downstream items (like reports) that automatically inherit a label from its Power BI semantic model. However, the focus of this section is on **_upstream inheritance**. Upstream inheritance refers to a Power BI semantic model that inherits a label from a data source that's upstream from the semantic model.

Consider an example where the organization's working definition for the sensitivity label of *Highly Restricted* includes financial account numbers. Because financial account numbers are stored in an Azure SQL Database, the *Highly Restricted* sensitivity label has been assigned to that source. When data from the Azure SQL Database is imported to Power BI, the intent is for the semantic model to inherit the label.

You can assign sensitivity labels to a supported data source in different ways.

- **Data discovery and classification**: You can scan a supported database to identify columns that could contain sensitive data. Based on the scan results, you can apply some or all the label recommendations. Data Discovery & Classification is supported for databases such as Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. SQL Data Discovery & Classification is supported for on-premises SQL Server databases.
- **Manual assignments**: You can assign a sensitivity label to an Excel file. You might also manually assign labels to database columns in Azure SQL Database or SQL Server.
- **Auto-labeling in Microsoft Purview**: Sensitivity labels can be applied to supported data sources that are registered as assets in the Microsoft Purview Data Map.

> **⚠ Warning**
>
> The details for how to assign sensitivity labels to a data source are out of scope for this article. The technical capabilities are evolving with respect to what's supported for inheritance in Power BI. We recommend that you conduct a technical proof of

concept to verify your goals, ease of use, and whether the capabilities meet your requirements.

Inheritance will happen only when you enable the **Apply sensitivity labels from data sources to their data in Power BI** tenant setting. For more information about tenant settings, see the Fabric tenant settings section later in this article.

> **Tip**
>
> You'll need to become familiar with the **inheritance behavior**. Be sure to include various circumstances in your test plan.

Checklist - When planning for the inheritance of labels from data sources, key decisions and actions include:

- ✓ **Decide whether Power BI should inherit labels from data sources**: Decide whether Power BI should inherit these labels. Plan to enable the tenant setting to allow this capability.
- ✓ **Review technical prerequisites**: Determine whether you need to take extra steps to assign sensitivity labels to data sources.
- ✓ **Test label inheritance functionality**: Complete a technical proof of concept to test how inheritance works. Verify that the feature works as you expect in various circumstances.
- ✓ **Include in user documentation**: Ensure that information about label inheritance is added to guidance provided to users. Include realistic examples in the user documentation.

Published label policies

After you define a sensitivity label, the label can be added to one or more *label policies*. A label policy is how you publish the label so it can be used. It defines which labels can be used by which set of authorized users. There are other settings as well, such as the default label and mandatory label.

Using multiple label policies can be helpful when you need to target different sets of users. You can define a sensitivity label once and then include in one or more label policies.

> **Tip**
>
> A sensitivity label isn't available for use until a label policy that contains the label is published in the Microsoft Purview compliance portal.

Authorized users and groups

When you create a label policy, one or more users or groups must be selected. The label policy determines which users can use the label. It allows users to assign that label to specific content, such as a Power BI Desktop file, an Excel file, or an item published to the Power BI service.

We recommend that you keep the authorized users and groups as simple as possible. A good rule of thumb is for the primary labels to be published for all users. Sometimes it's appropriate for a sub-label to be assigned, or scoped, to a subset of users.

We recommend that you assign groups instead of individuals whenever possible. The use of groups simplifies management of the policy and reduces how often it needs to be republished,

> **⚠ Warning**
>
> Authorized users and groups for a label are different from the users assigned to Azure RMS for a protected (encrypted) label. If a user is having issues opening an encrypted file, investigate the encryption permissions for specific users and groups (which are set up within the label configuration, rather than within the label policy). In most situations, we recommend that you assign the same users to both. This consistency will avoid confusion and reduce support tickets.

Authorized users and groups are set in the Microsoft Purview compliance portal when the label policy is published.

Checklist - When planning for authorized users and groups in your label policy, key decisions and actions include:

- ✓ **Determine which labels apply to all users**: Discuss and decide which sensitivity labels should be available for use by all users.

- ✓ **Determine which sub-labels apply to a subset of users**: Discuss and decide whether there are any sub-labels that will be available for use only by a specific set of users or groups.
- ✓ **Identify whether any new groups are needed**: Determine whether any new Microsoft Entra ID groups will need to be created to support the authorized users and groups.
- ✓ **Create a planning document**: If the mapping of authorized users to sensitivity labels is complicated, create a mapping of users and groups for each label policy.

Default label for Power BI content

When creating a label policy, you can choose a default label. For example, the *General Internal Use* label could be set as the default label. This setting will affect new Power BI items created in either Power BI Desktop or the Power BI service.

You can set up the default label in the label policy specifically for Power BI content, which is separate from other items. Most information protection decisions and settings apply more broadly. However, the default label setting (and the mandatory label setting that's described next) applies only to Power BI.

> **Tip**
>
> While you can set different default labels (for Power BI and non-Power BI content), consider whether that's the best option for users.

It's important to understand that a new default label policy will apply to content created, or edited, *after* the label policy is published. It won't retroactively assign the default label to existing content. Your Power BI administrator can use the information protection APIs to set sensitivity labels in bulk to ensure that existing content is assigned to a default sensitivity label.

The default label options are set in the Microsoft Purview compliance portal when the label policy is published.

Checklist - When planning whether a default label for Power BI content will be applied, key decisions and actions include:

- ✓ **Decide whether a default label will be specified**: Discuss and decide whether a default label is appropriate. If so, determine which label is best suited as the default.
- ✓ **Include in user documentation**: If necessary, ensure information about the default label is mentioned in the guidance provided for users. The goal is for users to understand how to determine whether the default label is appropriate, or whether it should be changed.

Mandatory labeling Power BI content

Data classification is a common regulatory requirement. To meet this requirement, you can choose to require users to label all Power BI content. This mandatory labeling requirement takes effect when users create or edit Power BI content.

You might choose to implement mandatory labels, default labels (described in the previous section), or both. You should consider the following points.

- A mandatory label policy ensures the label won't be empty
- A mandatory label policy requires users to choose what the label should be
- A mandatory label policy prevents users from removing a label entirely
- A default label policy is less intrusive for users because it doesn't require them to take action
- A default label policy can result in content that's mis-labeled since a user didn't expressly make the choice
- Enabling both a default label policy and a mandatory label policy can provide complementary benefits

> **Tip**
>
> If you choose to implement mandatory labels, we recommend that you also implement default labels.

You can set up the mandatory label policy specifically for Power BI content. Most information protection settings apply more broadly. However, the mandatory label setting (and default label setting) applies specifically to Power BI.

> **Tip**
>
> A mandatory label policy isn't applicable to service principals or APIs.

The mandatory labeling options are set in the Microsoft Purview compliance portal when the label policy is published.

Checklist - When planning whether mandatory labeling of Power BI content will be required, key decisions and actions include:

- ✓ **Decide whether labels will be mandatory**: Discuss and decide whether mandatory labels are necessary for compliance reasons.
- ✓ **Include in user documentation**: If necessary, ensure that information about mandatory labels is added to guidance provided for users. The goal is for users to understand what to expect.

Licensing requirements

Specific licenses must be in place to work with sensitivity labels.

A Microsoft Purview Information Protection license is required for:

- **Administrators**: The administrators who will set up, manage, and oversee labels.
- **Users**: The content creators and owners who will be responsible for applying labels to content. Users also includes those who need to decrypt, open, and view protected (encrypted) files.

You might already have these capabilities because they're included in license suites, such as Microsoft 365 E5. Alternatively, Microsoft 365 E5 Compliance capabilities can be purchased as a standalone license.

A Power BI Pro or Premium Per User (PPU) license is also required for users who will apply and manage sensitivity labels in the Power BI service or Power BI Desktop.

> **Tip**
>
> If you need clarifications about licensing requirements, talk to your Microsoft account team. Be aware that the Microsoft 365 E5 Compliance license includes additional capabilities that are out of scope for this article.

Checklist - When evaluating licensing requirements for sensitivity labels, key decisions and actions include:

- ✓ **Review product licensing requirements**: Ensure that you review all the licensing requirements.
- ✓ **Review user licensing requirements**: Verify that all users you expect to assign labels have Power BI Pro or PPU licenses.
- ✓ **Procure additional licenses**: If applicable, purchase more licenses to unlock the functionality that you intend to use.
- ✓ **Assign licenses**: Assign a license to each user who will assign, update, or manage sensitivity labels. Assign a license to each user who will interact with encrypted files.

Fabric tenant settings

There are several Fabric tenant settings that are related to information protection.

> ⓘ **Important**
>
> The Fabric tenant settings for information protection shouldn't be set until after all prerequisites are met. The labels and the label policies should be set up and published in the Microsoft Purview compliance portal. Until this time, you're still in the decision-making process. Before setting the tenant settings, you should first determine a process for how to test the functionality with a subset of users. Then you can decide how to do a gradual roll out.

Users who can apply labels

You should decide who will be allowed to apply sensitivity labels to Power BI content. This decision will determine how you set the **Allow users to apply sensitivity labels for content** tenant setting.

It's typically the content creator or owner who assigns the label during their normal workflow. The most straightforward approach is to enable this tenant setting, which allows all Power BI users to apply labels. In this case, standard workspace roles will determine who can edit items in the Power BI service (including applying a label). You can use the activity log to track when a user assigns or changes a label.

Labels from data sources

You should decide whether you want sensitivity labels to be inherited from supported data sources that are upstream from Power BI. For example, if columns in an Azure SQL Database have been defined with the *Highly Restricted* sensitivity label, then a Power BI semantic model that imports data from that source will inherit that label.

If you decide to enable inheritance from upstream data sources, set the **Apply sensitivity labels from data sources to their data in Power BI** tenant setting. We recommend that you plan to enable inheritance of data source labels to promote consistency and reduce effort.

Labels for downstream content

You should decide whether sensitivity labels should be inherited by downstream content. For example, if a Power BI semantic model has a sensitivity label of *Highly Restricted*, then all downstream reports will inherit this label from the semantic model.

If you decide to enable inheritance by downstream content, set the **Automatically apply sensitivity labels to downstream content** tenant setting. We recommend that you plan to enable inheritance by downstream content to promote consistency and reduce effort.

Workspace administrator overrides

This setting applies for labels that were applied automatically (such as when default labels are applied, or when labels are automatically inherited). When a label has protection settings, Power BI allows only authorized users to change the label. This setting enables workspace administrators to change a label that was applied automatically, even if there are protection settings on the label.

If you decide to allow label updates, set the **Allow workspace administrators to override automatically applied sensitivity labels** tenant setting. This setting applies to the entire organization (not individual groups). It allows workspace administrators to change labels that were automatically applied.

We recommend that you consider allowing Power BI workspace administrators to update labels. You can use the activity log to track when they assign or change a label.

Disallow sharing of protected content

You should decide whether protected (encrypted) content can be shared with everyone in your organization.

If you decide to disallow sharing of protected content, set the **Restrict content with protected labels from being shared via link with everyone in your organization** tenant setting. This setting applies to the entire organization (not individual groups).

We strongly recommend that you plan to enable this tenant setting to disallow sharing of protected content. When enabled, it disallows sharing operations with the entire organization for more sensitive content (defined by the labels that have encryption defined). By enabling this setting, you'll reduce the possibility of data leakage.

> **Important**
>
> There's a similar **tenant setting** named *Allow shareable links to grant access to everyone in your organization*. Although it has a similar name, its purpose is different. It defines which groups can create a sharing link for the entire organization, regardless of the sensitivity label. In most cases, we recommend this capability be limited in your organization. For more information, see the **Report consumer security planning** article.

Supported export file types

In the Fabric admin portal, there are many export and sharing tenant settings. In most circumstances, we recommend that the ability to export data be available to all (or most) users so as not to limit user productivity.

However, highly regulated industries could have a requirement to restrict exports when information protection can't be enforced for the output format. A sensitivity label that's applied in the Power BI service follows the content when it's exported to a supported file path. It includes Excel, PowerPoint, PDF, and Power BI Desktop files. Since the sensitivity label stays with the exported file, protection benefits (encryption that prevents unauthorized users from opening the file) are retained for these supported file formats.

> **Warning**
>
> When exporting from Power BI Desktop to a PDF file, the protection is not retained for the exported file. We recommend that you educate your content creators to export from the Power BI service to achieve maximum information protection.

Not all export formats support information protection. Unsupported formats, such as .csv, .xml, .mhtml, or .png files (available when using the ExportToFile API) might be disabled in the Fabric tenant settings.

> **Tip**
>
> We recommend that you restrict exporting capabilities only when you must meet specific regulatory requirements. In typical scenarios, we recommend that you use the Power BI activity log to identify which users are performing exports. You can then teach these users about more efficient and secure alternatives.

Checklist - When planning how tenant settings will be set up in the Fabric admin portal, key decisions and actions include:

- ✓ **Decide which users can apply sensitivity labels**: Discuss and decide whether sensitivity labels can be assigned to Power BI content by all users (based on standard Power BI security) or only by certain groups of users.
- ✓ **Determine whether labels should be inherited from upstream data sources**: Discuss and decide whether labels from data sources should be automatically applied to Power BI content that uses the data source.
- ✓ **Determine whether labels should be inherited by downstream items**: Discuss and decide whether labels assigned to existing Power BI semantic models should be automatically applied to related content.
- ✓ **Decide whether Power BI workspace administrators can override labels**: Discuss and decide whether it's appropriate for workspace administrators to be able to change protected labels that were automatically assigned.
- ✓ **Determine whether protected content can be shared with the entire organization**: Discuss and decide whether sharing links for "people in your organization" can be created when a protected (encrypted) label has been assigned to an item in the Power BI service.
- ✓ **Decide which export formats are enabled**: Identify regulatory requirements that will affect which export formats are available. Discuss and decide whether users will be able to use all export formats in the Power BI service. Determine whether certain formats need to be disabled in the tenant settings when the export format doesn't support information protection.

Data classification and protection policy

Setting up your label structure and publishing your label policies are necessary first steps. However, there's more to do to help your organization be successful with

classifying and protecting data. It's critical that you provide guidance to users about what can and can't be done with content that's been assigned to a certain label. That's where you'll find a *data classification and protection policy* is helpful. You can think of it as your label guidelines.

> ⓘ **Note**
>
> A data classification and protection policy is an internal governance policy. You might choose to call it something different. What matters is that it's documentation that you create and provide to your users so they know *how* to use the labels effectively. Since the label policy is a specific page in the Microsoft Purview compliance portal, try to avoid calling your internal governance policy by the same name.

We recommend that you iteratively create your data classification and protection policy while you're in the decision-making process. It will mean everything is clearly defined when it's time to set up the sensitivity labels.

Here are some of the key pieces of information that you might include in your data classification and protection policy.

- **Description of the label**: Beyond the label name, provide a full description of the label. The description should be clear yet brief. Here are some example descriptions:
  - *General Internal Use* - for private, internal, business data
  - *Restricted* - for sensitive business data that would cause harm if compromised or is subject to regulatory or compliance requirements
- **Examples**: Provide examples to help explain when to use the label. Here are some examples:
  - *General Internal Use* - for most internal communications, non-sensitive support data, survey responses, reviews, ratings, and imprecise location data
  - *Restricted* - for personally identifiable information (PII) data such as name, address, phone, email, government ID number, race, or ethnicity. Includes vendor and partner contracts, non-public financial data, employee, and human resources (HR) data. Also includes proprietary information, intellectual property, and precise location data.
- **Label required**: Describes whether assigning a label is mandatory for all new and changed content.
- **Default label**: Describes whether this label is a default label that's automatically applied to new content.

- **Access restrictions**: Additional information that clarifies whether internal and/or external users are permitted to see content assigned to this label. Here are some examples:
  - All users, including internal users, external users, and third parties with active non-disclosure agreements (NDAs) in place can access this information.
  - Internal users only can access this information. No partners, vendors, contractors, or third parties, regardless of NDA or confidentiality agreement status.
  - Internal access to information is based on job role authorization.
- **Encryption requirements**: Describes whether data is required to be encrypted at-rest and in-transit. This information will correlate to how the sensitivity label is set up and will affect the protection policies that can be implemented for file (RMS) encryption.
- **Downloads allowed and/or offline access**: Describes whether offline access is permitted. It can also define whether downloads are permitted to organizational or personal devices.
- **How to request an exception**: Describes whether a user can request an exception to the standard policy, and how that can be done.
- **Audit frequency**: Specifies the frequency of compliance reviews. Higher sensitive labels should involve more frequent and thorough auditing processes.
- **Other metadata**: A data policy requires more metadata, such as policy owner, approver, and effective date.

> **Tip**
>
> When creating your data classification and protection policy, focus on making it a straightforward reference for users. It should be as short and clear as possible. If it's too complex, users won't always take the time to understand it.

One way to automate the implementation of a policy, such as the data classification and protection policy, is with Microsoft Entra terms of use. When a terms of use policy is set up, users are required to acknowledge the policy before they're permitted to visit the Power BI service for the first time. It's also possible to ask them to agree again on a recurring basis, for example every 12 months.

Checklist - When planning the internal policy to govern expectations for usage of sensitivity labels, key decisions and actions include:

- ✓ **Create a data classification and protection policy**: For each sensitivity label in your structure, create a centralized policy document. This document should define what can or can't be done with content that's been assigned each label.
- ✓ **Obtain consensus on the data classification and protection policy**: Ensure that all necessary people in the team you assembled have agreed to the provisions.
- ✓ **Consider how to handle exceptions to the policy**: Highly decentralized organizations should consider whether exceptions might arise. Though it's preferable to have a standardized data classification and protection policy, decide how you'll address exceptions when new requests are made.
- ✓ **Consider where to locate your internal policy**: Give some thought to where the data classification and protection policy should be published. Ensure that all users can easily access it. Plan to include it on the custom help page when you publish the label policy.

User documentation and training

Before rolling out information protection functionality, we recommend that you create and publish guidance documentation for your users. The goal of the documentation is to achieve a seamless user experience. Preparing the guidance for your users will also help you make sure you've considered everything.

You can publish the guidance as part of the sensitivity label's custom help page. A SharePoint page or a wiki page in your centralized portal can work well because it will be easy to maintain. A document uploaded to a shared library or Teams site is also a good approach. The URL for the custom help page is specified in the Microsoft Purview compliance portal when you publish the label policy.

> **Tip**
>
> The **custom help page** is an important resource. Links to it are made available in various applications and services.

The user documentation should include the data classification and protection policy described earlier. That internal policy is targeted at all users. Interested users include content creators and consumers who need to understand the implications for labels that have been assigned by other users.

In addition to the data classification and protection policy, we recommend that you prepare guidance for your content creators and owners about:

- **Viewing labels**: Information about what each label means. Correlate each label with your data classification and protection policy.
- **Assigning labels**: Guidance on how to assign and manage labels. Include information they'll need to know, such as mandatory labels, default labels, and how label inheritance works.
- **Workflow**: Suggestions for how to assign and review labels as part of their normal workflow. Labels can be assigned in Power BI Desktop as soon as development begins, which protects the original Power BI Desktop file during the development process.
- **Situational notifications**: Awareness about system-generated notifications that users might receive. For example, a SharePoint site is assigned to a certain sensitivity label, but an individual file has been assigned to more sensitive (higher) label. The user who assigned the higher label will receive an email notification that the label assigned to the file is incompatible with the site where it's stored.

Include information about who users should contact if they have questions or technical issues. Since information protection is an organization-wide project, support is often provided by IT.

FAQs and examples are especially helpful for user documentation.

> **Tip**
>
> Some regulatory requirements include a specific training component.

Checklist - When preparing user documentation and training, key decisions and actions include:

- ✓ **Identify what information to include**: Determine what information should be included so all relevant audiences understand what's expected of them when it comes to protecting data on behalf of the organization.
- ✓ **Publish the custom help page**: Create and publish a custom help page. Include guidance about labeling in the form of FAQs and examples. Include a link to access the data classification and protection policy.
- ✓ **Publish the data classification and protection policy**: Publish the policy document that defines what exactly can or can't be done with content that's been assigned to each label.

- ✓ **Determine whether specific training is needed**: Create or update your user training to include helpful information, especially if there's a regulatory requirement to do so.

User support

It's important to verify who will be responsible for user support. It's common that sensitivity labels are supported by a centralized IT help desk.

You might need to create guidance for the help desk (sometimes known as a *runbook*). You might also need to conduct knowledge transfer sessions to ensure that the help desk is ready to respond to support requests.

Checklist - When preparing for the user support function, key decisions and actions include:

- ✓ **Identify who will provide user support**: When you're defining roles and responsibilities, make sure to include how users will get help for issues related to information protection.
- ✓ **Ensure the user support team is ready**: Create documentation and conduct knowledge transfer sessions to ensure that the help desk is ready to support information protection. Emphasize complex aspects that might confuse users, such as encryption protection.
- ✓ **Communicate between teams**: Discuss the process and expectations with the support team, as well as your Power BI administrators and Center of Excellence. Make sure that everyone involved is prepared for potential questions from Power BI users.

Implementation summary

After the decisions have been made and prerequisites have been met, it's time to begin implementing information protection according to your gradual rollout plan.

The following checklist includes a summarized list of the end-to-end implementation steps. Many of the steps have other details that were covered in previous sections of this article.

Checklist - When implementing information protection, key decisions and actions include:

- ✓ **Verify current state and goals**: Ensure that you have clarity on the current state of information protection in the organization. All goals and requirements for implementing information protection should be clear and actively used to drive the decision-making process.
- ✓ **Make decisions**: Review and discuss all the decisions that are required. This task should occur prior to setting up anything in production.
- ✓ **Review licensing requirements**: Ensure that you understand the product licensing and user licensing requirements. Procure and assign more licenses, if necessary.
- ✓ **Publish user documentation**: Publish your data classification and protection policy. Create a custom help page that contains the relevant information that users will need.
- ✓ **Prepare the support team**: Conduct knowledge transfer sessions to ensure that the support team is ready to handle questions from users.
- ✓ **Create the sensitivity labels**: Set up each of the sensitivity labels in the Microsoft Purview compliance portal.
- ✓ **Publish a sensitivity label policy**: Create and publish a label policy in the Microsoft Purview compliance portal. Start by testing with a small group of users.
- ✓ **Set the Fabric tenant settings**: In the Fabric admin portal, set the information protection tenant settings.
- ✓ **Perform initial testing**: Perform an initial set of tests to verify everything is working correctly. Use a non-production tenant for initial testing, if available.
- ✓ **Gather user feedback**: Publish the label policy to a small subset of users who are willing to test the functionality. Obtain feedback on the process and user experience.
- ✓ **Continue iterative releases**: Publish the label policy to other groups of users. Onboard more groups of users until the entire organization is included.

> **Tip**
>
> These checklist items are summarized for planning purposes. For more details about these checklist items, see the previous sections of this article.

Ongoing monitoring

After you've completed the implementation, you should direct your attention to monitoring and tuning sensitivity labels.

Power BI administrators and security and compliance administrators will need to collaborate from time to time. For Power BI content, there are two audiences who are concerned with monitoring.

- **Power BI administrators**: An entry in the Power BI activity log is recorded each time a sensitivity label is assigned or changed. The activity log entry records details of the event, including user, date and time, item name, workspace, and capacity. Other activity log events (such as when a report is viewed) will include the sensitivity label ID that's assigned to the item.
- **Security and compliance administrators**: The organization's security and compliance administrators will typically use Microsoft Purview reports, alerts, and audit logs.

Checklist - When monitoring information protection, key decisions and actions include:

- ✓ **Verify roles and responsibilities**: Ensure that you're clear on who is responsible for which actions. Educate and communicate with your Power BI administrators or security administrators, if they'll be directly responsible for some aspects.
- ✓ **Create or validate your process for reviewing activity**: Make sure the security and compliance administrators are clear on the expectations for reviewing the activity explorer regularly.

> **Tip**
>
> For more information about auditing, see **Auditing of information protection and data loss prevention for Power BI**.

Related content

In the next article in this series, learn about data loss prevention for Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Data loss prevention for Power BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article describes the planning activities related to implementing data loss prevention (DLP) in Power BI. It's targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators need to collaborate with information security teams and other relevant teams.
- **Center of Excellence, IT, and BI teams**: Others who are responsible for overseeing Power BI in the organization. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.

> **Important**
>
> Data loss prevention (DLP) is a significant organization-wide undertaking. Its scope and impact are far greater than Power BI alone. This type of initiative requires funding, prioritization, and planning. Expect to involve several cross-functional teams in your planning, usage, and oversight efforts.

We recommend that you follow a gradual, phased approach to rolling out DLP for Power BI. For a description of the types of rollout phases that you should consider, see Information protection for Power BI (Rollout phases).

Purpose of DLP

Data loss prevention (DLP) refers to activities and practices that safeguard the organization's data. The goal for DLP is to reduce the risk of data leakage, which can happen when sensitive data is shared with unauthorized people. Although responsible user behavior is a critical part of safeguarding data, DLP usually refers to policies that are automated.

DLP allows you to:

- Detect and inform administrators when risky, inadvertent, or inappropriate sharing of sensitive data has occurred. Specifically, it allows you to:
  - Improve the overall security setup of your Power BI tenant, with automation and information.
  - Enable analytical use cases that involve sensitive data.
  - Provide auditing information to security administrators.
- Provide users with contextual notifications. Specifically, it allows you to:
  - Help users make the right decisions during their normal workflow.
  - Guide users to follow your data classification and protection policy without negatively affecting their productivity.

DLP services

Broadly, there are two different services that can implement data loss prevention.

- Microsoft Purview DLP policies for Power BI
- Microsoft Defender for Cloud Apps

Microsoft Purview DLP policies for Power BI

A DLP policy for Power BI is set up in the Microsoft Purview compliance portal. It can detect sensitive data in a semantic model that's been published to a Premium workspace in the Power BI service.

> **ⓘ Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

The goal for this type of DLP policy is to bring awareness to users and inform administrators of where sensitive data is stored. The DLP policy can generate user notifications and administrator alerts based on sensitive information types or sensitivity labels. For example, you can determine whether credit card information or personally identifiable information (PII) is stored in a semantic model.

> **Note**
>
> DLP for Power BI is the focus of this article.

Microsoft Defender for Cloud Apps

Microsoft Defender for Cloud Apps is a tool with many capabilities. Some policies that can be set up in Microsoft Defender for Cloud Apps (with integration with Microsoft Entra ID) include DLP. These policies can block, log, or alert when certain user activities happen. For example, when a user attempts to download a report from the Power BI service that's been assigned a *Highly Restricted* sensitivity label, the download action is blocked.

The Defender for Cloud Apps for Power BI article covers using Defender for Cloud Apps for monitoring the Power BI service. The remainder of this article focuses on DLP for Power BI.

> **Important**
>
> A DLP policy for Power BI that's set up in the Microsoft Purview compliance portal can be applied only for content that's stored in a Power BI **Premium** workspace. However, policies that are set up in Defender for Cloud Apps don't have a similar Power BI Premium prerequisite. Be aware that the functionality, purpose, and available actions differ for the two toolsets. To achieve maximum effect, we recommend that you consider using both toolsets.

Prerequisites for DLP for Power BI

By now, you should have completed the organization-level planning steps that are described in the Information protection for Power BI article. Before proceeding, you should have clarity on:

- **Current state**: The current state of DLP in your organization. You should have an understanding to what extent DLP is already in use, and who's responsible for managing it.
- **Goals and requirements**: The strategic goals for implementing DLP in your organization. Understanding the goals and requirements will serve as a guide for your implementation efforts.

Usually, you implement information protection (described in the Information protection for Power BI article) before you implement DLP. However, that isn't a prerequisite for using DLP for Power BI. If sensitivity labels are published, they can be used with DLP for Power BI. You can also use sensitive information types with DLP for Power BI. Both types are described in this article.

Key decisions and actions

The intention of a DLP policy is to set up an automated action, based on rules and conditions, on the content you intend to protect. You'll need to make some decisions about the rules and conditions that will support your goals and requirements.

The advantage of defining separate rules within a single DLP policy is that you can enable customized alerts or user notifications.

There's a hierarchical precedence to the list of DLP policies, as well as to DLP policy rules, to consider. The precedence will affect which policy gets invoked when it's encountered first.

> **⊗ Caution**
>
> This section isn't an exhaustive list of all possible DLP decisions for all possible applications. Ensure that you work with other stakeholders and system administrators to make decisions that work well for all applications and use cases. For example, we recommend that you investigate additional DLP policies for protecting source files and exported files that are stored in OneDrive or SharePoint. This set of articles focuses only on content in the Power BI service.

Type of sensitive data

A DLP policy for Power BI that's set up in the Microsoft Purview compliance portal can be based on either a sensitivity label or a sensitive information type.

> **ⓘ Important**
>
> Although you can assign sensitivity labels to most types of items in Power BI, the DLP policies described in this article are focused specifically on semantic models. The semantic model must be published to a Premium workspace.

Sensitivity label

You can use sensitivity labels to classify content, ranging from less sensitive to more sensitive.

When a DLP policy for Power BI is invoked, a *sensitivity label rule* checks semantic models (that are published to the Power BI service) for the presence of a certain sensitivity label. As described in the Information protection for Power BI article, a label can be assigned either by a user or by an automated process (for example, an inherited label or a default label).

Here are some examples of when you might create a DLP rule based on a sensitivity label.

- **Regulatory compliance**: You have a sensitivity label that's reserved for data that's subject to a particular regulatory requirement. You want to raise an alert for your security administrators when users assign that sensitivity label to a semantic model in the Power BI service.
- **Reminders for content creators about confidential data**: You have a sensitivity label that's used for confidential data. You want to generate a user notification when a user views the semantic model details page within the OneLake catalog in the Fabric portal. For instance, you could remind users about how to appropriately handle confidential data.

Other considerations about user notifications and alerts are described in this next section of this article.

Checklist - When considering the needs for sensitivity label rules, key decisions and actions include:

- ✓ **Verify current state of information protection**: Ensure that sensitivity labels are deployed in the organization, and that they're ready for use by DLP policies.
- ✓ **Compile use cases for DLP based on sensitivity labels**: Determine which sensitivity labels would benefit from having DLP policies in place. Consider your goals, regulations, and internal requirements.
- ✓ **Prioritize the list of use cases for DLP based on sensitivity labels**: Discuss the top priorities with your team. Identify which items to prioritize on your project plan.

> ⓘ Note

DLP policies are typically automated. However, responsible user actions also play a crucial role in protecting data.

For more information, see Information protection for Power BI (Data classification and protection policy). It describes an internal governance policy that provides guidance about what users can and can't do with content that's been assigned to a certain sensitivity label.

Sensitive information types

Not all types of data are the same; certain types of data are inherently more sensitive than others. There are many different sensitive information types (SITs). Depending on your industry and the compliance requirements, only some SITs will be applicable to your organization.

Some common examples of SITs include:

- Passport, social security, and driver's license numbers
- Bank account and routing numbers
- Credit card and debit card numbers
- Tax identification and national ID numbers
- Health ID numbers and medical information
- Physical addresses
- Account keys, passwords, and database connection strings

Tip

If sensitive data doesn't have analytical value, ask yourself whether it should reside in an analytical system. We recommend that you educate your content creators to help them make good decisions about what data to store in Power BI.

SITs are pattern-based classifiers. They'll look for a known pattern in text by using regular expressions.

You'll find many pre-configured SITs in the Microsoft Purview compliance portal. When they meet your requirements, you should use a pre-configured SIT to save time. Consider the credit card number pre-configured SIT: It detects the correct patterns for all major card issuers, ensures the validity of the checksum, and searches for a relevant keyword within proximity to the credit card number.

If the pre-configured SITs don't meet your needs, or you have proprietary data patterns, you can create a custom SIT. For example, you can create a custom SIT to match the pattern of your employee ID number.

Once the SIT is set up, a DLP policy for Power BI is invoked when a semantic model is uploaded or refreshed. At that time, a *sensitive information type rule* will check semantic models (in the Power BI service) for the presence of sensitive information types.

Here are some examples of when you might create a DLP rule based on a sensitive information type.

- **Regulatory compliance**: You have a sensitive information type that's subject to regulatory requirements. You want to generate an alert for your security administrators when that type of data is detected within a semantic model in the Power BI service.
- **Internal requirements**: You have a sensitive information type that needs special handling. To meet internal requirements, you want to generate a user notification when a user views the semantic model settings or the semantic model details page in the OneLake catalog (in the Fabric portal).

Checklist - When considering the needs for sensitive information types, key decisions and actions include:

- ✓ **Compile use cases for DLP based on sensitive information types**: Determine which sensitive information types would benefit from having DLP policies in place. Consider your goals, regulations, and internal requirements.
- ✓ **Test existing sensitive information types**: Work with the information security team to verify that the pre-configured SITs will meet your needs. Use test data to confirm that patterns and keywords are correctly detected.
- ✓ **Create custom sensitive information types**: If applicable, work with your information security team to create SITs so they can be used in DLP for Power BI.
- ✓ **Prioritize the list of use cases**: Discuss the top priorities with your team. Identify which items to prioritize on your project plan.

User notifications

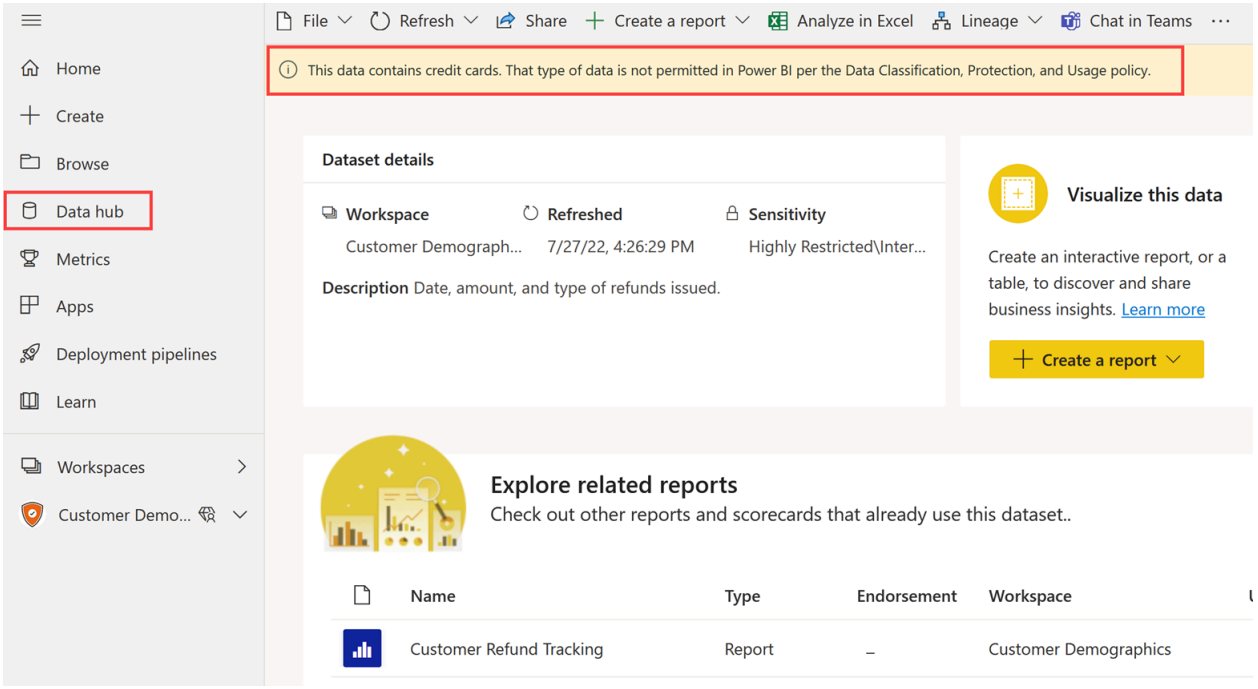
When you've identified use cases for DLP with sensitivity labels and SITs, you should next consider what happens when a DLP rule match occurs. Typically, it involves a user

notification.

User notifications for DLP policies are also known as *policy tips*. They're useful when you want to provide more guidance and awareness to your users during the normal course of their work. It's more likely that users will read and absorb user notifications if they're:

- **Specific**: Correlating the message to the rule makes it far easier to understand.
- **Actionable**: Offering a suggestion for what the user needs to do, or how to find more information.

For DLP in Power BI, user notifications appear in the semantic model settings. They also appear across the top of the semantic model details page in the OneLake catalog, as shown in the following screenshot. In this instance, the notification reads: *This data contains credit cards. That type of data is not permitted in Power BI per the Data Classification, Protection, and Usage policy.*



You can define one or more rules for each DLP policy. Each rule can optionally have a different policy tip that will be displayed to users.

Consider the following example of how you could define a DLP policy for detecting financial data stored within semantic models in the Power BI service. The DLP policy uses SITs and it has two rules.

- **Rule 1**: The first rule detects credit card numbers. The customized policy tip text reads: *This data contains credit card numbers. That type of data is not permitted in Power BI per the Data Classification and Protection policy.*
- **Rule 2**: The second rule detects financial accounts. The customized policy tip text reads: *This data contains sensitive financial information. It requires the use of the*

Highly Restricted label. Please refer to the Data Classification and Protection Policy for requirements when storing financial data.

Rule 1 is more urgent than rule 2. Rule 1 is intended to communicate that there's a problem that requires action. The second rule is more informational. For urgent issues, it's a good idea to set up alerting. Alerting for administrators is described in the next section.

When deciding what notifications users should receive, we recommend that you focus on showing only highly important notifications. If there are too many policy notifications, users could become overwhelmed by them. The result is that some notifications might be overlooked.

Users can take action by reporting an issue when they believe it's a false positive (misidentified). It's also possible to allow the user to override the policy. These capabilities are intended to allow communication between Power BI users and the security administrators who manage DLP for Power BI.

Checklist - When considering DLP user notifications, key decisions and actions include:

- ✓ **Decide when user notifications are needed**: For each DLP rule you intend to create, determine whether a custom user notification is required.
- ✓ **Create customized policy tips**: For each notification, define what message should be shown to users. Plan to correlate the message to the DLP rule so that it's specific and actionable.

Administrator alerting

Alerting is useful for certain DLP rules when you want to track incidents when a policy violation occurred. When you define DLP policy rules, consider whether alerts should be generated.

> **Tip**
>
> Alerts are designed to call the attention of an administrator to certain situations. They're best suited when you intend to actively investigate and resolve important alerts. You can find all DPS rule matches in the activity explorer in the Microsoft Purview compliance portal.

Alerting is useful when you want to:

- Make your security and compliance administrators aware that something occurred via the DLP alert management dashboard. Optionally, you can also send an email to a specific set of users.
- See more details for an event that occurred.
- Assign an event to someone to investigate it.
- Manage the status of an event or add comments to it.
- View other alerts generated for activity by the same user.

Each alert can be defined by a severity level, which can be low, medium, or high. The severity level helps to prioritize the review of open alerts.

Here are two examples of how alerts can be used.

Example 1: You've defined a DLP policy for detecting financial data stored in semantic models in the Power BI service. The DLP policy uses sensitive information types. It has two rules.

- **Rule 1**: This rule detects credit card numbers. Alerting is enabled with a high severity. An email is generated too.
- **Rule 2**: This rule detects financial accounts. Alerting is enabled with a high severity.

Example 2: You've defined a DLP policy that's invoked when the *Highly Restricted\Executive Committee and Board Members* sensitivity label is assigned to a semantic model in the Power BI service. It doesn't generate a user notification. In this situation, you might not want to generate an alert because you only want to log the occurrence. If needed, you can obtain more information from the activity explorer.

When an email alert is required, we recommend that you use a mail-enabled security group. For example, you might use a group named *Security and Privacy Admin Alerting*.

> **Tip**
>
> Keep in mind that DLP rules for Power BI are checked every time that a semantic model is uploaded or refreshed. That means an alert could be generated each time the semantic model is refreshed. Regular or frequent data refreshes could result in an overwhelming number of logged events and alerts.

Checklist - When considering DLP alerting for administrators, key decisions and actions include:

- ✓ **Decide when alerts are required**: For each DLP rule you intend to create, determine which situations warrant using alerts.
- ✓ **Clarify roles and responsibilities**: Determine the expectations and the specific actions that should be taken when an alert is generated.
- ✓ **Determine who will receive alerts**: Decide which security and compliance administrators will handle open alerts. Confirm that permissions and licensing requirements are met for each administrator that will use the Microsoft Purview compliance portal.
- ✓ **Create email groups**: If necessary, create new mail-enabled security groups to handle alerting.

Workspaces in scope

A DLP policy for Power BI that's set up in the Microsoft Purview compliance portal is intended to target semantic models. Specifically, it supports scanning semantic models that have been published to a Premium workspace.

You can set up the DLP policy to scan all Premium workspaces. Optionally, you can choose to include, or exclude, specific workspaces. For example, you might exclude certain development or test workspaces that are considered lower risk (particularly if they don't contain real production data). Alternatively, you might create separate policies for certain development or test workspaces.

> **Tip**
>
> If you decide that only a subset of your Premium workspaces will be included for DLP, consider the level of maintenance. DLP rules are easier to maintain when all Premium workspaces are included. If you decide to include only a subset of Premium workspaces, make sure that you have an auditing process in place so you can quickly identify whether a new workspace is missing from the DLP policy.

For more information about workspace, see the workspace planning articles.

Checklist - When considering which workspaces to include in scope for DLP, key decisions and actions include:

- ✓ **Decide which Premium workspaces should have DLP applied**: Consider whether the DLP policies should affect all Power BI Premium workspaces or only a subset of them.
- ✓ **Create documentation for workspace assignments**: If applicable, document which workspaces are subject to DLP. Include the criteria and reasons why workspaces are included or excluded.
- ✓ **Correlate DLP decisions with your workspace governance**: If applicable, update your workspace governance documentation to include details about how DLP is handled.
- ✓ **Consider other important file locations**: In addition to the Power BI service, determine whether it's necessary to create other DLP policies to protect source files and exported files that are stored in OneDrive or SharePoint.

Licensing requirements

To use DLP, there are several licensing requirements. A Microsoft Purview Information Protection license is required for the administrators who will set up, manage, and oversee DLP. You might already have these licenses because they're included in some license suites, such as Microsoft 365 E5. Alternatively, Microsoft 365 E5 Compliance capabilities can be purchased as a standalone license.

Also, DLP policies for Power BI require Power BI Premium. This licensing requirement can be met with a Premium capacity or a Premium Per User (PPU) license.

> **Tip**
>
> If you need clarifications about licensing requirements, talk to your Microsoft account team. Note that the Microsoft 365 E5 Compliance license includes other DLP capabilities that are out scope for this article.

Checklist - When evaluating DLP licensing requirements, key decisions and actions include:

- ✓ **Review product licensing requirements**: Ensure that you've reviewed all the licensing requirements for DLP.
- ✓ **Review Premium licensing requirements**: Verify that the workspaces you want to configure for DLP are Premium workspaces.
- ✓ **Procure additional licenses**: If applicable, purchase more licenses to unlock the functionality that you intend to use.
- ✓ **Assign licenses**: Assign a license to each of your security and compliance administrators who will need one.

User documentation and training

Before rolling out DLP for Power BI, we recommend that you create and publish user documentation. A SharePoint page or a wiki page in your centralized portal can work well because it will be easy to maintain. A document uploaded to a shared library or Teams site is a good solution, too.

The goal of the documentation is to achieve a seamless user experience. Preparing user documentation will also help you make sure you've considered everything.

Include information about who to contact when users have questions or technical issues. Because information protection is an organization-wide project, support is often provided by IT.

FAQs and examples are especially helpful for user documentation.

> **Tip**
>
> For more information, see **Information protection for Power BI (Data classification and protection policy)**. It describes suggestions for creating a data classification and protection policy so that users understand what they can and can't do with sensitivity labels.

Checklist - When preparing user documentation and training, key decisions and actions include:

- ✓ **Update documentation for content creators and consumers**: Update your FAQs and examples to include relevant guidance about DLP policies.

- ✓ **Publish how to get help**: Ensure that your users know how to get help when they're experiencing something unexpected or that they don't understand.
- ✓ **Determine whether specific training is needed**: Create or update your user training to include helpful information, especially if there's a regulatory requirement to do so.

User support

It's important to verify who will be responsible for user support. It's common that DLP is supported by a centralized IT help desk.

You might need to create guidance for the help desk (sometimes known as a *runbook*). You might also need to conduct knowledge transfer sessions to ensure that the help desk is ready to respond to support requests.

Checklist - When preparing for the user support function, key decisions and actions include:

- ✓ **Identify who will provide user support**: When you're defining roles and responsibilities, make sure to account for how users will get help with issues related to DLP.
- ✓ **Ensure the user support team is ready**: Create documentation and conduct knowledge transfer sessions to ensure that the help desk is ready to support DLP.
- ✓ **Communicate between teams**: Discuss user notifications and the process to resolve DLP alerts with the support team, as well as your Power BI administrators and Center of Excellence. Make sure that everyone involved is prepared for potential questions from Power BI users.

Implementation and testing summary

After the decisions have been made and prerequisites have been met, it's time to begin implementing and testing DLP for Power BI.

DLP policies for Power BI are set up in the Microsoft Purview compliance portal (formerly known as the Microsoft 365 compliance center) in the Microsoft 365 admin center.

> **Tip**
>
> The process to set up DLP for Power BI in the Microsoft Purview compliance portal involves just one step, instead of two, to set up the policy. This process is different from when you set up information protection in the Microsoft Purview compliance portal (described in the **Information protection for Power BI** article). In that case, there were two separate steps to set up the label and publish a label policy. In this case for DLP, there's just one step in the implementation process.

The following checklist includes a summarized list of the end-to-end implementation steps. Many of the steps have other details that were covered in previous sections of this article.

Checklist - When implementing DLP for Power BI, key decisions and actions include:

- ✓ **Verify current state and goals**: Ensure that you have clarity on the current state of DLP for use with Power BI. All goals and requirements for implementing DLP should be clear and actively used to drive the decision-making process.
- ✓ **Make decisions**: Review and discuss all the decisions that are required. This task should occur prior to setting up anything in production.
- ✓ **Review licensing requirements**: Ensure that you understand the product licensing and user licensing requirements. If necessary, procure and assign more licenses.
- ✓ **Publish user documentation**: Publish information that users will need to answer questions and clarify expectations. Provide guidance, communications, and training to your users so they're prepared.
- ✓ **Create DLP policies**: In the Microsoft Purview compliance portal, create and set up each DLP policy. Refer to all the decisions previously made for setting up the DLP rules.
- ✓ **Perform initial testing**: Perform an initial set of tests to verify everything is set up correctly. Use test mode with some sample data to determine whether everything behaves as you'd expect, while minimizing the impact on users. Use a small subset of Premium workspaces initially. Consider using a non-production tenant when you have access to one.
- ✓ **Gather user feedback**: Obtain feedback on the process and user experience. Identify areas of confusion, or unexpected results with sensitive information types, and other technical issues.

- ✓ **Continue iterative releases**: Gradually add more Premium workspaces to the DLP policy until they're all included.
- ✓ **Monitor, tune, and adjust**: Invest resources to review policy match alerts and audit logs on a frequent basis. Investigate false positives and adjust policies when necessary.

> **Tip**
>
> These checklist items are summarized for planning purposes. For more details about these checklist items, see the previous sections of this article.

For other steps to take beyond the initial rollout, see Defender for Cloud Apps with Power BI.

Ongoing monitoring

After you've completed the implementation, you should direct your attention to monitoring, enforcing, and adjusting DLP policies based on their use.

Power BI administrators and security and compliance administrators will need to collaborate from time to time. For Power BI content, there are two audiences for monitoring.

- **Power BI administrators**: An entry in the Power BI activity log is recorded each time there's a DLP rule match. The Power BI activity log entry records details of the DLP event, including user, date and time, item name, workspace, and capacity. It also includes information about the policy, such as the policy name, rule name, severity, and the matched condition.
- **Security and compliance administrators**: The organization's security and compliance administrators will typically use Microsoft Purview reports, alerts, and audit logs.

> **⚠ Warning**
>
> Monitoring for DLP for Power BI policies doesn't occur in real time because it takes time for DLP logs and alerts to be generated. If your goal is real-time enforcement, see Defender for Cloud Apps for Power BI (Real-time policies).

Checklist - When monitoring DLP for Power BI, key decisions and actions include:

- ✓ **Verify roles and responsibilities**: Ensure that you're clear on who is responsible for which actions. Educate and communicate with your Power BI administrators or security administrators, if they'll be directly responsible for some aspects of DLP monitoring.
- ✓ **Create or validate your process for reviewing activity**: Make sure that the security and compliance administrators are clear on the expectations for regularly reviewing the activity explorer.
- ✓ **Create or validate your process for resolving alerts**: Ensure that your security and compliance administrators have a process in place to investigate and resolve DLP alerts when a policy match occurs.

> **Tip**
>
> For more information about auditing, see **Auditing of information protection and data loss prevention for Power BI**.

Related content

In the next article in this series, learn about using Defender for Cloud Apps with Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Defender for Cloud Apps for Power BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article describes the planning activities related to implementing Defender for Cloud Apps as it relates to monitoring Power BI. It's targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators need to collaborate with information security and other relevant teams.
- **Center of Excellence, IT, and BI teams**: Others who are responsible for overseeing Power BI in the organization. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.

> **Important**
>
> Monitoring and data loss prevention (DLP) is a significant organization-wide undertaking. Its scope and impact are far greater than Power BI alone. These types of initiative require funding, prioritization, and planning. Expect to involve several cross-functional teams in planning, usage, and oversight efforts.

We recommend that you follow a gradual, phased approach to rolling out Defender for Cloud Apps for monitoring Power BI. For a description of the types of rollout phases that you should consider, see Information protection for Power BI (Rollout phases).

Purpose of monitoring

Microsoft Defender for Cloud Apps (formerly known as Microsoft Cloud App Security) is a Cloud Access Security Broker (CASB) that supports various deployment modes. It has a broad set of capabilities that extend well beyond the scope of this article. Some capabilities are real-time while others aren't real-time.

Here are some examples of real-time monitoring you can implement.

- **Block downloads from the Power BI service**: You can create a *session policy* to block certain types of user activities. For example, when a user tries to download a report from the Power BI service that's been assigned a *Highly Restricted* sensitivity label, the download action can be blocked in real-time.
- **Block access to the Power BI service by an unmanaged device**: You can create an *access policy* to prevent users from accessing certain applications unless they're using a managed device. For example, when a user tries to access the Power BI service from their personal mobile phone that action can be blocked.

Here are some examples of other capabilities that aren't real time.

- **Detect and alert certain activities in the Power BI service**: You can create an *activity policy* to generate an alert when certain types of activities occur. For example, when an *administrative activity* occurs in the Power BI service (indicating that a tenant setting was changed), you can receive an email alert.
- **Monitor advanced security activities**: You can view and monitor sign-ins and security activities, anomalies, and violations. Alerts can be raised for situations such as suspicious activity, unexpected locations, or a new location.
- **Monitor user activities**: You can view and monitor user activities. For example, a Power BI administrator could be assigned permission to view the Power BI activity log, in addition to user sign-in frequency within Defender for Cloud Apps.
- **Detect and alert unusual behavior in the Power BI service**: There are built-in policies for anomaly detection. For example, when a user downloads or exports content from the Power BI service significantly more often than normal patterns, you can receive an email alert.
- **Find unsanctioned applications**: You can find unsanctioned applications in use within the organization. For example, you might become concerned about users sharing files (like Power BI Desktop files or Excel files) on a third-party file sharing system. You can block use of an unsanctioned application, and then contact users to educate them on appropriate ways to share and collaborate with others.

> **Tip**
>
> The portal in Defender for Cloud Apps is a convenient place to view activities and alerts without creating a script to extract and download the data. This advantage includes viewing data from the Power BI activity log.

Power BI is one of many applications and services that can be integrated with Defender for Cloud Apps. If you're already using Defender for Cloud Apps for other purposes, it

can be used to monitor Power BI too.

Policies created in Defender for Cloud Apps are a form of DLP. The Data loss prevention for Power BI article covers DLP policies for Power BI that are set up in the Microsoft Purview compliance portal. We recommend that you use DLP policies for Power BI with the capabilities described in this article. Although there's some overlap conceptually, the capabilities are different.

> ⊗ **Caution**
>
> This article focuses on capabilities in **Microsoft Defender for Cloud Apps** that can be used to monitor and protect Power BI content. There are many other capabilities in Defender for Cloud Apps that aren't covered in this article. Be sure to work with other stakeholders and system administrators to make decisions that work well for all applications and use cases.

Prerequisites for Defender for Cloud Apps for Power BI

By now, you should have completed the organization-level planning steps that were described in the Data loss prevention for Power BI article. Before proceeding, you should have clarity on:

- **Current state**: The current state of DLP in your organization. You should have an understanding to what extent DLP is already in use, and who's responsible for managing it.
- **Goals and requirements**: The strategic goals for implementing DLP in your organization. Understanding the goals and requirements will serve as a guide for your implementation efforts.

Usually, information protection is already implemented before DLP is implemented. If sensitivity labels are published (described in the Information protection for Power BI article), they can be used in certain policies within Defender for Cloud Apps.

You might have already implemented DLP for Power BI (described in the Data loss prevention for Power BI article). Those DLP capabilities are different to the capabilities that are managed in the Microsoft Purview compliance portal. All DLP capabilities described in this article are managed in the Defender for Cloud Apps portal.

Key decisions and actions

You'll need to make some key decisions before you're ready to set up policies in Defender for Cloud Apps.

The decisions related to Defender for Cloud Apps policies should directly support the goals and requirements for protecting the data that you've previously identified.

Policy type and activities

You'll need to consider which user activities you're interested in monitoring, blocking, or controlling. The policy type in Defender for Cloud Apps influences:

- What you're able to accomplish.
- Which activities can be included in the configuration.
- Whether the controls will occur in real time or not.

Real-time policies

Access policies and session policies created in Defender for Cloud Apps allow you to monitor, block, or control user sessions in real time.

Access policies and session policies allow you to:

- **Programmatically respond in real time**: Detect, inform, and block risky, inadvertent, or inappropriate sharing of sensitive data. These actions allow you to:
  - Improve the overall security setup of your Power BI tenant, with automation and information.
  - Enable analytical use cases that involve sensitive data in a way that can be audited.
- **Provide users with contextual notifications**: This capability allows you to:
  - Help users make the right decisions during their normal workflow.
  - Guide users to follow your data classification and protection policy without affecting their productivity.

To provide real-time controls, access policies and session policies work with Microsoft Entra ID, relying on the reverse proxy capabilities of Conditional Access App Control. Instead of user requests and responses going through the app (the Power BI service in this case), they go through a reverse proxy (Defender for Cloud Apps).

Redirection doesn't affect the user experience. However, the URL for the Power BI service will change to *https://app.powerbi.com.mcas.ms* once you've set up Microsoft Entra ID for conditional access app control with Power BI. Also, users will receive a notification when they sign in to the Power BI service that announces that the app is monitored by Defender for Cloud Apps.

> **Important**
>
> Access policies and session policies operate in real time. Other policy types in Defender for Cloud Apps involve a short delay in alerting. Most other types of DLP and auditing also experience latency, including DLP for Power BI and the Power BI activity log.

Access policies

An access policy created in Defender for Cloud Apps controls whether a user is allowed to sign in to a cloud application like the Power BI service. Organizations that are in highly regulated industries will be concerned with access policies.

Here are some examples of how you might use access policies to block access to the Power BI service.

- **Unexpected user**: You can block access for a user who isn't a member of a specific security group. For example, this policy could be helpful when you have an important internal process that tracks approved Power BI users via a specific group.
- **Non-managed device**: You can block access for a personal device that isn't managed by the organization.
- **Updates needed**: You can block access for a user who's using an outdated browser or operating system.
- **Location**: You can block access for a location where you don't have offices or users, or from an unknown IP address.

> **Tip**
>
> If you have external users that access your Power BI tenant or employees who travel frequently, that can affect how you define your access control policies. These types of policies are usually managed by IT.

Session policies

A session policy is useful when you don't want to allow or block access completely (which can be done with an access policy as previously described). Specifically, it allows access for the user while monitoring or limiting what actively occurs during their session.

Here are some examples of ways that you can use session policies to monitor, block, or control user sessions in the Power BI service.

- **Block downloads**: Block downloads and exports when a specific sensitivity label, like *Highly Restricted*, is assigned to the item in the Power BI service.
- **Monitor sign-ins**: Monitor when a user, who meets certain conditions, signs in. For example, the user could be a member of a specific security group or they're using a personal device that isn't managed by the organization.

> **Tip**
>
> Creating a session policy (for example, to prevent downloads) for content that's assigned to a particular sensitivity label, like *Highly Restricted*, is one of the most effective use cases for real-time session controls with Power BI.

It's also possible to control file uploads with session policies. However, typically you want to encourage self-service BI users to upload content to the Power BI service (instead of sharing Power BI Desktop files). Therefore, think carefully about blocking file uploads.

Checklist - When planning your real-time policies in Defender for Cloud Apps, key decisions and actions include:

- ✓ **Identify use cases to block access**: Compile a list of scenarios for when blocking access to the Power BI service is appropriate.
- ✓ **Identify use cases to monitor sign-ins**: Compile a list of scenarios for when monitoring sign-ins to the Power BI service is appropriate.
- ✓ **Identify use cases to block downloads**: Determine when downloads from the Power BI service should be blocked. Determine which sensitivity labels should be included.

Activity policies

Activity policies in Defender for Cloud Apps don't operate in real time.

You can set up an activity policy to check events recorded in the Power BI activity log. The policy can act on a single activity, or it can act on repeated activities by a single user (when a specific activity occurs more than a set number of times within a set number of minutes).

You can use activity policies to monitor activity in the Power BI service in different ways. Here are some examples of what you can achieve.

- **Unauthorized or unexpected user views privileged content**: A user who isn't a member a specific security group (or an external user) has viewed a highly privileged report that's provided to the board of directors.
- **Unauthorized or unexpected user updates tenant settings**: A user who isn't a member a specific security group, like the *Power BI Administrators* group, has updated the tenant settings in the Power BI service. You can also choose to be notified anytime a tenant setting is updated.
- **Large number of deletes**: A user has deleted more than 20 workspaces or reports in a time period that's less than 10 minutes.
- **Large number of downloads**: A user has downloaded more than 30 reports in a time period that's less than five minutes.

The types of activity policy alerts described in this section are commonly handled by Power BI administrators as part of their oversight of Power BI. When setting up alerts within Defender for Cloud Apps, we recommend that you focus on situations that represent significant risk to the organization. That's because each alert will need to be reviewed and closed by an administrator.

> ⚠ **Warning**
>
> Because Power BI activity log events aren't available in real-time, they can't be used for real-time monitoring or blocking. You can, however, use operations from the activity log in activity policies. Be sure to work with your information security team to verify what's technically feasible before you get too far into the planning process.

Checklist - When planning your activity policies, key decisions and actions include:

- ✓ **Identify use cases for activity monitoring**: Compile a list of specific activities from the Power BI activity log that represent significant risk to the organization. Determine whether the risk relates to a single activity or repeated activities.
- ✓ **Coordinate effort with Power BI administrators**: Discuss the Power BI activities that will be monitored in Defender for Cloud Apps. Ensure that there's not a duplication of effort between different administrators.

Users impacted

One of the compelling reasons to integrate Power BI with Defender for Cloud Apps is to benefit from real-time controls when users interact with the Power BI service. This type of integration requires conditional access app control in Microsoft Entra ID.

Before setting up conditional access app control in Microsoft Entra ID, you'll need to consider which users will be included. Usually, all users are included. However, there might be reasons to exclude specific users.

> **Tip**
>
> When setting up the conditional access policy, it's likely that your Microsoft Entra administrator will exclude specific administrator accounts. That approach will prevent locking out administrators. We recommend that the excluded accounts are Microsoft Entra administrators rather than standard Power BI users.

Certain types of policies in Defender for Cloud Apps can apply to certain users and groups. Most often, these types of policies are applicable to all users. However, it's possible that you'll encounter a situation when you'll need to purposefully exclude certain users.

Checklist - When considering which users are affected, key decisions and actions include:

- ✓ **Consider which users are included**: Confirm whether all users will be included in your Microsoft Entra Conditional Access app control policy.
- ✓ **Identify which administrator accounts should be excluded**: Determine which specific administrator accounts should be purposefully excluded from the Microsoft Entra Conditional Access app control policy.
- ✓ **Determine whether certain Defender policies apply to subsets of users**: For valid use cases, consider whether they should be applicable to all or some users (when possible).

User messaging

Having identified use cases, you'll need to consider what should happen when there's user activity that matches the policy.

When an activity is blocked in real time, it's important to provide the user with a customized message. The message is useful when you want to provide more guidance and awareness to your users during their normal workflow. It's more likely that users will read and absorb user notifications when they're:

- **Specific**: Correlating the message to the policy makes it simple to understand.
- **Actionable**: Offering a suggestion for what they need to do, or how to find more information.

Some types of policies in Defender for Cloud Apps can have a customized message. Here are two examples of user notifications.

Example 1: You can define a real-time session control policy that prevents all exports and downloads when the sensitivity label for the Power BI item (like a report or semantic model) is set to *Highly Restricted*. The customized block message in Defender for Cloud Apps reads: *Files with a Highly Restricted label are not permitted to be downloaded from the Power BI service. Please view the content online in the Power BI service. Contact the Power BI support team with any questions.*

Example 2: You can define a real-time access policy that prevents a user from signing in to the Power BI service when they're not using a machine managed by the organization. The customized block message in Defender for Cloud Apps reads: *The Power BI service might not be accessed on a personal device. Please use the device provided by the organization. Contact the Power BI support team with any questions.*

Checklist - When considering user messages in Defender for Cloud Apps, key decisions and actions include:

- ✓ **Decide when a customized block message is needed**: For each policy you intend to create, determine whether a customized block message will be required.
- ✓ **Create customized block messages**: For each policy, define what message should be displayed to users. Plan to relate each message to the policy so that it's specific and actionable.

Administrator alerting

Alerting is useful when you want to make your security and compliance administrators aware that a policy violation has occurred. When you define policies in Defender for

Cloud Apps, consider whether alerts should be generated. For more information, see alert types in Defender for Cloud Apps.

Optionally, you can set up an alert to send an email to multiple administrators. When an email alert is required, we recommend that you use a mail-enabled security group. For example, you might use a group named *Security and Compliance Admin Alerting*.

For high priority situations, it's possible to send alerts by text message. It's also possible to create custom alert automation and workflows by integrating with Power Automate.

You can set up each alert with a low, medium, or high severity. The severity level is helpful when prioritizing the review of open alerts. An administrator will need to review and action each alert. An alert can be closed as true positive, false positive, or benign.

Here are two examples of administrator alerts.

Example 1: You can define a real-time session control policy that prevents all exports and downloads when the sensitivity label for the Power BI item (like a report or semantic model) is set to *Highly Restricted*. It has a helpful customized block message for the user. However, in this situation there isn't a need to generate an alert.

Example 2: You can define an activity policy that tracks whether an external user has viewed a highly privileged report that's provided to the board of directors. A high severity alert can be set up to ensure that the activity is promptly investigated.

> **Tip**
>
> Example 2 highlights the differences between information protection and security. Its activity policy can help identify scenarios where self-service BI users have permission to manage security for content. Yet these users can take actions that are discouraged by the organizational policy. We recommend that you set up these types of policies only in specific circumstances when the information is especially sensitive.

Checklist - When considering alerting for administrators in Defender for Cloud Apps, key decisions and actions include:

- ✓ **Decide when alerts are required**: For each policy you intend to create, decide which situations warrant using alerts.

- ✓ **Clarify roles and responsibilities**: Determine expectations and the action that should be taken when an alert is generated.
- ✓ **Determine who will receive alerts**: Decide which security and compliance administrators will review and action open alerts. Confirm permissions and licensing requirements are met for each administrator who will use Defender for Cloud Apps.
- ✓ **Create a new group**: When necessary, create a new mail-enabled security group to use for email notifications.

Policy naming convention

Before you create policies in Defender for Cloud Apps, it's a good idea to first create a naming convention. A naming convention is helpful when there are many types of policies for many types of applications. It's also useful when Power BI administrators become involved in monitoring.

> **Tip**
>
> Consider granting Defender for Cloud Apps access to your Power BI administrators. Use the **admin role**, which allows viewing the activity log, sign-in events, and events related to the Power BI service.

Consider a naming convention template that includes component placeholders:
<Application> - <Description> - <Action> - <Type of Policy>

Here are some naming convention examples.

Expand table

Type of policy	Real-time	Policy name
Session policy	Yes	Power BI - Highly restricted label - Block downloads - RT
Access policy	Yes	All - Unmanaged device - Block access - RT
Activity policy	No	Power BI - Administrative activity
Activity policy	No	Power BI - External user views executive report

The components of the naming convention include:

- **Application**: The application name. The *Power BI* prefix helps to group all the Power BI-specific policies together when sorted. However, some policies will apply to all cloud apps rather than just the Power BI service.

- **Description**: The description portion of the name will vary the most. It might include sensitivity labels affected or the type of activity being tracked.
- **Action**: (Optional) In the examples, one session policy has an action of *Block downloads*. Usually, an action is only necessary when it's a real-time policy.
- **Type of policy**: (Optional) In the example, the *RT* suffix indicates that it's a real-time policy. Designating whether it's real-time or not helps to manage expectations.

There are other attributes that don't need to be included in the policy name. These attributes include the severity level (low, medium, or high), and the category (such as threat detection or DLP). Both attributes can be filtered on the alerts page.

> **Tip**
>
> You can rename a policy in Defender for Cloud Apps. However, it's not possible to rename the built-in anomaly detection policies. For example, the *Suspicious Power BI report sharing* is a built-in policy that can't be renamed.

Checklist - When considering the policy naming convention, key decisions and actions include:

- ✓ **Choose a naming convention**: Use your first policies to establish a consistent naming convention that's straightfoward to interpret. Focus on using a consistent prefix and suffix.
- ✓ **Document the naming convention**: Provide reference documentation about the policy naming convention. Make sure your system administrators are aware of the naming convention.
- ✓ **Update existing policies**: Update any existing Defender policies to comply with the new naming convention.

Licensing requirements

Specific licenses must be in place to monitor a Power BI tenant. Administrators must have one of the following licenses.

- **Microsoft Defender for Cloud Apps**: Provides Defender for Cloud Apps capabilities for all supported applications (including the Power BI service).

- **Office 365 Cloud App Security**: Provides Defender for Cloud Apps capabilities for Office 365 apps that are part of the Office 365 E5 suite (including the Power BI service).

Also, if users need to use real-time access policies or session policies in Defender for Cloud Apps, they will need an Microsoft Entra ID P1 license.

> **Tip**
>
> If you need clarifications about licensing requirements, talk to your Microsoft account team.

Checklist - When evaluating licensing requirements, key decisions and actions include:

- ✓ **Review product licensing requirements**: Ensure that you've reviewed all the licensing requirements for working with Defender for Cloud Apps.
- ✓ **Procure additional licenses**: If applicable, purchase more licenses to unlock the functionality that you intend to use.
- ✓ **Assign licenses**: Assign a license to each of your security and compliance administrators who will use Defender for Cloud Apps.

User documentation and training

Before rolling out Defender for Cloud Apps, we recommend that you create and publish user documentation. A SharePoint page or a wiki page in your centralized portal can work well because it will be easy to maintain. A document uploaded to a shared library or Teams site is a good solution, too.

The goal of the documentation is to achieve a seamless user experience. Preparing user documentation will also help you make sure you've considered everything.

Include information about who to contact when users have questions or technical issues.

FAQs and examples are especially helpful for user documentation.

Checklist - When preparing user documentation and training, key decisions and actions include:

- ✓ **Update documentation for content creators and consumers**: Update your FAQs and examples to include relevant information about policies that users might encounter.
- ✓ **Publish how to get help**: Ensure that your users know how to get help when they're experiencing something unexpected or that they don't understand.
- ✓ **Determine whether specific training is needed**: Create or update your user training to include helpful information, especially if there's a regulatory requirement to do so.

User support

It's important to verify who will be responsible for user support. It's common that using Defender for Cloud Apps to monitor Power BI is done by a centralized IT help desk.

You might need to create documentation for the help desk and conduct some knowledge transfer sessions to ensure the help desk is ready to respond to support requests.

Checklist - When preparing for the user support function, key decisions and actions include:

- ✓ **Identify who will provide user support**: When you're defining roles and responsibilities, make sure to account for how users will get help with issues that they might encounter.
- ✓ **Ensure the user support team is ready**: Create documentation and conduct knowledge transfer sessions to ensure that the help desk is ready to support these processes.
- ✓ **Communicate between teams**: Discuss messages users might see and the process to resolve open alerts with your Power BI administrators and Center of Excellence. Make sure that everyone involved is prepared for potential questions from Power BI users.

Implementation summary

After the decisions have been made, and a rollout plan has been prepared, it's time to start the implementation.

If you intend to use real-time policies (session policies or access policies), your first task is to set up Microsoft Entra conditional access app control. You'll need to set up the Power BI service as a catalog app that will be controlled by Defender for Cloud Apps.

When Microsoft Entra Conditional Access app control is set up and tested, you can then create policies in Defender for Cloud Apps.

> **ⓘ Important**
>
> We recommend that you introduce this functionality to a small number of test users first. There's also a monitor-only mode that you might find helpful to introduce this functionality in an orderly way.

The following checklist includes a summarized list of the end-to-end implementation steps. Many of the steps have other details that were covered in previous sections of this article.

Checklist - When implementing Defender for Cloud Apps with Power BI, key decisions and actions include:

- ✓ **Verify current state and goals**: Ensure that you have clarity on the current state of DLP for use with Power BI. All goals and requirements for implementing DLP should be clear and actively used to drive the decision-making process.
- ✓ **Carry out the decision-making process**: Review and discuss all the decisions that are required. This task should occur prior to setting up anything in production.
- ✓ **Review licensing requirements**: Ensure that you understand the product licensing and user licensing requirements. If necessary, procure and assign more licenses.
- ✓ **Publish user documentation**: Publish information that users will need to answer questions and clarify expectations. Provide guidance, communications, and training to your users so they're prepared.
- ✓ **Create a Microsoft Entra Conditional Access policy**: Create a conditional access policy in Microsoft Entra ID to enable real-time controls for monitoring the Power BI service. At first, enable the Microsoft Entra Conditional Access policy for a few test users.

- ✓ **Set Power BI as a connected app in Defender for Cloud Apps**: Add or verify that Power BI appears as a connected app in Defender for Cloud Apps for conditional access app control.
- ✓ **Perform initial testing**: Sign in to the Power BI service as one of the test users. Verify that access works. Also verify that the message displayed informs you that the Power BI service is monitored by Defender for Cloud Apps.
- ✓ **Create and test a real-time policy**: Using the use cases already compiled, create an access policy or a session policy in Defender for Cloud Apps.
- ✓ **Perform initial testing**: As a test user, perform an action that will trigger the real-time policy. Verify the action is blocked (if appropriate) and that the expected alert messages are displayed.
- ✓ **Gather user feedback**: Obtain feedback on the process and user experience. Identify areas of confusion, unexpected results with sensitive information types, and other technical issues.
- ✓ **Continue iterative releases**: Gradually add more policies in Defender for Cloud Apps until all use cases are addressed.
- ✓ **Review the built-in policies**: Locate the built-in anomaly detection policies in Defender for Cloud Apps (that have Power BI in their name). Update the alert settings for the built-in policies, when necessary.
- ✓ **Proceed with a broader rollout**: Continue to work through your iterative rollout plan. Update the Microsoft Entra Conditional Access policy to apply to a broader set of users, as appropriate. Update individual policies in Defender for Cloud Apps to apply to a broader set of users, as appropriate.
- ✓ **Monitor, tune, and adjust**: Invest resources to review policy match alerts and audit logs on a frequent basis. Investigate any false positives and adjust policies when necessary.

> **Tip**
>
> These checklist items are summarized for planning purposes. For more details about these checklist items, see the previous sections of this article.

For more specific information about deploying Power BI as a catalog application in Defender for Cloud Apps, see the steps to deploy catalog apps.

Ongoing monitoring

After you've completed the implementation, you should direct your attention to monitoring, enforcing, and adjusting Defender for Cloud Apps policies based on their usage.

Power BI administrators and security and compliance administrators will need to collaborate from time to time. For Power BI content, there are two audiences for monitoring.

- **Power BI administrators**: In addition to alerts generated by Defender for Cloud Apps, activities from the Power BI activity log are also displayed in the Defender for Cloud Apps portal.
- **Security and compliance administrators**: The organization's security and compliance administrators will typically use Defender for Cloud Apps alerts.

It's possible to provide your Power BI administrators with a limited view in Defender for Cloud Apps. It uses a *scoped role* to view the activity log, sign-in events, and events related to the Power BI service. This capability is a convenience for Power BI administrators.

Checklist - When monitoring Defender for Cloud Apps, key decisions and actions include:

- ✓ **Verify roles and responsibilities**: Ensure that you're clear on who is responsible for which actions. Educate and communicate with your Power BI administrators if they'll be responsible for any aspect of monitoring.
- ✓ **Manage access for Power BI administrators**: Add your Power BI administrators to the scoped admin role in Defender for Cloud Apps. Communicate with them so they're aware of what they can do with this extra information.
- ✓ **Create or validate your process for reviewing activity**: Make sure your security and compliance administrators are clear on the expectations for regularly reviewing the activity explorer.
- ✓ **Create or validate your process for resolving alerts**: Ensure that your security and compliance administrators have a process in place to investigate and resolve open alerts.

Related content

In the next article in this series, learn about auditing for information protection and data loss prevention for Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Auditing of information protection and data loss prevention for Power BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article describes the type of auditing you can do after implementing information protection and data loss prevention (DLP). It's targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators need to collaborate with information security and other relevant teams.
- **Center of Excellence, IT, and BI teams**: Others who are responsible for overseeing Power BI in the organization. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.

It's important to understand how information protection and data loss prevention is used in your organization. You can achieve that by performing auditing, which can:

- Track usage patterns, activities, and adoption
- Support governance and security requirements
- Find non-compliance issues with specific requirements
- Document the current setup
- Identify user education and training opportunities

Checklist - When considering auditing for information protection and DLP, key decisions and actions include:

- ✓ **Decide what's most important to audit**: Consider what's most important from an auditing perspective. Prioritize areas of risk, major inefficiencies, or non-compliance

with regulatory requirements. When a situation arises that could be improved, educate users on appropriate ways to do things.

- ✓ **Implement relevant auditing processes**: Put processes in place to extract, integrate, model, and create reports so that auditing can be done.
- ✓ **Take appropriate action**: Using the information obtained from the auditing processes, make sure that someone has the authority and time to take appropriate action. Depending on the situation, it could involve adjusting which sensitivity labels are assigned to content. Other situations might involve user education or training.

The remainder of this article describes useful auditing processes and suggestions.

Power BI activity log

To help with information protection, you can use the Power BI activity log to track activities related to sensitivity labels.

When you've implemented DLP for Power BI, the activity log tracks when there's a DLP rule match.

- **What to look for**: You can determine when specific activities occur, such as:
  - Sensitivity labels were applied, changed, deleted, and by which users
  - Whether labels were applied manually
  - Whether labels were applied automatically (for example, by inheritance or a deployment pipeline)
  - Whether a changed label was upgraded (to a more sensitive label) or downgraded (to a less sensitive label)
  - How frequently DLP events are triggered, where, and by which users
- **Actions to take**: Ensure that data from the activity log data is extracted regularly by an administrator who has permission to extract tenant-level metadata. Determine how to classify activities to support your auditing needs. Some activities might justify review by an administrator or content owner (for example, when a label is deleted). Other activities might justify being included in regular audit reviews (for example, when labels are downgraded, or when DLP rule matches occur).
- **Where to find this data**: Power BI administrators can use the Power BI activity log to view activities related to Power BI content. Alternatively, in Defender for Cloud Apps, you can grant your Power BI administrators a limited view so they can see activity log events, sign-in events, and other events related to the Power BI service.

Power BI protection metrics

The data protection metrics report is a dedicated report in the Power BI admin portal. It summarizes how sensitivity labels are assigned to content in your Power BI tenant.

- **What to look for**: You can gain a quick sense for how frequently sensitivity labels are applied to each type of item (for example, semantic model or report) in the Power BI service.
- **Actions to take**: Review this report to become familiar with how much content doesn't have a label applied.
- **Where to find this data**: Power BI administrators can find the data protection metrics report in the Power BI admin portal.

> **Tip**
>
> The data protection metrics report is a summary report. You can also use the scanner APIs, which are described in the next section, to perform deeper analysis.

Power BI scanner APIs

The Power BI scanner APIs allow you to scan the metadata in your Power BI tenant. The metadata of Power BI items, like semantic models and reports, can help you to monitor and review self-service user activity.

For example, you might discover that content in a financial workspace has been assigned to three different sensitivity labels. If any of these labels aren't appropriate for financial data, you can apply more suitable labels.

- **What to look for**: You can create an inventory of Power BI items in your tenant, including the sensitivity label of each item.
- **Actions to take**: Create a process to scan your tenant on a weekly or monthly basis. Use the metadata retrieved by the scanner APIs to understand how Power BI content has been labeled. Investigate further if you find that some labels don't meet expectations for the workspace. Correlate metadata from the scanner APIs with events from the Power BI activity log to determine when a sensitivity label was applied, changed, deleted, and by which user.
- **Where to find this data**: Power BI administrators can use the Power BI scanner APIs to retrieve a snapshot of the sensitivity labels applied to all Power BI content. If you prefer to build your own inventory reports, you can use the APIs directly by writing scripts. Alternatively, you can use the APIs indirectly by registering Power BI in the Microsoft Purview Data Map (which uses the Power BI scanner APIs to scan the Power BI tenant).

Microsoft Purview activity explorer

Activity explorer in the Microsoft Purview compliance portal aggregates useful auditing data. This data can help you to understand the activities across applications and services.

> **Tip**
>
> Activity explorer exposes only certain types of Power BI events. Plan to use both the Power BI activity log and the activity explorer to see view events.

- **What to look for**: You can use activity explorer to view sensitivity label activity from various applications, including Teams, SharePoint Online, OneDrive, Exchange Online, and Power BI. It's also possible to see when a file was read, where, and by which user. Certain types of DLP policy events are also shown in the activity explorer. When a justification is provided to explain a change of sensitivity label, you can view the reason in activity explorer.
- **Actions to take**: Regularly review activity explorer events to identify whether there are areas of concern or events that warrant further investigation. Some events might justify review by an administrator or content owner (for example, when a label is removed). Other events might justify being included in regular audit reviews (for example, when labels are downgraded).
- **Where to find this data**: Microsoft 365 administrators can use activity explorer in the Microsoft Purview compliance portal to view all sensitivity label activities.

Microsoft Purview content explorer

Content explorer in the Microsoft Purview compliance portal provides a snapshot of where sensitive information is located across a broad spectrum of applications and services.

> **Tip**
>
> It's not possible to see Power BI Desktop (.pbix) files in content explorer. However, you can use content explorer to see certain types of supported files that were exported from the Power BI service, such as Excel files.

- **What to look for**: You can use content explorer to determine what sensitive data is found in various locations such as Teams, SharePoint Online, OneDrive, and Exchange Online.

- **Actions to take**: Review content explorer when you need to gain an understanding of what content exists and where it resides. Use this information to assess the decisions you've made, and whether other actions should be taken.
- **Where to find this data**: Microsoft 365 administrators can use content explorer in the Microsoft Purview compliance portal to locate where sensitive data currently resides.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see the Power BI implementation planning subject areas.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Data gateways

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to plan and implement on-premises data gateways and virtual network (VNet) data gateways for Microsoft Fabric. It's primarily targeted at:

- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators might need to collaborate with Power Platform administrators, database administrators, information security teams, networking teams, and other relevant teams.
- **Gateway administrators**: The individuals who are responsible for implementing, managing, and monitoring gateways and their data source connections.
- **Gateway contributors**: The decentralized teams and individuals who are responsible for adding and managing gateway data source connections.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for supporting users that need to access, connect to, and refresh data.
- **Content owners and creators**: The teams and individuals who use gateways to connect to their data sources and refresh Fabric data items.

To access source data for Power BI semantic models, dataflows, and other Fabric data items, you might need a *data gateway*. A data gateway securely transfers data between private networks or on-premises data sources and cloud services, including Fabric.

> **Note**
>
> This article provides an overview of gateways. It focuses on key considerations and actions for planning and implementing gateways to support your Fabric content.
>
> For more information about how gateways work, see:
>
> - What is an on-premises data gateway?
> - On-premises gateway architecture

- Virtual network (VNet) gateway architecture
- Plan, scale, and maintain a business-critical gateway solution

Plan key decisions

Gateways are often integral to successful Power BI and Fabric implementations. A COE or central data and BI team typically plans and centrally manages gateways, although some organizations use decentralized teams to manage gateways. To mitigate the possibility of future disruption and governance risks, it's important to carefully plan how and when you'll use gateways.

You typically plan gateway implementation during two distinct stages.

- **Tenant setup**: When you prepare to implement, or migrate to, Fabric, you should assess whether any data sources need a gateway. Gateway planning activities are also relevant for tenant-level planning of security, workspaces, and auditing and monitoring.
- **BI solution planning**: When planning a new BI solution, you should evaluate whether the solution requires a gateway while you gather technical requirements for the new solution. You might also need a gateway when adding a new data source to an existing solution.

Planning a gateway implementation begins by making key decisions, starting with whether you need a gateway or not.

> **Tip**
>
> First, create an inventory of your data sources. You should evaluate the following key decisions for each data source. Be sure to document your results and store them in a central location that's easily accessible, like a **communication hub** or your **centralized portal**.

Identify whether you need a gateway

In general, you need a gateway for your data source when:

- Your data source is located on-premises.
- Your data source is located in a private network.
- You require a host for connector software.
- You require security isolation for certain connectors or functions.

In these situations, you need a gateway to:

- **Refresh data in the Fabric portal.** This scenario applies when a content creator sets up data refresh in the Power BI service for a data source that requires a gateway.
- **Create content in the Fabric portal.** This scenario applies when a content author creates or modifies data items (like a semantic model or dataflow) in the Power BI service that require a gateway.
- **Support DirectQuery connections.** This scenario applies when a content creator publishes a semantic model that includes DirectQuery (or Dual) storage mode tables, and the data source for those tables requires a gateway. This usage scenario also covers the ability to enforce per-user data permissions that are defined in the data source. For example, a SQL Server database can enforce row-level security (RLS), and Power BI can manage single sign-on (SSO) connectivity. For more information, see Enforce data security based on consumer identity.
- **Live connect to SQL Server Analysis Services.** This scenario applies when a content creator publishes a report that uses a live connection to a SQL Server Analysis Services (SSAS) database.

The following sections describe when you need a gateway.

On-premises data sources

You need a gateway to connect to on-premises data sources from the Fabric portal. The gateway serves as a bridge, evaluating query expressions on the gateway machine and securely transferring on-premises data to the cloud.

This scenario is relevant when connecting to:

- Data sources that reside on on-premises machines.
- Files stored in a local directory.
- Cloud and on-premises data sources that are combined in a single query.
- A cloud-based virtual machine (VM)—known as infrastructure as a service, or IaaS.

Private network data sources

You need a gateway to connect to data sources that are located in a private network, such as an Azure Virtual Network (Azure VNet). A virtual network, or VNet, is a logically isolated segment of a network that insulates traffic from the public internet. A VNet provides enhanced network security.

This scenario is relevant when the data source:

- Resides in a data center within a private network, such as the organizational network (or is behind a firewall).
- Is a cloud-based VM within a VNet (known as infrastructure as a service, or IaaS).
- Is a cloud-based database service within a VNet (known as platform as a service, or PaaS).

> **Note**
>
> It's a common misconception that you don't need a gateway for cloud data sources. When a cloud data source resides within a private organizational network, a gateway is required.

Host connector software

Sometimes you might need a gateway to host supporting items that are required to connect to your data source. This software might include custom data connectors, drivers, or libraries that you install on the gateway machine. The Power BI service doesn't have access to this software, so it's unable to refresh data sources that use it without relying on the gateway—even when you're connecting to a cloud data source.

This scenario is relevant when you connect to a data source with connectors such as a:

- **Driver.** An official connector could require installation of prerequisite drivers. For example, when connecting to an Oracle database, you might need the Oracle Data Access Client software.
- **Custom connector.** Some data sources might require custom or third-party connectors. For example, you might need a custom connector to connect to a legacy or proprietary system.
- **Client library.** Some data sources might require a supporting library to allow client tools to connect to it. For example, when connecting to an Analysis Services database, a client library must be installed.
- **ODBC or OLE DB connector.** An official connector might require an ODBC driver or OLE DB provider. For example, when connecting to SAP HANA, you'll need an ODBC driver.

> **Important**
>
> When content creators use a client tool such as Power BI Desktop and their solutions rely on drivers, connectors, or providers, you should install the same components on the gateway machine as are installed on the content creators machines. Missing or mismatched components between creator machines and data

gateways is a common reason for data refresh failures of published content. For more information, see User tools and devices.

Security isolation

You need a gateway to use certain Power Query connectors or functions, such as the Web connector or Web.BrowserContents function. These connectors and functions require a gateway for many reasons, including security isolation.

> **Tip**
>
> Consider the following alternatives when connecting to web page data sources.
>
> - **Web.Contents function**: If you're connecting to web content that doesn't need to be accessed through a browser, consider using the Web.Contents function. This function doesn't require a gateway because it doesn't use a browser control.
> - **Notebooks**: If you have Fabric capacity, consider using Fabric notebooks to transform data. Notebooks don't require a gateway for webpage data and they can perform better when retrieving webpage information, compared to Power Query.

This scenario is relevant when you connect to a data source by using connectors and drivers such as:

- A query that uses the Web connector.
- A query that uses the Web.PageorWeb.BrowserContents functions.
- A connection that uses the Access Database Engine (ACE) driver.

Determine which type of gateway you need

When you've identified that you need a gateway, you should next determine which type of gateway to install. There are three types of gateway.

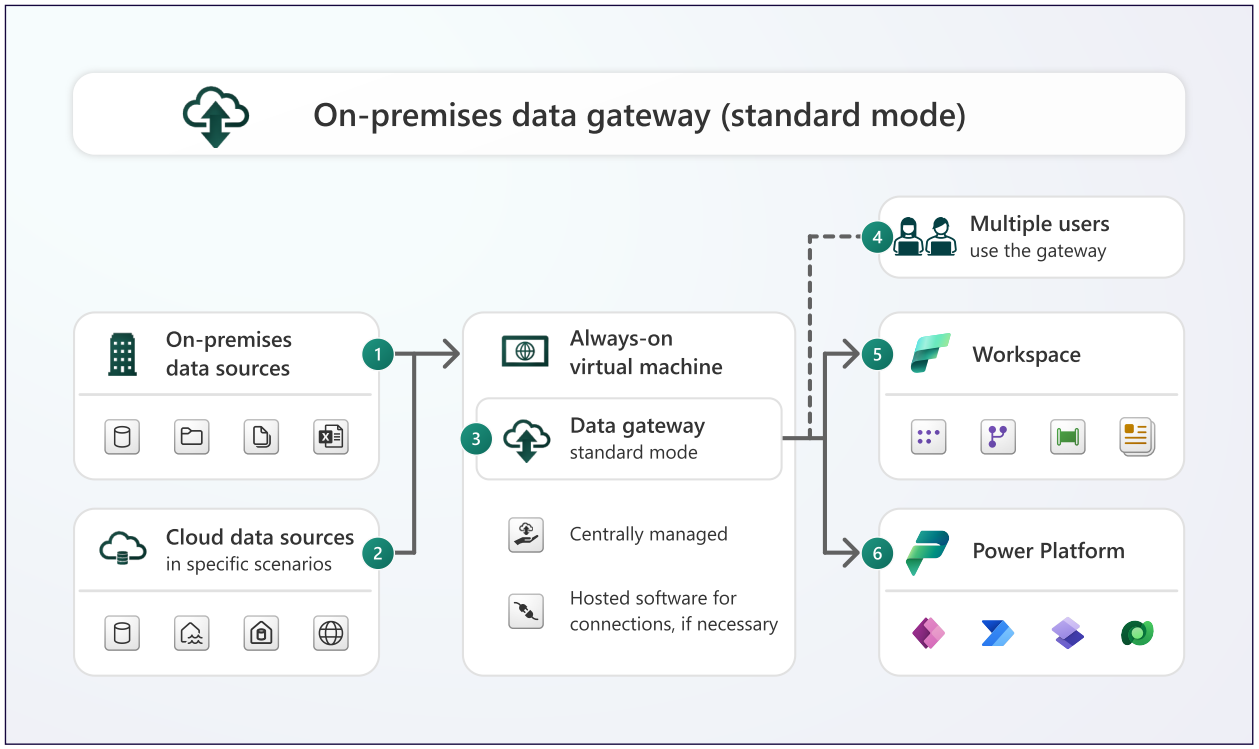
- On-premises data gateway (standard mode)
- On-premises data gateway (personal mode), known as a *personal gateway*
- Virtual network (VNet) gateway service

The gateway type you choose will depend on your requirements and the data sources. The following sections describe each of the three gateway types.

On-premises data gateway (standard mode)

An on-premises data gateway (standard mode) allows multiple users to connect to data sources that through a single shared gateway. Typically, you centrally install and manage standard mode gateways on an always-on VM. With a standard mode gateway, you can connect to data from multiple services, such as Fabric, Power BI, and other Power Platform services.

The following diagram depicts a high-level overview of a standard mode gateway.



> **Important**
>
> This diagram doesn't depict the architecture of an on-premises data gateway.

The diagram depicts the following concepts.

Expand table

Item	Description
1	The on-premises data gateway (standard mode) securely transfers data from on-premises data source to cloud services.
2	A standard mode data gateway is required for cloud data sources in specific scenarios (described in the previous section).
3	The standard mode data gateway is installed on an always-on VM. Administrators centrally manage the VM and data gateway. Gateway administrators install software required for

Item	Description
	data source connections, if necessary.
4	Multiple users can connect to the data gateway data sources.
5	Users can use the data gateway for items published to Fabric workspaces, like semantic models, dataflows, pipelines, or paginated reports.
6	Users can use the data gateway for other Power Platform cloud services, like Power Platform dataflows.

A standard mode gateway is required in the following specific situations.

- Different Microsoft cloud services (like Power Apps and Fabric) and Fabric data items (like dataflows) need to query on-premises data sources (or cloud data sources that require a gateway).
- Paginated reports need to query on-premises data sources (or cloud data sources that require a gateway).
- Semantic models use DirectQuery storage mode that needs to connect to on-premises data sources (or cloud data sources that require a gateway).
- SSAS live connections.
- Data sources depend on custom data connectors, drivers, or libraries.
- When you anticipate the need to relocate or migrate the gateway.

Personal gateway

An on-premises gateway (personal mode), commonly known as a *personal gateway*, allows a user to connect to on-premises data sources that reside on the same machine. A user typically installs and manages a personal gateway from their own machine. With a personal gateway, users can't connect to data from other Power Platform services. They also can't share the gateway or connections with other users.

A personal gateway is intended for limited, personal use by a single individual. Typically, content creators install and use these gateways to conduct personal BI. These gateways are limited to personal BI because they can't be shared. Additionally, a personal gateway requires that the user has machine rights and policy approval to download and install the personal gateway software.

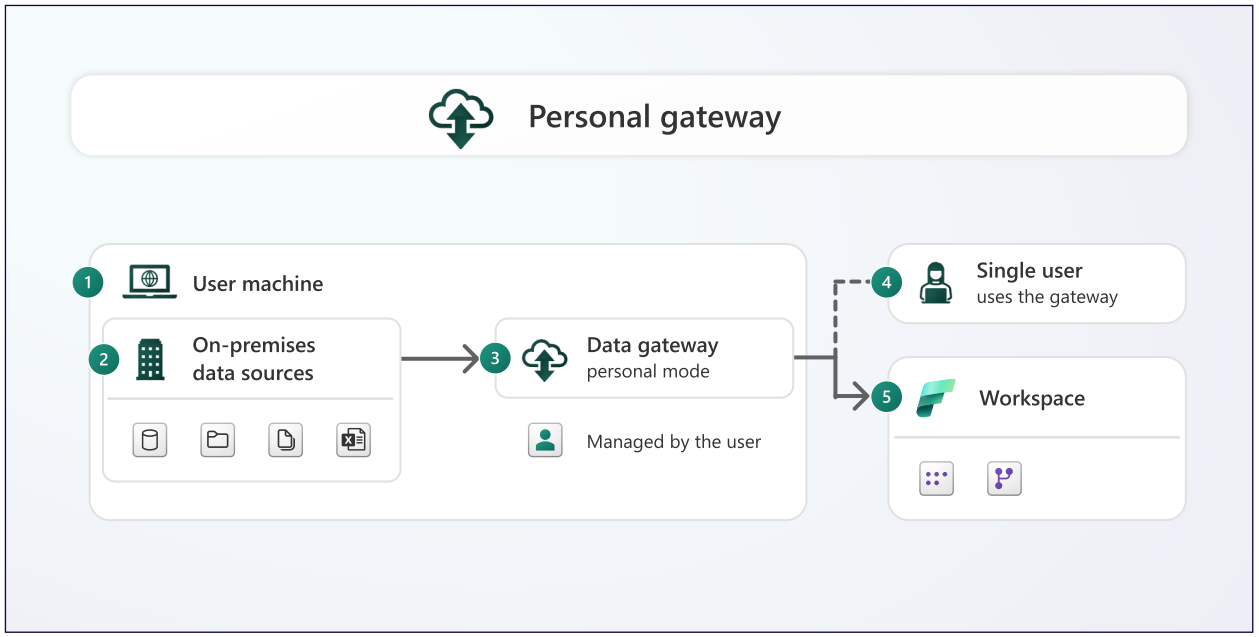
> **Tip**
>
> Don't use a personal gateway with **team**, **departmental**, or **enterprise BI** solutions.

For most scenarios where you connect to on-premises data, you should use a gateway in standard mode (described in the previous section). That's because you can share a standard mode gateway with multiple users, it supports DirectQuery queries and live connections, and it has more options to centralize gateway governance and management.

⊗ Caution

Because a personal gateway is typically installed on a user machine, it is more difficult to manage and govern. If you *do* need to use a personal gateway, consider moving it to a centrally-managed VM that uses a service account. This approach ensures gateway availability isn't dependent on a user machine (which might be turned off) and improves gateway governance and management.

The following diagram depicts a high-level overview of a personal gateway.



ⓘ Important

This diagram doesn't depict the architecture of an on-premises data gateway.

The diagram depicts the following concepts.

Expand table

Item	Description
1	A personal gateway is typically installed on a user machine.
2	The personal gateway securely transfers data from local data sources on the user machine to cloud services.
3	The personal gateway is typically managed by the user who installed it.
4	A single user uses the personal gateway for limited, personal use. A personal mode gateway can't be shared.
5	A personal gateway can only be used for items published to a Power BI workspace, like semantic models or Power BI dataflows.

To reiterate, a personal gateway is intended for limited, personal use by a single individual. However, there are two specific scenarios that require you to use a personal gateway.

- Self-service content creators need to refresh published content that's connected to local data sources on their machine or other on-premises data sources.
- Semantic models use Python or R code in Power Query.

> **Tip**
>
> Whenever possible, avoid using a personal gateway. Instead, consider the following alternatives.
>
> - **SharePoint**: If you need to connect to local files, consider uploading those files to SharePoint or OneDrive for Work or School instead. You can connect to these files by using the **SharePoint folder** connector, which doesn't require a gateway.
> - **OneLake**: If you need to connect to local files and you have Fabric capacity, you can also use **OneLake file explorer** to upload and synchronize files with a **lakehouse**. Connecting to a Fabric lakehouse doesn't require a gateway.
> - **Notebooks**: If you need to transform data with Python or R and you have Fabric capacity, consider using **Fabric notebooks** to transform the data and write it to tables stored in OneLake. Notebooks don't require a gateway to run Python or R code. You also benefit from the enhanced performance and additional features available in notebooks.

VNet gateway

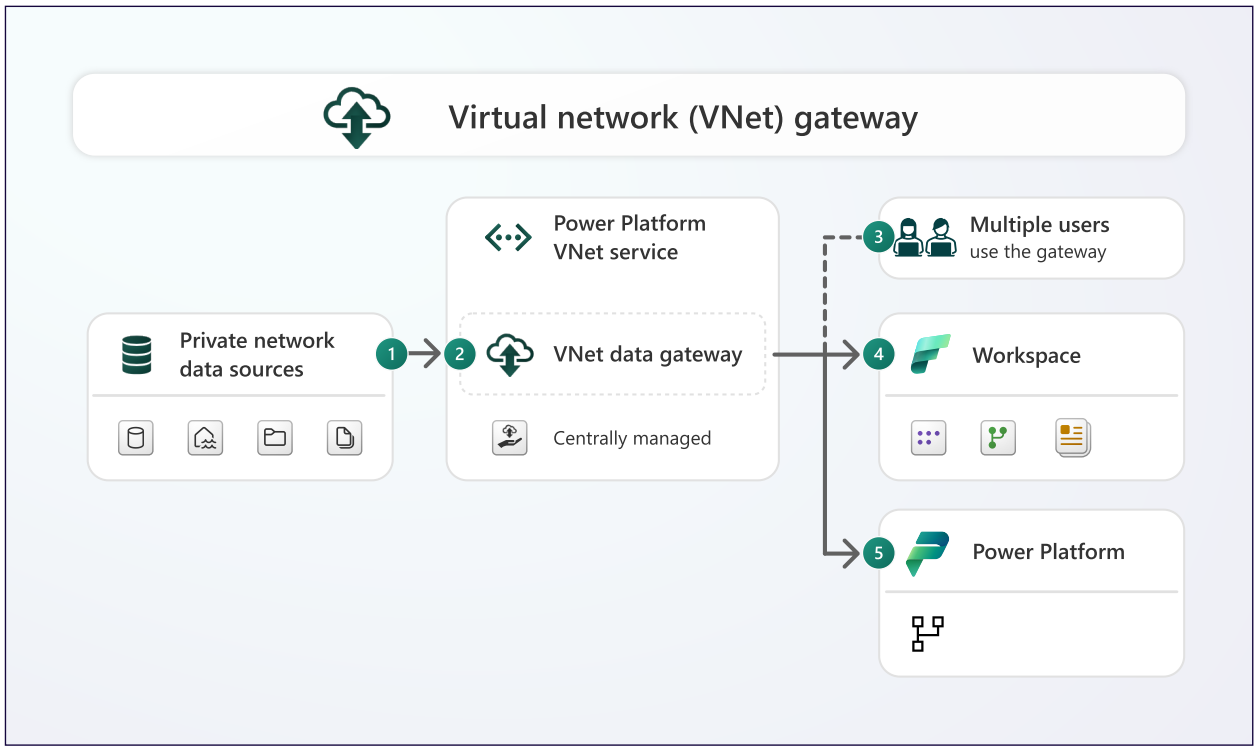
A VNet gateway allows multiple users to connect to data sources secured with private networks, including data sources that use private endpoints. With a VNet gateway, you can connect to data with multiple services, and you can share the gateway or connections with multiple users.

A VNet gateway is a Microsoft managed service. If your organization uses private networks, you need a VNet gateway.

> **Important**
>
> If you're considering using the VNet gateway service, discuss it with your IT teams that handle networking and security. These teams can help ensure that everything is set up, such as **private endpoints** (if applicable) and gateway **communication**.
>
> A VNet gateway is only supported for Power BI Fabric or Premium capacities. The VNet gateway is billed as an **additive premium infrastructure charge** for that capacity.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

The following diagram depicts a high-level overview of a VNet gateway.



> **Important**
>
> This diagram doesn't depict the architecture of a VNet data gateway.

The diagram depicts the following concepts.

Expand table

Item	Description
1	You use a virtual network (VNet) gateway is to connect to data sources on a private network, like those in an Azure VNet.
2	The VNet data gateway is a Microsoft managed service. You centrally manage the VNet data gateway from the Azure portal and the Power Platform admin portal.
3	Multiple users can use a VNet data gateway.
4	Users can use a VNet data gateway for items published to a Fabric workspace, like semantic models.
5	Users can use a VNet data gateway for other Power Platform services, like Power Platform dataflows.

> **Warning**
>
> VNet gateways have some limitations, and they don't support all data sources or scenarios. Validate that your data sources and scenarios are supported and consult

the frequently asked questions before you proceed with VNet gateway implementation and solution planning.

Determine the number of gateways

When you've determined that you need a gateway, and which type of gateways, you should next determine how many gateways you need.

Depending on your needs, you might require multiple gateways. Consider the following factors when deciding how many gateways to install and use.

Availability and performance

It's important that gateways have high availability to avoid disruption caused by refresh or query delays. One way to ensure gateway availability is to install multiple gateways in a high-availability *gateway cluster*. A gateway cluster is a collection of gateways that you install on different VMs, and that are logically associated together as a single, functional unit (the cluster). Each gateway machine is sometimes called a *node*.

Here are the benefits of using a gateway cluster.

- **Avoid a single point of failure**: Failover avoids a single point of failure when the primary gateway machine becomes unavailable. If it becomes unavailable, queries are sent to another node in the cluster. Using a cluster of multiple machines reduces risk. It also increases uptime, which helps you meet your high availability and disaster recovery requirements.
- **Better performance**: Load balancing improves performance when there's high concurrent usage. Load balancing distributes the workload by sending queries to other nodes in the cluster. That's helpful when the primary gateway is busy or when a single operation (like a long data refresh) consumes many resources.
- **Avoid downtime**: When installing gateway software updates, you can perform the installation on one node of the cluster at a time. That way, it avoids taking the entire cluster offline.

> **ⓘ Important**
>
> We strongly recommend that you use gateway clusters for business-critical workloads.

For more information and guidance on setting up a gateway cluster, see Plan, scale, and maintain a business-critical gateway solution..

Environments

Content creators typically use separate environments to develop and manage business-critical solutions, like development, test, and production. Depending on the number of environments you use and how you use them, you might want to have separate gateway clusters for each environment.

Separating gateway clusters into different environments can:

- Segregate and minimize disruption caused by development and test activities.
- Improve availability and performance of production workloads.
- Provide a safe environment to install and test gateway software updates.

> **ⓘ Important**
>
> We recommend that you have separate gateway clusters for production workloads. If you have one gateway cluster across all environments, that can represent additional risk. To minimize cost and management effort, it's common to allocate fewer resources (such as memory and CPU) to a development gateway cluster.

Regions

To ensure good performance of data refreshes, it's important that you consider the location of your data sources, gateways, and where your users are located. To reduce latency, you should install gateways as close to your data sources as possible. For this reason, you might need to install multiple gateway clusters to support different regions or tenants.

> **⊗ Caution**
>
> Ensure that your gateway installation complies with any data residency requirements for your organization.

> **ⓘ Important**
>
> To minimize latency, we recommend that you install gateways on machines that are in the same region as your data sources. Additionally, for VNet gateways, the gateways and data sources should be on the same subnet.

Checklist - When planning a gateway implementation, key decisions and actions include:

- **Create an inventory of data sources**: An inventory of your data sources allows you to verify and document which data sources need a gateway.
- **Determine which situations require a gateway**: Consider how content creators and consumers work. Ensure that you're familiar with when a gateway is required. Create documentation and training for the user community.
- **Decide on the type of gateway that you need**: Ensure that you validate any assumptions and assess possible limitations so that you're certain the selected gateway type meets your requirements.
- **Avoid personal gateways**: Consider using a gateway in standard mode instead. Determine whether there are any personal gateway data sources that can be redirected to use a standard mode gateway (so that it's not limited for use by a single individual).
- **Decide on whether you need a gateway cluster**: Use gateway clusters for business-critical solutions. Gateway clusters provide high-availability and load balancing. They also help to avoid a single point of failure and improve performance during periods of high concurrent usage.
- **Decide on how many gateways you need**: Consider separate gateway clusters for different environments to avoid disruption. Consider other factors, like usage or regions.

Install gateways

At this point, you know which types of gateway you need and how many. Next, you need to plan for the gateway installation. Typically, gateways are installed on VMs that you dedicate to this purpose (often called *gateway machines*). Each machine in the gateway cluster should always be on to ensure continuous support of user activity and data refresh operations.

> ⓘ **Note**
>
> Because a VNet gateway is a managed service, you don't download and install it. Instead, you provision and set up a VNet gateway in your Azure portal, and then

bind it to a Fabric or Power BI Premium capacity. For more information, see Create virtual network data gateways.

Identify gateway owners and installers

Before installing the gateway, identify who will install and own the gateway.

Gateway owners

Typically, the gateway owner is a technical person who installs, owns, and manages the gateway. Gateway owners are responsible for various activities.

- **Planning**: Make key decisions as described earlier and define the technical specifications for a gateway machine, including initial system resources. The gateway owner should also ensure that a support plan is in place.
- **Installation**: Select an appropriate machine to install the gateway software and perform first-time installation and setup.
- **Management**: Change gateway settings or preferences for optimization (such as configuring streaming instead of spooling data) or monitoring purposes (such as configuring performance logging). The gateway owner also makes decisions about when to scale up (add more resources to the gateway machines) or scale out (install more gateways in the cluster).
- **Testing**: Validate the gateway usage during first-time setup, ensuring that sufficient resources are available for the gateway machines. The gateway owner should also test monthly updates before installing them.
- **Updating**: Update and install the gateway software and supporting items (like connector software) on a timely basis.
- **Monitoring**: Monitor gateway uptime and health, including the collection of the gateway log files that allow monitoring for issues and anomalous activity.
- **Migration**: Store recovery keys in a safe place that's accessible to a wider team. The gateway owner should also be the person to use these keys to migrate, restore, or relocate the gateway, if necessary.

> **Important**
>
> Ensure that the gateway owner is aware of, and agrees with, these responsibilities. If the gateway owner isn't prepared to manage the gateway, it can quickly become a dependency that blocks content owners and creators. Additionally, identify whether the gateway owner understands how to install and manage a gateway and, if not, how you'll train them to do this.

> **Tip**
>
> Some organizations successfully allow gateway ownership within business units and departments, whereas others reserve gateway ownership for a centralized team (such as IT). One way to handle it is to form a partnership where IT manages the gateway cluster nodes, and the business unit manages the data source connections.

Because gateway ownership is an important responsibility, you should clearly define who can install gateways in your organization.

Gateway installers

To reduce management overhead and mitigate governance risk, it's important to limit the number of active gateways in your organization. To this end, we recommend that you limit the number of users who can install gateways.

> **Warning**
>
> Gateway owners have full control over the gateways that they manage. This means that malicious gateway owners could potentially intercept information as it flows through an on-premises data gateway. For this reason, it's essential that you restrict the ability to install gateways to trusted people.

For standard mode gateways, you manage gateway installers either from the Fabric portal or the Power Platform admin center. You also manage who can create VNet data gateways by using the gateway installer setting.

- **Fabric connections and gateways page**: You can manage gateway installers from within the connections and gateways page of the Fabric portal.
- **Power Platform admin center**: You can also manage gateway installers from the Power Platform admin center. Settings you change here affect gateways you use from Fabric.

You can also manage gateway installers programmatically by using the PowerShell cmdlets for on-premises gateway management. For personal gateways and standard mode gateways, you can use these cmdlets to set the gateway tenant policy. Setting the gateway tenant policy by using PowerShell is the only way to manage who can install personal gateways in your tenant.

> **Important**

We recommend that you closely regulate who can install personal gateways, limiting their installation and use for valid, approved business cases.

Prepare gateway installation

When you've identified who will install and own the gateway, you should prepare for the gateway installation. You should:

- Identify where to install the gateway.
- Decide on the resources that the gateway machine needs.
- Agree on how you'll name your gateway when it's installed.

The following sections describe these key considerations for planning a gateway installation.

Identify where to install the gateway

Typically, you install a gateway on an always-on VM (also called the *gateway machine*). You can install only one gateway of each type (personal mode or standard mode) on a machine.

Here are the key factors for determining where you'll install a gateway.

- **Location**: Typically, the gateway machine should be located close to the data source to minimize latency. Usually, a standard gateway needs to be installed in your default data region. However, if your Premium capacity location is different from the default data region for your tenant, investigate using an Azure Relay as an option to satisfy data residency requirements.
- **Supporting items**: Determine what connectors, drivers, or libraries you need to install on the gateway machine.
- **Domain**: Determine what the relationship of the gateway machine is with the target domain. The VM must be a domain-joined machine with a trust relationship with the target domain. It can't be a domain controller.

Tip

To avoid resource contention, don't install unrelated software on a gateway machine. The gateway machine should be fully dedicated to hosting the on-premises data gateway.

Decide on the gateway machine resources

The gateway machine should have sufficient resources to handle the expected query workload.

Here are the key factors for determining gateway machine resources.

- **Usage**: Determine how many and which type of items will use the gateway, and what query concurrence (from many users) will be. Higher usage requires gateway machines with more resources.
- **Connection type**: Determine whether Power BI semantic models import data, use DirectQuery, or a live connection. For import semantic models, it's important to check the number of data refreshes to estimate gateway resource needs (such as RAM). For DirectQuery or live connections, you should assess the number of report consumers to estimate resource needs (such as CPU).

> **Tip**
>
> Validate the gateway machine resources by performing load testing. You can conduct this type of test by monitoring gateway machine health when performing dataset refreshes, and by simulating high concurrent usage of DirectQuery or live connection reports.

Agree upon naming conventions

How you'll name the gateway and its data source connections is important. The name should make it straightforward for content creators to know what to connect to. To ensure gateways and data source connections have clear names, you should use a logical naming convention.

When you define your naming conventions, consider the following points.

- Include a variation of *Gateway* or *DataGW* in the name to identify the gateway for auditing, logging, and troubleshooting purposes.
- Include the specific purpose of the gateway when it supports specific Fabric items, operations, regions, or business areas.
- Use a variation of *Dev*, *Test*, or *Prod* in the name when the gateway supports a specific environment.
- Give the gateway a name that aligns with the name of the cluster it belongs to. For example, give each gateway machine within the cluster a unique name, like *Node1*, *Node2*, and so on.

Here are some examples of logical gateway names.

- *DataGW-Prod-Node1*
- *Gateway-DevTest-Node1*
- *Gateway-FinanceTeam-Prod-Node1*

Install and set up gateways

After making key decisions and preparation, the gateway owner installs the gateways and performs first-time setup.

> ⓘ **Note**
>
> For information on how to download and install a gateway, see:
>
> - Download and install a standard mode gateway
> - Download and install a personal mode gateway

When installing and setting up gateways, consider the following factors.

- **Installation location**: When you want to install the gateway to a location other than the default installation path, you can change the installation location.
- **Recovery key**: When you want to migrate, restore, or takeover an existing gateway, you need to use the gateway's recovery key. Ensure that you keep the recovery key in a safe and secure place that's accessible to other gateway administrators.
- **Data center region**: When you want the region to be different from the registered service's tenant, you can change the data center region.
- **Azure Relay**: When you want to use your own relay instead of the default, you can provide your own relay details.
- **Proxy settings**: When your work environment requires the gateway to go through a proxy server to connect to the Fabric portal, you must set up proxy settings.
- **Gateway service account**: When you want to use an explicit domain account, you can change the gateway service account from the default, which is *PBIEgwService*.
- **Communication settings**: When a firewall blocks outbound connections, your security and networking teams can set up the firewall to allow outbound connections from the gateway to its associated Azure region.
- **Tenant registration**: When you want to restrict which tenants are allowed to register the on-premises data gateway application to prevent data exfiltration.
- **Integration tenant settings**: When you want to ensure that your gateway operates with single sign-on (SSO) (for instance, with Microsoft Entra ID-based authentication) the way you intend.

> **Important**
>
> We recommend that you restrict tenant registration to only tenants within the organization. This step helps to improve gateway security because the default setting has no restriction on tenant registration.

Checklist - When preparing for and installing a gateway, key decisions and actions include:

- **Identify the gateway owner and installers**: Ensure the gateway owner is aware of their responsibilities. Restrict gateway installation to the appropriate people.
- **Conduct training**: If needed, train gateway owners and installers to effectively install, manage, and support the gateway. Conduct cross-training for a backup when necessary.
- **Create naming conventions**: Create gateway naming conventions that correspond to purpose, environment, cluster node, and the use cases it supports or the operations it performs.
- **Consider resource needs**: Determine what the workload and usage will be to determine initial resources (such as memory and CPU).
- **Set the integration tenant settings**: Review and set the integration tenant settings to ensure your gateway operates with single sign-on (SSO) the way you intend.
- **Provision the gateway machine**: Set up an always-on VM with sufficient resources to support gateway operations.
- **Install the gateway**: Perform first-time setup of the gateway on the gateway machine.
- **Install supporting items**: Install custom data connectors or dependent software to support your scenario.

Manage gateways

After installing the gateways, you should then add data source connections. When adding these connections, you should also plan how you'll manage access to the gateway and its connections.

Add data source connections

You must add the initial data source connections before you can use the gateway. You can add connections manually from the Power BI service or the Power Platform admin center, or programmatically with the Power BI REST APIs.

When adding connections, consider the following points.

- **Stored credentials**: Consider which credentials will be used to connect to the data source. When you add a connection, you must provide credentials for that data source (unless it supports anonymous authentication). It's an important decision, because all queries to the data source run by using these credentials, unless you're using Microsoft Entra single sign-on (SSO) for the data gateway.
- **Naming conventions**: Like gateways, connections should also use logical and consistent naming conventions. Ensure that connection names correspond to the data source name. For example: *FinanceDB-Prod* is a logical name that indicates the data source.
- **Single sign-on**: In the Fabric administrator settings, you should enable the Microsoft Entra single sign-on (SSO) for gateway option when you want to use single-sign on (SSO) with DirectQuery (either by using Active Directory SSO or Microsoft Entra SSO). You should use SSO when you want to enforce data security in the data source system based on the report user identity.
- **Privacy levels**: For import data source connections, you must set the privacy level. Ensure that you select the appropriate privacy level to appropriately isolate the data source, if necessary. It's important to understand that the privacy levels set in Power BI Desktop aren't honored by gateways.

> **Note**
>
> The data source name can be modified afterwards, but the server and database names can't be changed after they've been setup. To avoid errors, ensure that the data source information **matches what will be used in Power BI Desktop**.

> **Tip**
>
> To improve efficiency and accuracy, consider automating the creation of data source connections by using the **Power BI REST APIs**. In this case, we recommend including review and approval processes rather than automatically processing each request that creates or updates a connection.

Provision gateway access

After adding the initial data source connections, you should decide on how to manage access to both the gateway and its connections.

Content creators will need access to a gateway connection to successfully connect to a data source. User access to gateway connections is done for each connection, so consider who needs access to each gateway connection, and how you'll manage that access. You should manage access by using security roles for both gateways and connections.

Gateway roles

Gateway roles let you control who can manage the gateway and its data source connections. These roles work similarly to workspace roles, allowing different permissions depending on the role. Using roles helps you to manage gateway access more effectively.

> **Tip**
>
> We recommend using **security groups** to manage role membership instead of individual accounts. That way, it's easier to manage users, particularly across multiple gateways. You can use the same security groups to manage other access control, like **row-level security** role membership and **app audience** membership.

> **Important**
>
> A user who just needs to use the gateway to connect to a data source doesn't need to belong to a gateway role. In this case, they'll only have the *User* connection role.

There are three gateway roles for managing an on-premises standard gateway.

- **Admin**: Members of this role can manage and update the gateway. A gateway administrator is typically the gateway owner, but there can also be multiple administrators for a gateway. Gateway administrators should be Fabric administrators, or members of the COE or central BI team. The responsibilities of an administrator are the same as a gateway owner.
- **Connection creator with sharing**: Members of this role can create gateway connections, test the gateway status, and share the gateway with others. This role can't remove users from the gateway. Consider adding someone to this role when they're responsible for a subset of analytical solutions, like in a decentralized team for a business unit. The responsibilities of someone with this role include:

    - Setting up and testing new connections.
    - Managing connections that they own, such as setting credentials.
    - Sharing the gateway with other users who need it.
    - Regularly reviewing who has access to a gateway, validating whether they still need it, and removing them when they don't need it.
- **Connection creator**: Members of this role can create connections on the gateway and test their status. The connection creator should be a content owner who can appropriately set up the right connections to use the gateway. The responsibilities of a *Connection creator* role are the same as the *Connection creator with sharing* role, with the exception that they can't share access to the gateway.

> **Note**
>
> VNet gateways only support the *Admin* gateway role.

Data source connection roles

Data source connection roles let you control who can use, manage, and share connections. A user with a connection role doesn't need to belong to a gateway role.

There are three data source connection roles.

- **Owner**: Members of this role can manage the connection or delete it when it's no longer required. Owners can manage the connection roles, including adding other connection owners. An owner is typically also a connection creator. Consider making someone a connection owner when they're the steward or administrator of that data source, or they have significant knowledge about the data source and its contents. The responsibilities of an owner include:
    - Managing connections, such as updating credentials, if necessary.
    - Deleting the connection when it's no longer needed.
    - Managing connection roles from the Power Platform admin center.
- **User with sharing**: Members of this role can use and share the data source by adding other users. Consider adding someone to this role when they play an important role in the user community. Champions can be good candidates for this role. The responsibilities of someone with this role include:
    - Sharing the connection with other users who need it.
    - Regularly reviewing who has access to a connection, validating whether they still need it, and removing them when they don't need it.
- **User**: Members of this role can use the data source in Power BI reports and Power BI dataflows. Users are only responsible for querying the data from their workloads and client tools.

> **Tip**
>
> To prevent governance risk from oversharing, you should limit who can share gateways and connections to specific individuals who can accomplish this task effectively and responsibly.

Document gateway and connections

After you've set up your gateway cluster, you should document it. You should document your gateways so that they're easy for content creators to find, and easy for gateway administrators to maintain. Consider storing gateway documentation in an accessible location, like your centralized portal of the relevant community of practice.

Consider documenting the following information.

- Gateway name and GUID
- Gateway machine name (and any respective identifiers)
- Gateway owner
- Last software update date (gateway version)
- Purpose of the gateway cluster (such as the environment, region, business area)
- Which data source connections should be maintained on this gateway
- Whether the gateway uses a user identity or stored credentials
- Access management policy (how you intend to use gateway roles and connection roles)

> **Important**
>
> Ensure that you document your gateway **recovery keys** for standard mode gateways. These recovery keys are required should you ever need to recover or re-locate the gateway. Keep this information in a safe and secure place that's accessible to multiple trusted people in a central team. If you have an organizational password vault, that's an ideal location.

Update gateways

To ensure your gateways remain functional and perform well, you need to perform several tasks.

- Install Windows updates and perform other server maintenance.

- Update the gateway software. The gateway software is updated monthly, and Microsoft only supports the last six releases of the on-premises gateway.
- Update custom data connectors, drivers, and libraries, when necessary.

> **Note**
>
> The gateway owner must manually apply gateway updates to each gateway. For this reason, it's important to plan a process to update your gateways periodically.

> **Tip**
>
> When you work with the Power Query Online experience, the **Power Query** engine uses the latest version of Power Query that's available. However, when you're using a gateway to apply transformations, it uses the version that's installed on the gateway machine. To ensure a consistent user experience, it is important to keep your gateway machines up to date.

The remainder of this section describes how to update the gateway software.

Install updates on development or test gateways

It's important to keep your gateways up to date to avoid unexpected disruption, and also to benefit from the latest improvements. Nonetheless, updates can have unexpected consequences for your gateway performance and function. To avoid impacting business-critical solutions, you should first install gateway software updates to a development or test gateway (before they're applied to gateways that support production environments).

Validate updates

You can test gateways by first applying the update to gateways that support development and test environments.

Consider the following points when validating gateway updates.

- **Define repeatable test conditions**: You should define a list of repeatable test conditions to help you ensure that you test all the relevant gateway operations and data sources. For example, you could identify which reports and semantic models are deemed critical and require validation. You might also have some compliance requirements to be met that qualify as repeatable test conditions.

- **Use a set of test reports**: Keep a set of reports to use for functional testing each time the gateway is updated. These reports help you quickly validate your repeatable test conditions. These test reports often only show totals and counts. Your goal is to ensure that you test access, rendering, and refresh for:
  - Each commonly used data source.
  - Each key type of data item, like the most critical semantic models.
  - Different storage modes, like import and DirectQuery.
- **Identify business-critical reports**: Have access to (or copies of) business-critical reports that you can test for the new update. These reports can help you ensure that the data can refresh and DirectQuery reports work as expected.
- **Automate testing processes**: Use the Power BI REST APIs to test the data refresh for import data items and to evaluate DAX queries. Ensure that you can catch and log refresh failures or query errors.

> **Important**
>
> We strongly recommend that you test gateway updates on the development and test cluster before applying them to production. Testing updates is important, as there's no rollback process. As an alternative, before starting the update you can create a **VM image**, which is a complete copy of the file system structure and the data on the machine.

Install updates on production

After validating the gateway update, you should apply the update to all gateways that support production environments. A gateway is unavailable while it's being updated, so you should be consistent in when and how you update your gateways.

Consider the following points about updating gateways.

- In your centralized portal, document the current version of the gateway.
- Apply updates during periods of historically low gateway usage.
- Use a consistent a consistent schedule when you test and apply updates. If monthly updates are too often, ensure that your gateways are updated at least quarterly.
- Update one gateway cluster machine at a time. By rotating through each machine, you can avoid downtime.

Update credentials

For data source connections that require stored credentials, you might need to rotate the credentials regularly. For example, your organization might have a policy that requires frequent password resets. This practice is also useful when a key team member leaves the organization. To improve efficiency, you can use the Power BI REST APIs to update credentials.

Checklist - When managing data gateways, key decisions and actions include:

- **Create data source connections**: Set up the data source connections for common organizational data sources. Ensure that connections follow clear, consistent naming conventions.
- **Document gateways and data sources**: Create concise documentation about gateways and connections. Ensure this documentation is easily accessible from your centralized portal for gateway owners and administrators.
- **Handle connection requests**: Create a process to gather and manage connection requests. Determine whether an approval process is required for connection requests. Consider automating the process by using the Power BI REST APIs.
- **Provision gateway roles**: Use the principle of least privilege to assign individuals to gateway roles. Consider adding data source stewards to the *Connection creator* or *Connection creator with sharing* role so that they can contribute to connection management.
- **Provision connection roles**: Assign content creators (and consumers when applicable) to connection roles so they can use the gateway. Limit users with sharing to those users who will responsibly share the connection and help to regularly review and manage access.
- **Create concise user documentation**: Document the key items important for content creators to find and use the gateway and its connections. Place the documentation somewhere central and easily accessible to the user community, like a centralized portal or SharePoint support site.
- **Carefully document and store recovery keys**: Store recovery keys in a safe, secure location accessible to more than one team member. Ensure that they can be easily found should you need to recover the gateway.
- **Create a process for installing updates**: Determine how often you'll install gateway software updates and the process to be followed. Aim to update gateways within one to three months of the update release.
- **Install gateway updates first on development and test**: Ensure that development and test gateways are updated prior to production and used for initial testing.

- **Test gateway updates before they're applied to production gateways**: Set up a process to test monthly gateway updates by using repeatable test conditions and items.
- **Install gateway updates promptly and regularly on production**: Ensure that production gateways are kept up to date.
- **Update connection credentials**: As needed, update stored credentials use by connections.

Monitor, audit, and optimize gateways

Gateways are a crucial part of Fabric data integration. To avoid disruption and mitigate risk, you should monitor and regularly audit the gateways in your tenant.

Monitoring gateways helps to:

- Mitigate governance risk.
- Prevent issues.
- Investigate incidents, such as performance issues or data refresh failures.

The following table provides an overview of the typical issues you might encounter when managing a data gateway cluster, and how to monitor or investigate issues.

Expand table

Potential issue	Type of issue	How to monitor or investigate the issue
Too many gateways	Governance	• Power Platform admin center • Gateway PowerShell cmdlets • Power BI REST APIs
Gateway oversharing	Governance	• Power Platform admin center • Gateway PowerShell cmdlets • Power BI REST APIs • Power BI activity log
Gateway is offline	Performance and availability	• Power Platform admin center • Gateway machine monitoring • Gateway logs

Potential issue	Type of issue	How to monitor or investigate the issue
Gateway errors	Performance and availability	• Gateway logs
Query failures	Performance and availability	• Gateway logs • Gateway logs (additional logging)
Slow refreshes or queries	Performance and availability	• Gateway machine monitoring • Windows event log • Windows performance counters • Gateway logs • Gateway logs (additional logging) • Server monitoring tools

> **Note**
>
> For more information about auditing and monitoring gateways, see:
>
> - Monitor and optimize on-premises data gateway performance
> - Power BI implementation planning: Data-level auditing
> - Power BI implementation planning: Tenant-level auditing

> **Tip**
>
> If you use Fabric capacity, the tools in Fabric can provide the ideal components for you to build and orchestrate an organizational gateway monitoring solution. For instance, you can:
>
> - Use Data Factory to copy gateway logs and land them in OneLake.
> - Use notebooks to programmatically gather gateway information by using the Power BI REST APIs and transform gateway log files to tables.
> - Use Power BI to create a semantic model and report on gateway health.
> - Use Fabric Activator to send alerts to gateway owners and administrators when there's anomalies or disruptions.

Monitor gateways

You should regularly review how many gateways are installed on your tenant, and who installed them. You can monitor prevalence from the connections and gateways page of the Fabric portal, and the Power Platform admin center. Both views provide a concise, functional overview of all the gateways that you have access to. Administrators should review this information regularly.

> **Note**
>
> You can also programmatically retrieve a list of gateways by using the PowerShell cmdlets or by using the Power BI REST APIs. You can also identify gateway installation events by using the activity log.
>
> Consider combining this information with aggregate analytics about the number and type of gateway data sources. You can present this information in consolidated, tenant-wide governance or auditing and monitoring reporting.

When monitoring gateway prevalence, focus attention on the following metrics.

- **Rising number of gateways or installers**: Ensure that you investigate unexpected new gateways or installers.
- **Redundancy in connections across gateways**: Try to consolidate connections to avoid extra maintenance effort of the gateways.
- **Unexpected installers or gateways**: Ensure that any new installers or gateways have undergone an approval process before they're installed.
- **Unexpected gateway machines, connections, or configuration**: Ensure that you identify any anomalous gateway properties such as gateways installed on user machines, or connections to local files. Also identify settings that create risk, like ignoring privacy level.

Collect and analyze gateway logs

Each gateway machine produces detailed logs that you can use to identify and resolve issues. These logs are a collection of detailed technical files stored on the gateway machine. You can also temporarily enable additional logs from the on-premises gateway app to collect more detailed information about queries and their timings.

To avoid introducing networking delays, we recommend that you allow the gateway logs to be written to the local gateway machine. However, you can choose to change the path where the logs are written with the gateway configuration files. You can also

update how long the logs are retained. Always make a copy of the configuration files before you edit them.

Create a data integration solution to gather and consolidate the log files from each gateway machine so you can analyze the data. Ideally, this process should be automated and output to an analytical report to easily view all gateways and identify anomalies.

> **Tip**
>
> Consider using data alerts to notify gateway administrators and data source connection creators about anomalous activity for their gateways and connections, respectively. That way, they can take immediate corrective actions.

Monitor gateway machine health

The health of your gateway is dependent on the health of its server. To avoid disruption, you should monitor gateway machines to detect when a machine isn't performing well or is offline.

> **Tip**
>
> Ensure that your gateway machine is added to any enterprise monitoring tools that your organization uses to monitor servers.

Optimize and troubleshoot gateways

When issues arise with a gateway, you'll need to investigate and identify the problem. Typically, troubleshooting involves investigating the gateway logs described in the previous section, and testing various optimizations to see whether they resolve the issue.

Here are some common gateway optimizations.

- **Change from spooling to streaming**: By default, gateways will spool data to the gateway machine when evaluating a query. It results in the temporary storage of data before it's transferred to the cloud. Spooling can be slower than the alternative of streaming, where data is transferred directly. You can change this setting in the gateway configuration files.
- **Antivirus scanning**: Excluding certain folders from antivirus scanning on the gateway machine can improve performance when using file-level antivirus

software.

- **Planning to scale up or scale out**: Consider when you'll scale up a gateway cluster by adding more resources to the gateway machine, or scale out the cluster by adding another gateway to another machine.

> **Important**
>
> VNet gateways have a single hardware configuration, which can't be scaled or changed.

> **Note**
>
> For more information about optimizing and troubleshooting gateways, see:
>
> - Troubleshoot the on-premises gateway
> - Troubleshoot a VNet network gateway
> - Plan, scale, and maintain a business-critical gateway solution

Checklist – When monitoring gateways, key decisions and actions include:

- **Monitor gateway machines**: Ensure gateways are installed on machines that are monitored by enterprise monitoring solutions. Otherwise, ensure that you can detect when these machines aren't performing well.
- **Measure gateway prevalence**: Monitor the evolution of how many gateways are installed in your tenant over time.
- **Gather and analyze gateway logs**: Create a solution to automatically collect and combine the log files from the different gateway machines. Analyze these files to extract meaningful information. Consider setting up two types of analytical monitoring solutions: one for alerts and action, and another for exploratory root cause analysis when issues arise.
- **Verify roles and responsibilities**: Ensure that roles and responsibilities are defined for monitoring, optimizing, and troubleshooting.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Integration with other services

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to plan how and when to integrate Power BI and Microsoft Fabric with other services. This article is primarily targeted at:

- **BI and analytics directors and managers**: Decision makers who are responsible for overseeing the BI program and strategy. These individuals decide whether to use other services to support specific strategic objectives, or to complement Fabric or Power BI.
- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators control which services can integrate with Fabric by enabling Integration tenant settings, and they set up tenant-level integration with services in Azure or Microsoft Teams. Often, Fabric administrators need to collaborate with other administrators to facilitate this integration.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams look for opportunities to use services that when integrated help people solve problems or use Power BI more effectively.
- **Content owners and content creators**: The teams and individuals that champion analytics in a team or department. These teams and individuals perform workspace-level and solution-level integration to support specific needs and use cases, where permitted.

When you use Power BI, you might experience certain needs or challenges that you can't address with the core Power BI tools and features. In these situations you can consider integrating Power BI with other services. Most of these services are Microsoft services, like Azure or Microsoft 365, but you can also integrate Power BI with custom or third-party services. Extending the functionality of Power BI in these ways can help solve new problems, and it allows people to become more effective with their regular tasks.

Here are some common scenarios involving the integration of Power BI with other services:

- You have specific requirements that mandate the use of another service. For example, you must integrate with Azure Private Link to connect to services over a private endpoint in your virtual network.
- You encounter specific challenges that can't be overcome with Power BI alone. For example, you use Azure Log Analytics integration to obtain detailed query diagnostics of your semantic models for troubleshooting and auditing.
- You want to use services that you already use, or extend the capabilities of Power BI. For example, you can allow Excel users to connect to semantic models by using the Excel add-in to insert connected PivotTables.

You can integrate Power BI with other services at the level of your tenant, your workspace, or individual solutions (like semantic models and reports):

- **Tenant-level integration**: Affects the entire tenant and is set up by Fabric administrators, usually in collaboration with other administrators. For example, Teams integration is set up at the tenant level. Another example that affects networking is Azure ExpressRoute.
- **Workspace-level integration**: Affects all content in the workspace and is set up by workspace administrators. For example, Git integration is set up at the workspace level to achieve source control with Azure Repos, which is a service of Azure DevOps.
- **Solution-level integration**: Affects a single content item and is set up by the content creator. For example, Python or R is set up at the solution level to enable the creation of custom, interactive visuals.

For all three of these levels, there are considerations you should keep in mind when you integrate Power BI with other services:

- **Security considerations**: Integrating other services inevitably results in more risks that you must mitigate to use them successfully. For example, integration with AI services has the potential to expose internal data to external services that train their models. To mitigate this risk, ensure that you proactively evaluate any security risks and considerations of integrating a service. Also, identify concrete actions to ensure compliance with data security and privacy policies in your region and organization.
- **Licensing considerations**: Integrating other services might require a specific subscription or license. For example, integrating Power BI reports with PowerApps is only possible when you have the appropriate PowerApps licenses. For each service, ensure that you evaluate whether you need a specific license or

subscription to integrate it, and what the estimated cost is per user or capacity. Do this evaluation not only for the services, but also for Fabric and Power BI per-user and per-capacity licenses.

- **Governance considerations**: Integrating other services results in more diverse activities and operations that people undertake in your tenant, some of which might lead to inappropriate practices. For example, integration of Power BI reports with OneDrive or SharePoint might lead to people sharing Power BI Desktop (.pbix) files directly with report viewers. This approach diverts from the better practices of publishing the report to a workspace and sharing it via direct access, workspace viewer roles, or a Power BI app. Therefore, you should proactively identify any potential governance risks before you integrate a service, and identify the effort needed to monitor and support the service in your tenant.
- **Mentoring and user enablement considerations**: Integrating other services might require time and effort to train users to use any new capabilities effectively. For example, if you allow users to integrate Excel with Power BI, you should train users about how to effectively use Analyze in Excel. Training should guide them on when to use it, and inform them of its considerations and limitations. Ensure that you proactively plan how to train and support people that will use this integration.

The remainder of this article describes the possibilities to integrate Power BI with other services at the level of your tenant, workspaces, and individual solutions (like reports or semantic models).

> **Note**
>
> This article provides an overview of the different services that you can integrate with Power BI, and the potential use cases to do so. The purpose of this article *is not* to guide you in the technical steps required to set up or troubleshoot the integration. You will find links to technical information in each respective section of this article.

Tenant-level integration

Fabric administrators can integrate some services for use across the entire tenant. Typically, this integration facilitates broader interoperability between Fabric or Power BI and related services, like those available in Azure. Tenant-level integration can also affect how certain data is handled.

> **Important**

For an overview of the relevant administration settings a Fabric administrator can use to control integration of Microsoft Fabric or Power BI with external services, see **Integration tenant settings**. A Fabric administrator can control integration with services across all levels with these tenant settings.

Integration with Azure services

You can integrate your tenant with a wide range of Azure services that you might already use to store or manage your data. This integration helps you apply the scope and benefits of Azure services from within Fabric and Power BI. It also enables more advanced capabilities that can support many roles, from administrators and centralized teams to decentralized content owners or creators.

Integrating with Azure services requires that you have an active Azure subscription for them. Additionally, there are some specific licensing considerations for this option. Using sensitivity labels and DLP policies requires an Azure Information Protection Premium P1 or Premium P2 license. Users might require a Power BI Pro or Premium Per User (PPU) license to use features resulting from this integration, like applying sensitivity labels. Finally, some of these services also require that you have Fabric or Premium capacity, and they might use your capacity resources.

For guidance on how to integrate with Azure services, see:

- Azure Information Protection for sensitivity labels and data loss prevention (DLP) policies
- Azure Synapse Analytics
- Azure Databricks
- Databricks Unity Catalog
- Azure HDInsight
- Azure Automation

While not necessarily Azure services, you can also use the following tools available for tenant-level integration with Power BI:

- PowerShell
- Azure Storage Explorer

Integration with AI services

Apart from Copilot in Fabric, there are various AI services that you can integrate with Fabric and Power BI. These services can help you perform advanced analytics to apply specific models to your data, depending on your needs and use cases.

Integrating with AI Azure services requires that you have an active Azure subscription for them. Additionally, some of these services also require that you have Fabric or Premium capacity, and they will use your capacity resources. To ensure that these workloads don't have a negative impact on your capacity utilization, ensure that you set a memory limit for AI workloads within your capacity. That way, you can avoid unexpected usage of your capacity units (CUs). For more information, see Manage impact on a Premium capacity.

For guidance on how to integrate with the different AI services in Azure, see:

- Azure OpenAI Services
- Azure AI Services (Text Analytics and Vision)
- Azure AI Translator
- Azure Machine Learning

Integrate Azure AI Services in Power Query

You can invoke specific AI functions in Power Query by using Azure AI Services. These functions run by using Fabric capacity or Premium capacity for a selected workspace. They can derive useful information from less-structured text or image data.

Use cases for Azure AI Services integration with a semantic model or dataflow include:

- Language detection from text in a field.
- Key phrase extraction from text in a field.
- Sentiment analysis of text from input fields.
- Image classification of images of recognizable objects, entities, scenes, or actions.

Integrate Azure Machine Learning in Power Query

Similarly to how you can use Azure AI Services, you can apply machine learning models to your data by invoking dynamic Power Query functions. These machine learning models must have schema files generated in Python by the model creator.

Dataflow Gen1 creators can also use AutoML to create their own machine learning models by using Power BI during data preparation. Creators can choose a specific type of model, either binary prediction, general classification, or regression. Next, they train these models with input data, and evaluate the results before they apply the model to new or updated data after dataflow refresh.

Use cases for Azure Machine Learning integration with a semantic model or dataflow Gen1 include:

- Conduct predictive modeling in Power BI without needing a deep expertise in data science tools or Python.
- Perform simple churn prediction and forecasting.
- Apply organizational models in Azure Machine Learning to enrich data in Power BI.

Integration for Independent Software Vendors

Independent software vendors (ISVs), who produce and sell software, can integrate with Fabric to support and extend their applications.

There are three different models that ISVs can use to integrate with Fabric:

- **Interop model**: ISVs can integrate with OneLake through various tools, such as the OneLake APIs, and others.
- **Develop on Fabric model**: ISVs can develop their own products and services on Fabric, and even embed the capabilities of Fabric in their software.
- **Build a Fabric workload model**: ISVs can use the Microsoft Fabric Workload Development Kit to create and monetize workloads.

For more information about how ISVs can integrate with Fabric, see Microsoft Fabric Integration Pathways for ISVs.

Microsoft Teams integration

You can integrate your tenant with Microsoft Teams to allow users to access Fabric and Power BI from within the Teams application. This capability is a convenient way to centralize collaboration and promote adoption of both Teams and Power BI.

For more information about how to integrate teams with Power BI, see:

- Add the Power BI app to Microsoft Teams: Integrate the Power BI experience into Microsoft Teams.
- Embed interactive reports in Teams channels and chats with a Power BI tab: Help colleagues find and discuss your team's data.
- Use interactive reports in Teams meetings: Discuss a report during a meeting, or use the report to support the meeting objectives.
- Create a link preview in the Teams message box: Paste links to reports, dashboards, or Power BI apps.
- Chat in Microsoft Teams directly from within the Power BI service: Share a filtered view of reports and dashboards, and start conversations.
- View all the Power BI tabs you have in Microsoft Teams: Select the *In Teams* tab on the Power BI app home page.

- Get notified in the Teams activity feed: Quickly learn when important events happen in Power BI.

Use cases for Teams integration with Power BI include:

- Curate a centralized portal for your community of practice and embed key Power BI reports and resources.
- Create dedicated teams or teams channels for content distributed from a Power BI app, where people can share feedback, issues, or ask questions about the content.
- Train users to make shared views that they can share via Teams to discuss specific perspectives or data points.

Geospatial services integration

When you work with geospatial data, you'll probably want to visualize it in interactive map visuals with Power BI. However, these visuals require integration with other services, which you can control at the tenant-level by using the tenant settings. These visuals can be effective in reports that present geospatial data, but you should ensure that using these services doesn't violate any data residency or compliance requirements.

For more information about how to integrate Power BI with various geospatial services, see:

- ArcGIS visualizations in Power BI reports, which use Esri services.
- Azure Maps visualizations for Power BI reports, which use Azure services.
- Map and filled map visuals, which use Bing services.

> ⚠ **Warning**
>
> Geospatial services might use other services that are outside of the geographic region of your Power BI tenant, compliance boundary, or national cloud instance. Furthermore, these services might store and process your data where they maintain facilities, and use of these services might be subject to separate terms and privacy policies beyond Power BI.
>
> This warning also applies to any third-party custom visual that you use to visualize geospatial information.

Workspace-level integration

You can integrate certain services at the level of individual workspaces. These services can enable capabilities to help you develop, manage, and view content in a workspace.

Git integration

If your workspace uses Fabric capacity, Premium capacity, or PPU license modes, you can use Git integration to connect a workspace to a remote Git repository to support more advanced lifecycle management scenarios. A remote Git repository facilitates source control of files, which allows content creators to track and manage changes. Git integration also promotes collaboration among developers, particularly when using branches to isolate development of specific features before integrating those changes into a main branch with a merge before deployment.

In brief, content creators can develop content either locally or in the Power BI service, then commit and push those changes to a remote Git repository, like Azure Repos or GitHub Enterprise. For information about how to set up and use Git integration for Power BI and Fabric, see Get started with Git integration or Tutorial: end-to-end lifecycle management.

Content creators store Power BI Project (.pbip) files, metadata files, and documentation in a central Azure Repos remote repository. These files are curated by a technical owner. While a content creator develops a solution, a technical owner is responsible for managing the solution and reviewing the changes and merging them into a single solution. Azure Repos provides more sophisticated options for tracking and managing changes compared to SharePoint and OneDrive. Maintaining a well-curated, documented repository is essential because it's the foundation of all content and collaboration.

Consider using source control to track and manage changes in the following scenarios:

- Centralized or decentralized teams create and manage the content.
- Content creators collaborate by using Azure DevOps.
- Content creators are familiar with Git, source control management, or DataOps architecture design.
- Content creators manage complex or important content, or they expect the content to scale and grow in complexity and importance.

To help you effectively use source control with Azure DevOps, you need to be aware of considerations and meet certain prerequisites:

- **Git**: To commit and push changes to a remote repository, content creators need to download and install *Git*. Git is a distributed version control system that tracks changes in your files. To learn about the basics of Git, see What is Git?.

- **Tools**: To use Git, content creators need to use either a command line interface (CLI) or a graphical user interface (GUI) client that has integrated source control management (SCM), like Visual Studio or Visual Studio Code.
- **Licenses and permissions**: To create and use an Azure Repos Git repository, content creators must:
  - Have their Azure DevOps access level set to *Basic* (as opposed to *Stakeholder*).
  - Belong to an Azure DevOps organization and a project.
  - Have appropriate Azure DevOps repository permissions.
  - Work with Power BI items only due to the Git integration constraints when using a Power BI Premium capacity (A SKUs) or PPU workspaces.
- **Fabric Git integration**: To sync content in a remote repository with a Fabric workspace, content creators use Fabric Git integration. This tool is important because it tracks and manages changes to content that's created in the Fabric portal, like dataflows.

Integrate Azure Log Analytics

You can use Azure Log Analytics to gather valuable information to support data-level auditing of workspace items. Azure Log Analytics is a component of the Azure Monitor service. Specifically, Azure Log Analytics integration with Power BI allows you to capture semantic model events from all semantic models in a Power BI workspace. It's only supported for workspaces that use Fabric or Premium capacity. For information about how to set up and use Azure Log Analytics for Power BI and Fabric, see Data-level auditing: Azure Log Analytics and Configure Azure Log Analytics in Power BI.

After you set up Azure Log Analytics integration and the connection is enabled (for a supported workspace), semantic model events are automatically captured and continually sent to an Azure Log Analytics workspace. The semantic model logs are stored in Azure Data Explorer, which is an append-only database that's optimized for capturing high-volume, near real-time telemetry data.

Use cases for using Azure Log Analytics include:

- You want to monitor strategically important semantic models, like centralized models that you provide to decentralized teams in a managed self-service usage scenario.
- You want to audit or investigate semantic models that have a high impact on resource utilization, like Fabric capacity.
- You want detailed analytics about query and usage patterns for semantic models.

To use Azure Log Analytics, you must set up and pay for an Azure Log Analytics workspace as part of your Azure subscription. You pay for Azure Log Analytics with a

pay-as-you-go subscription. For more information, see Azure Log Analytics Pricing.

Integrate Azure Data Lake Storage Gen2

You can connect a workspace to an Azure Data Lake Storage (ADLS) Gen2 account. When you connect a workspace to ADLS Gen2, you can store data for Power BI dataflows (also called dataflows Gen1) and semantic model backups. For information about how to set up and use ADLS Gen2 to store data from Power BI dataflows, see Configuring dataflow storage to use Azure Data Lake Gen 2.

Setting Azure connections in the Fabric admin portal doesn't mean that all Power BI dataflows for the tenant are stored by default in an ADLS Gen2 account. To use a specific storage account (instead of internal storage), each workspace must be explicitly connected. It's critical that you set the workspace Azure connections *before you create any Power BI dataflows* in the workspace.

The following two sections present reasons why you might integrate a workspace with ADLS Gen2.

Storage of Power BI dataflows data

If you bring your own data lake, the data for Power BI dataflows (Gen1) can be accessed directly in Azure. Direct access to dataflow storage in ADLS Gen2 is helpful when you want other users or processes to view or access the data. It's especially helpful when your goal is to reuse dataflows data beyond Power BI.

There are two options for assigning storage:

- Tenant-level storage: This option is helpful when you want to centralize all data for Power BI dataflows into one ADLS Gen2 account.
- Workspace-level storage: This option is helpful when business units manage their own data lake or have certain data residency requirements.

> **Tip**
>
> If you use Fabric, we recommend that you use **dataflows Gen2**, which can store data in **different destinations**, including **OneLake**. Dataflows Gen2 are more flexible than dataflows Gen1, because they provide more options to **integrate with other data pipelines** and they benefit from **high-scale compute**.

Backup and restore for Power BI semantic models

The Power BI semantic model backup and restore feature is supported for workspaces that are assigned to Fabric capacity, Premium capacity, or PPU. This feature uses the same ADLS Gen2 account that's used to store Power BI dataflows data (described in the previous section).

Semantic model backups help you:

- Comply with data retention requirements.
- Store routine backups as part of a disaster recovery strategy.
- Store backups in a different region.
- Migrate a data model.

Solution-level integration

You can integrate certain services at the level of individual items, like semantic models or reports. These integrations can enable specific use cases and extend the functionality of your Power BI items.

Integration with Microsoft Fabric

Power BI is part of Fabric, but Power BI is a distinct workload in Fabric that can integrate with the other experiences that are unified under the Fabric umbrella. If you're familiar with only working with Power BI, it's important to understand the possibilities and opportunities to apply other workloads, items, and features in Fabric.

The following sections present examples of how you can integrate Power BI content with Fabric to extend the capabilities of Power BI.

OneLake integration with semantic models

Content creators who make Power BI semantic models can use OneLake integration to write model tables to Delta tables in OneLake. After the initial copying of the in-memory tables, they can then be reused from OneLake for other use cases, without the need to copy them. The Delta tables are accessible via a lakehouse in Fabric. Users can also create shortcuts to access the tables so they can use them from another lakehouse or a different item type, like a data warehouse.

Use cases for using OneLake integration with semantic models include:

- Reuse data from a semantic model that isn't already available in OneLake.
- Reuse data from a semantic model for use in another Fabric experience.
- Create snapshots of a semantic model table.

Semantic link integration with semantic models via notebooks

Content creators who build semantic models or analyze data in notebooks can use semantic link to read and write to semantic models from a notebook in Fabric. Semantic link has a wide range of benefits for Power BI developers, including enhanced productivity, automation, and the ability to quickly and easily perform ad hoc analysis in code.

Use cases for using semantic link integration with semantic models include:

- Automate testing of semantic models by evaluating DAX queries and comparing the results to known baselines.
- Programmatically manage semantic models by running Best Practice Analyzer over multiple models at the same time to identify and classify possible issues.
- Save common templates and patterns for DAX measures and business logic (like currency conversion) that can be applied to new semantic models.
- Analyze and visualize data from a semantic model by using Python.
- Validate models created by data scientists by using the business logic from a semantic model.
- Use data from a semantic model to enrich analysis.

> **Tip**
>
> The **semantic-link-labs** Python library further extends the utility of semantic link. It's a valuable tool for anyone who creates and manages semantic models and wants to improve productivity and efficiency of the model creation or management process.

Even if you don't know Python, you can use Copilot and Chat-magics to gain assistance writing functional Python code to obtain a useful result.

Fabric Activator integration with Power BI reports

Content creators or consumers who build or use Power BI reports can use Activator to automate actions and notifications based on data changes. Similar to data alerts from dashboard tiles, a user can set alerts on a Power BI visual and define triggers for those alerts. The user can also extend this functionality to use Custom actions to trigger a Power Automate flow that can initiate other downstream changes.

Use cases for Activator integration with Power BI include:

- Automated anomaly detection, by setting an alert to trigger when a value exceeds a threshold.
- Automated regression testing of business-critical reports, by setting an alert to trigger when a value (like previous year sales, or a budget variance) exceeds a threshold.

Integration with Microsoft Office 365

There are many ways to integrate Power BI with Microsoft 365 products, like Excel, PowerPoint, and Outlook.

Use Power BI data in Excel

Users who prefer working in Excel can use either Analyze in Excel or live connected tables to use Power BI data.

Content consumers who have Build permission for a semantic model can connect to the model from Excel to use Analyze in Excel. This approach allows users to explore models so they can perform their own ad hoc analysis with PivotTables.

Use cases for Analyze in Excel include:

- Users prefer to analyze data in Excel instead of using Power BI.
- Users want to conduct personal BI to create their own reports in Excel.
- Users want to use Power BI data to support existing analysis in Excel.

> **Tip**
>
> If you expect users to connect to a semantic model from Excel, ensure that you take the necessary steps to train them how to use it, and organize your semantic model in a helpful way. For example, organize fields into display folders, and hide tables and fields that aren't intended for use in reports.

Analyze in Excel uses Multidimensional Expressions (MDX) for queries instead of Data Analysis Expressions (DAX) used by Power BI reports. MDX queries can experience poorer performance compared to equivalent DAX queries. Ensure that users understand that they should use Analyze in Excel for high-level aggregate analysis, and perform more detailed analysis by using Power BI or other Fabric experiences.

Also, some features in a semantic model, like field parameters and dynamic measure format strings, don't work in Analyze in Excel. For other considerations and limitations, see this article.

You can also get Power BI data in Excel by using live-connected tables. With this approach, users who export data from a Power BI report visual receive an Excel workbook that contains a table populated with data. The table query automatically retrieves the latest data when the workbook is opened, or when the table is manually refreshed.

Use cases for live-connected tables include:

- Users want to investigate or analyze the data in a particular visual.
- Users need to regularly export data to support a valid business case.
- You're performing manual testing of a semantic model or report.

While exporting live-connected tables is better than exporting disconnected tables from a Power BI report, you should encourage users to avoid exporting any data. Exported data presents governance challenges and data security risks that can lead to data exfiltration from the organization. Instead, consider training users to connect to semantic models from Excel or Power BI Desktop to perform their own analysis, and to safely share the results with their colleagues.

Managing data exports is an important change management exercise to improve the maturity of your data culture and enable people to use Power BI effectively.

Integrate Power BI reports in PowerPoint

You can use the Power BI add-in for PowerPoint to add live, interactive Power BI reports or specific visuals to PowerPoint slides. This feature is a good alternative to inserting static screenshots because the visuals can be filtered and interacted with during a presentation.

PowerPoint is a useful tool to complement existing Power BI reports, but it doesn't scale as a primary distribution method. Instead, use report distribution methods, like Power BI apps, and look for opportunities for PowerPoint integration to complement or extend them.

Managing distribution of reports as flat files and PowerPoint presentations is an important change management exercise to improve the maturity of your data culture and content delivery scope, and to enable people to use Power BI effectively.

Use cases of Power BI integration in PowerPoint include:

- Continuously play a presentation in slide show mode with up-to-date Power BI reports, for example on a large screen in a factory.

- Freeze snapshots of a specific view so that report data doesn't update automatically, for example when you want to review point-in-time reports from a past date.
- Share a presentation with live Power BI reports so people can see the latest data, for example when you want an audience to review the presentation and reports before you present it.

Fabric administrators can control its use with the *Enable Power BI add-in for PowerPoint* tenant setting. For other considerations and limitations, see this article.

Integration with Power Platform

Power BI is part of Power Platform. As such, Power BI integrates well with other applications in the Power Platform family, such as Power Apps, Power Automate, and Power Pages.

- Power Apps lets you quickly create and deploy low-code applications in your organization.
- Power Automate lets you automate tasks and workflows by creating logical flows that trigger either automatically, on a schedule, or in response to a manual action. You can create cloud flows that run unattended without a dedicated machine. You can also use the Power Automate desktop application to author desktop flows that require a machine because they use robotic process automation to simulate user actions.
- Power Pages lets you create external-facing business websites with a low-code user interface.

Use the Power Apps visual in a Power BI report

You can integrate Power Apps in Power BI by using the Power Apps visual. This visual allows you to display an interactive, functional Power Apps canvas app within a Power BI report. In Power BI, you can select fields to add to the Power Apps visual. Then, in Power Apps, you can use these fields to create data-driven labels and functionality to enhance your app. Together, the integration of Power BI reports and Power Apps enables a wide range of use cases that can help people make decisions and take actions by using data in a report.

There are some licensing considerations to keep in mind if you take this approach. To use the Power Apps visual in the Power BI report, a report viewer must have a Power Apps license in addition to any required Power BI per-user licenses. Alternatively, you can use a pay-as-you-go plan for Power Apps and Power Automate.

Use cases of the Power Apps visual include:

- Facilitate writeback to a database, for example to add comments to certain customers or to modify forecast values from within a Power BI report.
- Facilitate direct actions informed by the Power BI report, such as contacting customers from a customer satisfaction report.
- Allow users to submit forms from within the Power BI report, such as feedback forms, polls, or surveys.

In an embedded scenario, the Power Apps visual is only supported for the Embed for your organization scenario and not the Embed for your customers scenario. For other limitations, see Limitations of the Power Apps visual.

Integrate a Power BI report in a Power Apps canvas app

You can integrate Power BI dashboard tiles within a Power Apps canvas app. With this approach, the primary consumption medium is the Power App, which is enhanced by the Power BI tile. You embed tiles by using the Power BI tile control during canvas app development.

Take actions in Power BI from Power Automate

You can use Power Automate to automate specific actions in Power BI, such as report export, semantic models refresh, or DAX query evaluation. This capability can be helpful to streamline certain tasks or improve productivity.

Use cases for automating Power BI from Power Automate include:

- Trigger refresh of a semantic model when an upstream data source is updated.
- Automate distribution of Power BI reports or paginated reports.
- Add rows to a Power BI semantic model table when a flow is triggered.

Trigger a Power Automate flow from Power BI

You can also use Power BI to trigger a Power Automate cloud flow in three ways:

- Use the Power Automate visual in a Power BI report.
- Use data alerts from a Power BI dashboard tile.
- Create flows to notify changed goals in Power BI.

With this approach, you aren't automating Power BI actions as much as you're responding to events that happen in Power BI. These events can either be triggered manually (like the Power Automate visual) or automatically (like data alerts). You can also

use data from Power BI downstream in the flows, which can help you automate more specific and relevant actions.

There are some licensing considerations to keep in mind with this approach. To use the Power Automate visual in a Power BI report, the report viewer must have access to the Power Automate flow and a Power Automate license, if necessary, in addition to any required Power BI per-user licenses. Alternatively, you can use a pay-as-you-go plan for Power Apps and Power Automate.

Use cases for triggering a Power Automate flow from Power BI include:

- Update or add rows to an Excel table from within a Power BI report by using the Power Automate visual.
- Automate regression testing by setting up reports and dashboards to report differences in current values with known baselines, and by setting data alerts on dashboard tiles.
- Notify a team or individual when there are unexpected values or anomalies in semantic model data by using data alerts. To use the Power Automate visual in a Power BI report, the report viewer must have access to the Power Automate flow and a Power Automate license, if necessary, in addition to any required Power BI per-user licenses. Alternatively, you can use a pay-as-you-go plan for Power Apps and Power Automate.

Embed a Power BI report in a Power Pages website

You can embed a Power BI report in a Power Pages website, which allows you to show Power BI reports on your external-facing website made with Power Pages. This approach streamlines the Embed for your customers scenario by enabling the Power BI Embedded service from the Power Platform admin center.

There are some licensing considerations to keep in mind with this approach. To embed Power BI reports in a Power Pages website, you must have an F, P, EM, or A SKU. You also need an appropriate Power Pages license.

Use cases for embedding a Power BI report in a Power Pages website include:

- Distribute reports via a custom portal to external users or customers.
- Display website analytics, like subscribers or traffic for your website.
- Enhance your Power Pages website with interactive Power BI visualizations.

In addition to the limitations of Power BI Embedded, there are also specific limitations for embedding a report in Power Pages. For example, the report must be published to the same workspace as its connected semantic model. Ensure that you take these

considerations into account before you decide to embed Power BI content in a Power Pages website.

OneDrive and SharePoint integration

OneDrive and SharePoint are commonly used because they're convenient options to store content and data files for Power BI. By integrating OneDrive and SharePoint, you can further enhance their sharing capabilities.

OneDrive refresh of Power BI Desktop files

When you save a Power BI Desktop (.pbix) file to OneDrive for Work or School, or SharePoint, you can import that file into a workspace from OneDrive instead of publishing it from Power BI Desktop. By doing so, you can benefit from OneDrive refresh, where the data model is automatically updated, usually within an hour.

Use cases for using OneDrive refresh include:

- Self-service users want to streamline publishing of Power BI Desktop files.
- Content creators want to track and manage changes while collaborating in OneDrive.

In addition to integrating OneDrive for an individual .pbix file for semantic models and reports, you can also set up workspace-level integration with OneDrive.

Preview Power BI Desktop files in OneDrive and SharePoint

When you share a Power BI Desktop file with people via OneDrive or SharePoint, they can preview the report from OneDrive or SharePoint without opening it in Power BI Desktop. This capability works only for reports that are connected to a shared semantic model, or Power BI Desktop files that contain a report and an import semantic model. Additionally, you can't preview Power BI Desktop files that are 1 GB or larger. For more information, see Considerations and limitations.

There are some licensing considerations to keep in mind with this approach. Users require a Power BI Pro license to preview Power BI Desktop files in OneDrive or SharePoint. For more information, see Prerequisites to viewing reports in OneDrive and SharePoint.

Use cases for using OneDrive to preview files include:

- Content creators use OneDrive or SharePoint to facilitate collaboration.

- Content creators who use OneDrive integrations, like OneDrive refresh, or use it to track and manage changes to .pbix files, want the convenience to preview the files before they open them.

Embed Power BI reports in SharePoint Online

You can integrate Power BI with SharePoint by embedding Power BI reports in SharePoint Online (also known as *secure embed*). The report experience is the same as when users view them in a Fabric workspace by using a link shared with direct access. Row-level security is enforced, together with item permissions. Users must have direct access to reports in order to view them in a SharePoint site.

Use cases for embedding Power BI reports in SharePoint Online include:

- You want to distribute reports from a SharePoint portal instead of via a Fabric workspace. This approach can be useful when you want to distribute reports from several workspaces to a specific audience.
- You want to embed reports that support collaboration or decision making in your SharePoint site.

Integration with Visual Studio and VS Code

Many developers are familiar with using Visual Studio or Visual Studio Code (VS Code) to manage source files and metadata. These tools provide several options to integrate with Power BI and Fabric.

Develop semantic models by using Visual Studio with Analysis Services projects

If developers prefer working in Visual Studio, they can develop and deploy semantic models from Visual Studio instead of Power BI Desktop. In this case, they need Visual Studio 2017 or a later edition, and the 2.9.14 version (or higher) of the SQL Server Data Tools (SSDT) extension.

> **Tip**
>
> Developers who prefer a Visual Studio-like experience to build and manage semantic models may find it more effective to use **Tabular Editor**. Tabular Editor is an external tool that connects to a local model open in Power BI Desktop, or a remote model via the XMLA read/write endpoint. It also supports scripting and batch tasking to improve developer productivity.

For more information, see **Advanced data model management**.

Manage items with VS Code

If developers prefer working in VS Code, they can use extensions to facilitate some of their work with Power BI via the VS Code application.

There are several tools that they can use to manage different parts of Power BI from VS Code:

- **TMDL**: An official VS Code extension from Microsoft that provides language support for the Tabular Model Definition Language (TMDL) to work with semantic models that use the TMDL metadata format.
- **Power BI Studio**: A community-developed VS Code extension that uses the Power BI REST API to view and manage items in a workspace.
- **Power BI VSCode Extension Pack**: A collection of VS Code extensions that enables developers to work in VS Code with Fabric and Power BI. It includes both the TMDL and Power BI Studio extensions.

VS Code integration is also supported by other Fabric experiences, like notebooks for data engineering and data science, or for managing Power BI semantic models by using semantic link (described earlier).

Python or R integration

You can run Python or R scripts in Power BI semantic models and reports to extend the functionality of these items. This capability can be helpful for content creators who are familiar with Python or R, and who create and distribute content for business users via Power BI.

Content owners or creators who are proficient in Python or R might benefit from using notebook items in a Fabric capacity. For many use cases, notebooks are a preferred option over Python and R integration with Power BI. That's because they provide more options to create and maintain solutions built in these languages, they also have fewer limitations and typically involve less effort to support.

Run Python or R code in a semantic model

You can integrate Python or R code as part of the data transformations that you perform in a semantic model that uses import storage mode. This integration lets you transform data or perform advanced analytics with Python or R whenever you refresh the model.

To refresh a published semantic model that uses Python or R integrated in Power Query, you must use an on-premises data gateway in personal mode. That's because the Python or R code runs locally by using the Python or R installed on the machine. This setup is typically challenging to manage and maintain. If you need to use Python or R in a semantic model, we recommend alternative approaches, like notebooks in Fabric.

Create Python or R visuals in Power BI reports

You can integrate Python or R with Power BI reports in order to create custom visuals with Python libraries, like Seaborn or R packages like ggplot2. These visuals are fully customizable and support interactive features in Power BI like rendering a filtered result, cross-filtering, custom tooltips, drilldown, and drillthrough.

Ensure that all your Python or R visuals use Python libraries and R packages that are supported in Fabric. If you use an unsupported library or package, the visual won't render in the Power BI service, even when the visual renders in your report in Power BI Desktop.

While you can transform data and make calculations as part of a Python or R custom visual, it isn't recommended. Placing this logic in the Python or R visual can result in slower render durations, and greater difficulty to maintain the visual and achieve harmonization across visuals and reports in calculation logic.

Instead, add your logic to DAX calculations by creating measures, and perform your transformations further upstream, such as in Power Query or the data source, if possible.

Custom visuals for Power BI reports

There are other options for creating custom visuals in Power BI reports aside from Python and R. While not explicitly integration, it's possible to use custom visuals in Power BI reports for advanced or specific use cases. You can create your own custom visual—which requires no integration with other services—or obtain a visual from AppSource, which can be either free or requires a license. Depending on the custom visual, it might involve integration with a third-party service and you will need to agree to their license terms.

If you're thinking about using custom visuals to extend the functionality of Power BI reports, consider Deneb. Deneb is a community-developed, certified custom visual that allows you to use a declarative JSON syntax of the Vega or Vega-Lite languages to build your own visualizations. Deneb has a large community and many templates, which makes it a good choice for report creators who want to create their own visuals without using JavaScript, Python, or R.

Integration with other third-party services

There are other third-party services which offer integration with Power BI.

The following section presents third-party services, together with use cases that are important to consider.

Integration with semantic models via the XMLA endpoint

In Power BI, external tools can connect to your Power BI semantic models by using the XMLA endpoint. There are both open source and commercially available tools that you can use to enhance productivity or extend the functionality of your existing semantic models.

Here are some examples of tools that can integrate with semantic models via the XMLA endpoint:

- PowerShell cmdlets to automate certain semantic model tasks.
- Power BI report builder to query semantic models with DAX and build paginated reports.
- Tabular Editor, a third-party tool to develop and manage semantic models.
- DAX Studio, a third-party tool to author and optimize DAX queries.
- ALM toolkit, a third-party tool to compare and deploy semantic models.

For more information about XMLA endpoints and client applications and tools that use them, see Semantic model connectivity and management with the XMLA endpoint in Power BI. The XMLA endpoint is only supported for workspaces that have their license mode set to Fabric capacity, Premium capacity, or Premium Per User.

You can enable the XMLA endpoint and set it to read, or read/write from the Power BI workload options in the admin portal. There are also several tenant settings that you can use to control which users and groups can use the XMLA endpoint.

Checklist - When planning to integrate Power BI with other services, key decisions and considerations include:

- **Define the requirement**: Describe what you're trying to achieve and what the expected benefit is of doing so.

- **Describe why you can't accomplish the task in Power BI alone**: Define the challenges or limitations that prevent you from fulfilling this requirement with the built-in tools and features in Power BI.
- **Identify the services that can help you fulfill the requirement**: Compile a list of the services that can help you achieve your objective. Depending on the requirement, there might be only one plausible option.
- **Identify any potential risks, limitations, or considerations**: Carefully plan and consider the implications for this integration for different areas, like security, licensing, governance, and user enablement.
- **Research how you'll set up the integration**: Read the appropriate technical documentation and compile a step-by-step protocol that applies to your specific scenario for how you'll integrate Power BI with the service or tool. Pay special attention to possible troubleshooting or customization of this integration that you might need to do.
- **Conduct a test or proof of concept (POC)**: Before you set up the integration for your tenant, workspace, or item, first perform a representative trial to test any assumptions and reveal any challenges or limitations. Conducting a test or POC is important.
- **Set up training and monitoring**: Ensure that centralized teams are equipped to monitor the new service and its effect on usage in your tenant. Prepare relevant training material so that people can use the new service, and that helps them to avoid issues.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Auditing and monitoring

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article introduces the Power BI auditing and monitoring articles. These articles are targeted at multiple audiences:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators might need to collaborate with information security and other relevant teams.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators, information security teams, and other relevant teams.
- **Content creators and workspace administrators**: Users who need to understand usage and adoption for the content that they've created, published, and shared with others in the organization.

The terms *auditing* and *monitoring* are closely related.

- **Auditing**: Actions taken to understand a system, its user activities, and related processes. Auditing activities can be manual, automated, or a combination. An auditing process might focus on one specific aspect (for example, auditing security for a workspace). Or it might reference an end-to-end auditing solution, which involves extracting, storing, and transforming data so that you can analyze and act upon it.
- **Monitoring**: Ongoing activities that inform you about what's occurring. Monitoring usually involves alerting and automation, though sometimes monitoring is done manually. Monitoring can be set up for a process you've selected to audit (for example, notifications when a specific tenant setting changes).

The primary purpose of this series of auditing and monitoring articles is to help you understand *how* Power BI is used to oversee and govern your Power BI implementation.

Troubleshooting and performance tuning are important components of auditing and monitoring your data assets. However, providing deep performance tuning guidance isn't a goal of these articles. Also, these articles aren't intended to provide a complete reference of all options available to developers.

> **ⓘ Important**
>
> We recommend that you closely follow the Microsoft Fabric release plan to learn about future enhancements of the auditing and monitoring capabilities.

Value of auditing and monitoring

The data that's produced from auditing is incredibly valuable for many reasons. Most people think of auditing as an oversight and control function. While that's true, you can also audit data to improve the user experience.

This article describes some valuable ways you can use auditing data.

Analyze adoption efforts

As described in the Fabric adoption roadmap, adoption isn't just about *using* the technology regularly; it's also about using it *effectively*. Adopting a technology like Power BI can be considered from three inter-related perspectives:

- **Organizational adoption**: The effectiveness of Power BI governance. It also refers to data management practices that support and enable business intelligence (BI) efforts.
- **User adoption**: The extent to which Power BI consumers and creators continually increase their knowledge. It's concerned with whether they're actively using Power BI, and whether they're using it in an effective way.
- **Solution adoption**: The impact and business value achieved for individual requirements and deployed Power BI items (like semantic models and reports).

All types of auditing data can be used in many ways to assess and contribute towards actions that improve each aspect of adoption.

Understand usage patterns

Analyzing usage patterns is primarily about understanding the user activities that occur in your Power BI tenant.

It's helpful to have the data to support whether actual user behavior meets expectations. For example, a manager might be under the impression that a set of reports is critical, whereas the auditing data shows that the reports aren't regularly accessed.

> **Important**
>
> If you're not already extracting and storing user activities data, make that an urgent priority. Even if you're not ready to build an end-to-end auditing solution, ensure that you're extracting and storing all the activity log data. For more information about the decisions and activities involved, and the different ways to obtain the data, see **Access user activity data**.

The following sections describe some of the most common usage patterns you should understand.

Content usage

It's valuable to understand the extent to which content is used. The types of questions you might ask include:

- What content is viewed most frequently?
- What content is viewed by the greatest number of users?
- What content is considered most critical (and is therefore vital to the organization) based on its usage patterns?
- What content is frequently used by executives and senior leadership?
- What content requires the most stability and support (due to a high level of usage or usage by a critical user audience)?
- What content should be endorsed (certified or promoted) based on its usage patterns?
- What percentage of content is endorsed (certified or promoted)?
- Are there high numbers of report views for non-certified reports?
- What content has consistent usage versus sporadic usage?
- What content is updated most frequently, when, and by which users?
- What content *isn't* used, with the potential to be retired? (For more information about creating an inventory of data assets, see Access tenant inventory data.)
- What type of devices are used to view reports?
- Are there unexpected, or irregular, usage patterns that raise concerns?

User activities

It's useful to understand which users are most active. The types of questions you might ask include:

- Which content consumers are most active?
- Which content creators are most active?
- Which content creators publish the most content?
- Which content creators publish the content that's used by the most content consumers?
- How many distinct (licensed) users are there? What percentage of those users are active?
- Are there content creators who are assigned a Power BI Pro or Power BI Premium Per User (PPU) license, but aren't actively using that license?
- Are the most active users members of your Power BI champions network?

> **Tip**
>
> For analytical reporting, it's important that you add classifications to the data model to analyze users based on their level of usage, or to analyze content based on its level of usage. For more information, see **Create classifications**

For more information about the Power BI activity log, see Access user activity data. For more information about pre-built reports, see What is the admin monitoring workspace?.

Understand published items

It's helpful to have an inventory of the content that exists in your Power BI tenant. Whereas the previous section is concerned with *user activities*, this section is concerned with a *tenant inventory*.

A tenant inventory is a snapshot of metadata at a given point in time. It describes what's been published to the Power BI service, and it includes an inventory of all workspaces, all reports, all semantic models, and more. It can also include metadata for data sources or supporting resources, like gateways and capacities.

A tenant inventory is helpful to:

- **Understand how much content you have and where**: Because the tenant inventory includes all published items, it represents a complete inventory at that time. You can use it to identify where content is published, and its dependencies and lineage.

- **Examine the ratio of semantic models to reports**: You can use lineage information from the tenant inventory to determine the extent to which data reuse occurs. For example, if you discover many duplicate semantic models, it could justify user training on shared semantic models. You might also decide that a consolidation project to reduce the number of semantic models is justified.
- **Understand trends between points in time**: You can compare multiple snapshots to identify trends. For example, you might find that a large number of new items are published every month. Or you might you discover that users are republishing a new report (with a different report name) every time they modify it. Those types of discoveries should prompt you to improve user training.
- **Scrutinize user permissions**: You can gain valuable insight into how user permissions are assigned. For example, you might routinely analyze which users have access to what content. You might undertake further investigation to determine whether asset oversharing is occurring. One approach you can take is to correlate certain sensitivity labels (such as *Highly Restricted*) with high numbers of user permission assignments.
- **Understand lineage and find highly used data sources and gateways**: By correlating the lineage information from the tenant inventory with user activities, you can identify the most frequently used data sources and gateways.
- **Find unused content**: You can compare your tenant inventory to the activity log to find unused (or under-utilized) content. For example, there are 20 reports in a workspace, yet only 12 reports have recent data in the activity log. You can investigate why the other eight reports aren't used, and whether they should be retired. Discovery of unused content might help you detect Power BI solutions that need further development because they aren't useful.

> **Tip**
>
> We recommend that you store your tenant inventory snapshot every week or every month. Also, by combining activity log data with your tenant inventory snapshot, you can produce a complete picture and enhance the value of the data.

For more information about the tenant inventory, see Access tenant inventory data.

Educate and support users

Auditing data allows you to gain a deep understanding of how users in your organization are using Power BI. In turn, your ability to educate and support users can improve dramatically.

Based on actual user data, here are some examples of the types of actions you might take.

- **Provide helpful information to users**: When you see an activity for the first time (for example, the first time a user publishes a report), you can send them an email with information about your internal best practices for workspaces and security.
- **Teach users a better way**: When you see certain activities that concern you (for example, a significant and recurring number of report exports), you can contact the user. You can then explain the downsides of their actions and provide them with a better alternative.
- **Include in office hours**: Based on recent activities, you can choose to discuss a relevant topic during office hours.
- **Improve training curriculum**: To better prepare new users (and avoid the same missteps you see happening with existing users), you can improve or expand your training content. You might also conduct some cross-training sessions for your support team.
- **Improve the centralized portal**: To improve consistency, you can invest time by adding or changing the guidance and resources available in your centralized portal.

Mitigate risk

Auditing data helps you understand what's happening in your Power BI tenant. This data allows you to mitigate risk in various ways.

Auditing data helps you to:

- **Govern the Power BI tenant**: Find out whether users are following your governance guidelines and policies. For example, you might have a governance policy that requires all content that's published to a particular workspace be certified. Or, you might have guidelines for when groups (rather than users) should be used for security. The activity log, your tenant inventory (described previously), and admin APIs are helpful resources to help you govern your Power BI tenant.
- **Review security**: Determine whether there are security concerns. For example, you might see overuse of sharing from a personal workspace. Or you might see unrelated content published to a single workspace (which leads to more complicated security for the items in such a broadly defined workspace). The activity log, your tenant inventory (described previously), and the admin APIs are helpful for security auditing.
- **Minimize security issues**: Use the activity log data to avoid or minimize the effect of security issues. For example, you might detect that an organization-wide sharing link was used in an unexpected way. By noticing this event in the activity log

shortly after it happens, you can take action to resolve the issue before the link is used inappropriately.

- **Monitor tenant setting changes**: Use the activity log data to determine when a tenant setting has changed. If you see that an unexpected change occurred, or that it was done by an unexpected user, you can act quickly to correct or revert the setting.
- **Review data sources**: Determine whether unknown or unexpected data sources are used by semantic models, dataflows, or datamarts. You might also determine what types of data source are in use (such as files or databases). You might also check whether files are stored in an appropriate location (such as OneDrive for work or school).
- **Information protection**: Review how sensitivity labels are used to reduce the risk of data leakage and misuse of data. For more information, see the information protection and data loss prevention series of articles.
- **Mentoring and user enablement**: Take action to change user behaviors when necessary. As you gain more knowledge about what users need and what actions they are taking, you can influence mentoring and user enablement activities.
- **Monitor tenant setting changes**: Use the activity log data to determine when a tenant setting has changed. If you see that an unexpected change has occurred, or that it was done by an unexpected user, you can act quickly to correct or revert the setting. You can also use the Get Tenant Settings REST API to regularly extract a snapshot of the tenant settings.

Improve compliance

Auditing data is critical to strengthen your compliance status and respond to formal requests in different scenarios.

The following table provides several examples.

Expand table

Type of requirement	Example
Regulatory requirements: You need data to strengthen your compliance status for industry, governmental, or regional regulatory requirements.	You have a compliance requirement to track where personal data exists. Data Loss Prevention (DLP) for Power BI is one option for detecting certain types of sensitive data that's stored in published semantic models. The metadata scanning APIs are another option for locating personal data. For example, you could check for certain

Type of requirement	Example
	column names or measure names that exist within published semantic models.
Organizational requirements: You have internal governance, security, or data management requirements.	You have an internal requirement to use row-level security (RLS) on all certified semantic models. The Get Datasets As Admin API can help you verify whether this requirement is being met. Or, you have an internal requirement that sensitivity labels are mandatory for all content in Power BI. You can use the metadata scanning APIs to verify which sensitivity labels have been applied to Power BI content.
Contractual requirements: You have contractual requirements with partners, vendors, or customers.	You have a customer that provides your organization with data. According to your agreement with the customer, the data must be stored in a specific geographic region. You can use the Get Capacities API to verify which region a capacity is assigned to. You can use the metadata scanning APIs or the Get Groups As Admin API to verify which capacity a workspace is assigned to.
Internal audit requests: You need to fulfill requests made by internal auditors.	Your organization conducts an internal security audit every quarter. You can use several APIs to audit requests for details about permissions in Power BI. Examples include the metadata scanning APIs, the Get User Artifact Access As Admin API for report sharing, the Get Group Users As Admin API for workspace roles, and the Get App Users As Admin API for Power BI app permissions.
External audit requests: You need to respond to requests made by external auditors.	You receive a request from the auditors to summarize how all your Power BI data assets are classified. The metadata scanning APIs are one way to compile the sensitivity labels that are applied to Power BI content.

Manage licenses and costs

Because auditing data contains information about actual user activities, it can help you manage costs in various ways.

You can use the auditing data to:

- **Understand the mix of user licenses**: To reduce cost, consider reassigning unused user licenses (Pro or PPU) to other users. You might also be able to reassign a user to a Free user license. When there are many consumers who only view content, it can be more cost effective to use a Power BI Premium (P SKU) capacity or Fabric

F64 or greater capacity with Free user licenses (known as unlimited content distribution).

- **Assess the use of capacity licenses**: Assess whether a Power BI capacity (purchased with a P SKU, EM SKU, or A SKU) is sized appropriately for your workload and usage patterns. To balance cost with decentralized management needs, you might consider using multiple decentralized capacities (for example, three P1 capacities that are each managed by a different team). To reduce cost, you might provision one larger capacity (for example, a P3 capacity that's managed by a centralized team).
- **Monitor the use of capacity autoscale**: Determine whether Autoscale (available with Power BI Premium) is set up in a cost-effective way. If autoscale is invoked too frequently, it can be more cost-effective to scale up to a higher P SKU (for example, from a P1 capacity to a P2 capacity). Or, you could scale out to more capacities (for example, provision another P1 capacity).
- **Perform chargebacks**: Perform intercompany chargebacks of Power BI costs based on which users are using the service. In this situation, you should determine which activities in the activity log are important, and correlate those activities to business units or departments.
- **View trials**: The activity log records when users sign up for a PPU trial. That information can prepare you to purchase a full license for those users before their trial period ends.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Performance monitoring

Certain types of auditing data include information that you can use as an input to performance tuning activities.

You can use auditing data to monitor:

- The levels of semantic model usage, when, and by which users.
- The queries submitted by users who open live connection reports.

- Which composite models depend on a shared semantic model.
- Details of data refresh operations.
- When a data gateway is used for queries or data refresh operations.
- Which data sources are used, how frequently, and by which users.
- When query caching is enabled for semantic models.
- When query folding isn't occurring.
- The number of active DirectQuery connections for data sources.
- The data storage mode(s) used by semantic models.

For more information, see Data-level auditing.

Related implementation planning content

The remainder of the auditing and monitoring content is organized into the following articles.

- **Report-level auditing**: Techniques that report creators can use to understand which users are using the reports that they create, publish, and share.
- **Data-level auditing**: Methods that data creators can use to track the performance and usage patterns of data assets that they create, publish, and share.
- **Tenant-level auditing**: Key decisions and actions administrators can take to create an end-to-end auditing solution.
- **Tenant-level monitoring**: Tactical actions administrators can take to monitor the Power BI service, updates, and announcements.

Related content

In the next article in this series, learn about report-level auditing.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Report-level auditing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This report-level auditing article is targeted at multiple audiences:

- **Report creators**: Users who need to understand usage, adoption, and performance of the reports that they've created, published, and shared.
- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators might need to collaborate with IT, security, internal audit, and other relevant teams.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators and other relevant teams.

The concepts covered in this article apply primarily to solutions created for three content delivery scopes, specifically enterprise BI, departmental BI, and team BI. Creators of personal BI solutions might find the information in this article useful as well; however, they're not the primary target.

This article focuses on auditing and monitoring of reports and visuals. However, achieving good performance for a report and its visuals isn't possible when the underlying semantic model and/or data source doesn't perform well. For information about auditing and monitoring of semantic models, dataflows, and datamarts, see Data-level auditing.

This article is the first article in the auditing and monitoring series because it describes built-in tools that content creators are likely to discover first. Ideally, you create shared semantic models (intended for reuse among many reports) before users create reports. Therefore, we recommend that you read this article together with the Data-level auditing article.

> **Tip**

Whether you're conversing with colleagues or reading online, you'll need to discern whether the term *report* is used literally or more generally. Often, it's used in a general way to refer to a single Power BI Desktop file (.pbix). The file might contain a data model (which when published becomes a semantic model), a report, or both. The term can be used literally to refer to a report only (for example, a report with a Live Connection to a semantic model). In this article, the term is used literally.

Report performance targets

To effectively monitor reports, we recommend that you define what report performance targets, like *excellent performance, good performance,* and *poor performance*, mean to your organization. There aren't any universal definitions. You should always consider these targets from the consumer's perspective.

Ideally, performance is a primary concern during the report design process. Here are several situations when you might choose to set performance targets.

- When validating or reviewing a new report (especially when you expect it to have a content delivery scope to a large number of users).
- Before endorsing a report (particularly when it's to be certified).
- Prior to publishing a report to a production workspace.
- When including a report in a Power BI app.

You might choose to create a standard performance target that's intended to apply to all reports throughout the organization. For example, the first report page should render within five seconds. However, because there are so many different considerations, it's not typically realistic to expect that every solution should meet the same target. Consider ranges for your performance targets that factor in the complexity level of the solution.

Checklist - When considering how report creators should verify report performance, key decisions and actions include:

- ✓ **Identify report performance targets**: Ensure that you have a good understanding of what acceptable report performance means from the consumer's perspective.
- ✓ **Document and communicate performance targets**: If there are specific targets, make sure they are communicated to the report creators in your organization.

Provide helpful information so that report creators understand how to measure performance, and how to apply design techniques that improve performance.

The remainder of this article describes techniques that you can use to audit and monitor report performance.

Report usage metrics

The main auditing resource available to report creators are the usage metrics reports, which are built into the Power BI service.

The primary objective of the usage metrics reports is to assess the impact of one report, or all reports in a workspace. Because it's focused on report views and performance of reports and dashboards (rather than other items, such as semantic models and dataflows), it's targeted at report creators.

Use the usage metrics reports to:

- Determine which users are most actively viewing reports.
- Understand how often reports are viewed and rank those reports by popularity (based on usage).
- Determine which report pages users access most frequently.
- Find reports that haven't been viewed recently.
- View high-level report performance statistics. These statistics can help guide report design optimization efforts, and identify reports that could have intermittent or consistent performance issues.
- Understand which consumption methods (for example, browser or Power BI mobile app) report consumers use. This information can help report creators decide how much effort to put into optimizing reports for mobile use.

> **Tip**
>
> Power BI captures usage metrics for activity that occurs for content that's been published to the Power BI service (including when it's rendered by using **Power BI Embedded**). Access to usage metrics is just one reason to encourage report creators to publish their reports to the Power BI service, rather than sharing Power BI Desktop files.

Usage metrics are built into the Power BI service, which is a key advantage because report creators don't need to set up a process to extract and store the usage data. It's fast and simple for them to get started.

Another advantage of the usage metrics is that the internal semantic model (that contains the usage metrics data) includes information that's not easily found elsewhere. For example, it includes views per report page and report opening time duration. The report page views are obtained by using client telemetry, which has limitations. Client telemetry (used by report usage metrics) is different from server-side telemetry data (used by the activity log).

Usage metrics include an internal semantic model and a report. While the internal semantic model can't be edited or customized, you can customize the usage metrics report. You can also update the report filters to learn about usage for all reports in a workspace (rather than just one report). By using this approach, the broadest range available is one workspace. You can view up to 30 days of history, including the most recent fully completed day.

> **Important**
>
> The Power BI activity log is a better alternative when you want to:
>
> - Retrieve user activities for more than one workspace.
> - Extract and retain activity data for more than 30 days.
> - Analyze all activities that users perform in the Power BI service.
>
> For more information about the activity log, see **Tenant-level auditing**.

The usage metrics reports are available to report creators and owners who are assigned to the Contributor, Member, or Admin workspace role. To make the usage metrics reports visible to workspace viewers (content consumers), you can create a copy of the usage report and customize it.

> **Tip**
>
> For more information about workspace roles, see the **Content creator security planning** article.

There are two tenant settings related to usage metrics.

- The *Usage metrics for content creators* tenant setting controls which groups of report creators (who also have the necessary workspace role) can generate and view the usage metrics reports. Commonly, Power BI administrators leave this setting enabled for the entire organization. That way, all self-service report creators can view the usage patterns for their content.

- The *Per-user data in usage metrics for content creators* tenant setting determines whether the names and email addresses of report consumers are displayed in the usage metrics reports. When this setting is disabled (for some or all report creators), Power BI suppresses names and email addresses in the usage metrics reports, which is referred to as *user masking*. Most often, Power BI administrators leave this setting enabled so that report creators can understand exactly who's using their reports. Also, the ability to contact other users directly for feedback about the content is valuable because it can help to improve the content. Occasionally, you might have a security need to mask user information for certain groups of report creators. When the setting is disabled, the report creator sees *unnamed user* in place of the user details.

The *ViewUsageMetrics* operation in the Power BI activity log allows Power BI administrators to monitor which content creators and owners are using the usage metrics reports. You can use that information to guide training and documentation efforts.

Checklist - When planning for using the usage metrics report, key decisions and actions include:

- ✓ **Confirm usage metrics are enabled**: Decide whether any Power BI report creator (who has permission to edit the report) can view usage metrics. Set the *Usage metrics for content creators* tenant setting to align with this decision.
- ✓ **Decide whether per-user data is displayed in usage metrics**: Determine whether names and email can be shown to all or some users. Set the *Per-user data in usage metrics for content creators* tenant setting to align with this decision.
- ✓ **Verify workspace roles**: Validate the workspace role assignments. Ensure that appropriate report creators and owners have permission to edit content in the workspace (thereby making the usage metrics reports available).
- ✓ **Create and customize the usage metrics reports**: For content you want to analyze, generate a usage metrics report. When appropriate, customize the usage metrics report to include all reports in the workspace.
- ✓ **Include in documentation and training for report creators**: Include guidance for your report creators about how they can take advantage of the usage metrics reports. Ensure that report creators understand the use cases and key limitations. Include examples of key metrics they can track, and how they can use the information to continually improve the solutions they create and publish.

- ✓ **Monitor who's using usage metrics**: Use the Power BI activity log to track which content creators and owners are using the usage metrics reports.
- ✓ **Determine whether usage metrics is sufficient**: Consider the situations when the built-in usage metrics report would be sufficient. Determine whether data-level and tenant-level auditing solutions (described in other articles in this series) would be more appropriate.

Performance Analyzer

Performance Analyzer is a tool available in Power BI Desktop to help you investigate and monitor report performance. It can help report creators understand the performance of visuals and DAX formulas.

> **Tip**
>
> In addition to Performance Analyzer, there are other tools that you can use to troubleshoot report performance issues. For example, you can troubleshoot specific report consumption issues that impact a Premium capacity by using the **Fabric Capacity Metrics app** or the semantic model event logs that are sent to Azure Log Analytics. For more information about these tools (and other tools), see **Data-level auditing**.

Performance Analyzer captures operations while a user interacts with a report in Power BI Desktop. It produces a log that records how each report element performs and for each interaction. For example, when you interact with a report slicer, cross-filter a visual, or select a page, the action and time duration are recorded in the log. Depending on the type of operation, other details are recorded too.

Summarized information is available in the **Performance Analyzer** pane. You can export log results to a JSON file, allowing you to follow through with more in-depth analysis. The export file contains more information about the logged operations. For more information about using the export file, see the Performance Analyzer documentation on GitHub.

> **Important**
>
> Keep in mind that Performance Analyzer runs within Power BI Desktop. The environment of the report creator's machine might differ from the environment of the Power BI service.
>
> Some common differences that you should account for include:

- Data volume in the underlying semantic model
- The number of concurrent users viewing the report
- Table storage mode(s)
- Whether a data gateway is used
- Whether a Power BI Premium capacity is involved
- Whether query caching is enabled
- Whether query parallelization is used
- The number of active connections
- Whether row-level security (RLS) is enforced by the Power BI service.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Data is logged when a user interacts with a report element. Logged data includes more than the visual display elements. It also includes:

- Visual display activity.
- DAX queries (when the visual retrieves data from the data model instead of the cache).
- DirectQuery activity (when applicable).
- Other activities performed by a visual, such as query preparation, background processing activities, and wait time.

Depending on their experience level, and how roles and responsibilities are divided, a report creator might need assistance to resolve performance issues. That's especially true when trying to understand why a query or calculation is slow. Assistance for a report creator could come in the form of:

- **Collaborating with a data creator**: The root cause of performance issues is often related to the design of the data model.
- **User support**: Assistance is often intra-team support from close colleagues or internal community support from other Power BI users in the organization. In some situations, it could also involve help desk support.

- **Skills mentoring from the Center of Excellence**: Assistance could also be in the form of skills mentoring activities, such as office hours.

Some organizations have specific requirements for endorsed (certified or promoted) reports. That's particularly true for reports that are widely used throughout the organization. In that case, you might be required (or encouraged) to verify Performance Analyzer results before publishing the report, or before it's certified.

> **Tip**
>
> Well-performing reports have a positive impact on **solution adoption**. We recommend that you encourage report creators to test report performance before publishing a new solution to the Power BI service. You should also encourage them to retest performance when significant changes are made to an existing solution (report or semantic model).
>
> For more information about optimization techniques, see **Optimization guide for Power BI**.

Checklist - When considering how report creators should use Performance Analyzer, key decisions and actions include:

- ✓ **Create documentation and training for report creators**: Include guidance for your report creators about what performance targets exist and how they can validate, measure, and test performance. Provide guidance to your report creators about how to create well-performing reports. Help new report creators adopt good design habits early.
- ✓ **Ensure support and skills mentoring are available**: Ensure that your report creators know how to get assistance to resolve performance issues.
- ✓ **Include in requirements to certify reports**: Decide whether you want to include Performance Analyzer results as a prerequisite to certifying (endorsing) reports. If so, ensure that this requirement is documented and communicated to report creators.

Related content

In the next article in this series, learn about data-level auditing.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Data-level auditing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This data-level auditing article is targeted at multiple audiences:

- **Data creators and workspace administrators**: Users who need to understand usage, adoption, and performance of the semantic models, dataflows, and datamarts that they create, publish, and share.
- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators might need to collaborate with IT, security, internal audit, and other relevant teams. Power BI administrators might also need to collaborate with content creators when troubleshooting performance.
- **Power BI capacity administrators**: The administrators responsible for overseeing Premium capacity in the organization. Power BI capacity administrators might need to collaborate with content creators when troubleshooting performance.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators and other relevant teams.
- **System administrators**: The team that's responsible for creating and securing Azure Log Analytics resources, and the database administrators who manage data sources.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.

For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

The concepts covered in this article apply primarily to solutions created for three content delivery scopes, specifically enterprise BI, departmental BI, and team BI. Creators of personal BI solutions might find the information in this article useful as well; however, they're not the primary target.

Achieving good performance in reports and visuals isn't possible when the underlying semantic model and/or data source isn't performing well. This article focuses on auditing and monitoring of semantic models, dataflows, and datamarts. It's the second article in the auditing and monitoring series because the tools and techniques are more complex than what's described in the Report-level auditing article. Ideally, you create shared semantic models (intended for reuse among many reports) before users create reports. Therefore, we recommend that you read this article together with the Report-level auditing article.

Because Power BI semantic models are built upon the Analysis Services tabular engine, you can connect to a local data model (in Power BI Desktop) or a Premium semantic model (in the Power BI service) as if it's an Analysis Services database. Therefore, many of the auditing and monitoring capabilities of Analysis Services are supported for Power BI Premium semantic models.

> **Note**
>
> For more information about models hosted in Analysis Services, see Monitoring overview.

The remainder of this article primarily focuses on models published to the Power BI service.

Semantic model event logs

Over time, data creators and owners might experience situations with their semantic models. A semantic model can:

- Become more complex and include complex measures.
- Grow larger in data volume.
- Consume more memory (sometimes unnecessarily when poor design decisions were made).
- Use more diverse data sources, and more complex table relationships.

- Include more row-level security (RLS) rules. For more information, see Enforce data security based on consumer identity.
- Have more reports that depend on it. For more information about using live connections with a shared semantic model, see the managed self-service BI usage scenario.
- Have more downstream data models that depend on it. For more information about using DirectQuery for Power BI semantic models and Analysis Services with a shared semantic model, see the customizable managed self-service BI usage scenario.
- Experience slower query execution and slower data refresh times.
- Contribute to slower rendering of reports and visuals.

To ensure usability, good performance, and adoption of the content they create, you should audit the usage and performance of the data assets you're responsible for managing. You can use the dataset event logs, which capture user-generated and system-generated activities that occur for a semantic model. They're also referred to as *trace events*, *dataset logs*, or *dataset activity logs*. System administrators often call them *low-level trace events* because they're detailed.

> **Note**
>
> The **dataset name change** has been rolled out in the Power BI service and in documentation, though there might be some instances—like with event log **operation names**—where the change hasn't occurred yet.

You should analyze semantic model trace events to:

- Audit all activities that occurred on a semantic model.
- Troubleshoot and optimize semantic model performance, memory usage, and query efficiency.
- Investigate semantic model refresh details and duration.
- Monitor Power Query formula language (M queries) sent by Power Query.
- Monitor DAX formulas and expressions sent to the semantic model (Analysis Services engine).
- Verify whether the correct storage mode was selected based on the workloads and the need to balance fresh data and optimal performance.
- Audit which row-level security roles are invoked, for which users, and on which semantic models.
- Understand the number of concurrent users.
- Validate a semantic model (for example, to verify data quality and performance before endorsing a semantic model, or before publishing it to a production

workspace).

The events generated by a Power BI semantic model are derived from existing diagnostic logs available for Azure Analysis Services. There are many types of trace events that you can capture and analyze, which are described in the following sections.

Azure Log Analytics

Azure Log Analytics is a component of the Azure Monitor service. Azure Log Analytics integration with Power BI allows you to capture semantic model events from all semantic models in a Power BI workspace. It's supported only for Premium workspaces. After you set up integration and the connection is enabled (for a Power BI Premium workspace), semantic model events are automatically captured and continually sent to an Azure Log Analytics workspace. The semantic model logs are stored in Azure Data Explorer, which is an append-only database that's optimized for capturing high-volume, near-real time telemetry data.

You assign a Power BI Premium workspace to a Log Analytics workspace in Azure. You must create a new Log Analytics resource in your Azure subscription to enable this type of logging.

Logs from one or more Power BI workspaces will be sent to a target Log Analytics workspace. Here are some ways you can choose to organize the data.

- **One target workspace for all audit data**: Store all the data in one Log Analytics workspace. That's helpful when the same administrator or users will access all data.
- **Target workspaces organized by subject area**: Organize the content by subject area. This technique is particularly helpful when different administrators or users are permitted to access the audit data from Azure Log Analytics. For example, when you need to segregate sales data from operations data.
- **One target workspace for each Power BI workspace**: Set up a one-to-one relationship between a Power BI workspace and an Azure Log Analytics workspace. That's useful when you have particularly sensitive content, or when the data is subject to specific compliance or regulatory requirements.

> **Tip**
>
> Thoroughly review the **documentation** and **frequently asked questions** on this functionality so that you're clear on what's possible and that you understand the technical requirements. Before making this functionality broadly available to

workspace administrators in your organization, consider doing a technical proof of concept (POC) with one Power BI workspace.

> **Important**
>
> Although the names are similar, the data captured by Azure Log Analytics isn't the same as the Power BI activity log. Azure Log Analytics captures detail-level trace events from the Analysis Services engine. Its sole purpose is to help you analyze and troubleshoot semantic model performance. Its scope is at the workspace level. Conversely, the purpose of the activity log is to help you understand how often certain **user activities** occur (such as editing a report, refreshing a semantic model, or creating an app). Its scope is the entire Power BI tenant.
>
> For more information about the user activities you can audit for your Power BI tenant, see **Tenant-level auditing**.

The *Azure Log Analytics connection for workspace administrators* tenant setting controls which groups of users (who also have the necessary workspace admin role) can connect a Power BI workspace to an existing Azure Log Analytics workspace.

Before you can set up integration, you must meet security prerequisites. Therefore, consider enabling the Power BI tenant setting only for Power BI workspace administrators who also have the required permissions in Azure Log Analytics, or who can obtain those permissions upon request.

> **Tip**
>
> Collaborate with your Azure administrator early in the planning process, especially when getting approval to create a new Azure resource is a challenge in your organization. You'll also need to plan for the security prerequisites. Decide whether to grant permission to your Power BI workspace administrator in Azure, or whether to grant permission to the Azure administrator in Power BI.

The semantic model logs captured by Azure Log Analytics include the semantic model queries, query statistics, detailed refresh activity, CPU time consumed on Premium capacities, and more. Because they're detail-level logs from the Analysis Services engine, the data can be verbose. Large data volumes are common for large workspaces that experience high semantic model activity.

To optimize cost when using Azure Log Analytics with Power BI:

- Connect Power BI workspaces to Azure Log Analytics only when you're actively troubleshooting, testing, optimizing, or investigating semantic model activity. When connected, a trace runs on all the semantic models in the workspace.
- Disconnect Azure Log Analytics from a Power BI workspace when you no longer need to actively troubleshoot, test, optimize, or investigate semantic model activity. By disconnecting, you're terminating the trace from running on all the semantic models in the workspace.
- Make sure you understand the cost model for how Azure Log Analytics bills for data ingestion, storage, and queries.
- Don't store the data in Log Analytics for longer than the default 30-day retention period. That's because semantic model analysis typically focuses on immediate troubleshooting activities.

There are several ways to access the events that are sent to Azure Log Analytics. You can use:

- The prebuilt Log Analytics for Power BI Semantic Models template app.
- The Power BI Desktop connector for Azure Data Explorer (Kusto). Use the Kusto Query Language (KQL) to analyze the data that's stored in Log Analytics. If you have SQL query experience, you'll find many similarities with KQL.
- The web-based query experience in Azure Data Explorer.
- Any query tool that can run KQL queries.

> **Tip**
>
> Because there's a high volume of semantic model trace events, we recommend that you develop a DirectQuery model to analyze the data. A DirectQuery model allows you to query the data in near-real time. The events usually arrive within five minutes.

For more information, see Govern Azure connections.

Checklist - When planning to use Azure Log Analytics, key decisions and actions include:

- ✓ **Consider a technical POC**: Plan for a small project to ensure that you fully understand the technical requirements, security requirements, which events to capture, and how to analyze the logs.

- ✓ **Decide which workspaces should be integrated with Log Analytics**: Determine which Premium workspaces contain semantic models that you're interested to analyze.
- ✓ **Decide whether Log Analytics should be enabled full-time for any workspaces**: For cost optimization, determine whether there are situations (or specific workspaces) where logging should be enabled permanently. Decide whether workspaces should be disconnected when troubleshooting isn't occurring.
- ✓ **Decide how long to retain Log Analytics data**: Determine whether there's a need to set a longer retention period than the 30-day default.
- ✓ **Clarify the process for requesting new Log Analytics workspace**: Collaborate with your Azure administrator to clarify how requests for a new Log Analytics resource should be submitted by Power BI workspace administrators.
- ✓ **Decide how security will work**: Collaborate with your Azure administrator to decide whether it's more feasible for a Power BI workspace administrator to be granted rights to an Azure Log Analytics workspace, or for a Azure administrator to be granted rights to a Power BI workspace. As you make this security decision, consider your plan to connect and disconnect workspaces regularly (for cost optimization).
- ✓ **Decide how to organize the target Log Analytics workspaces**: Consider how many Azure Log Analytics workspaces will be appropriate to organize the data from one or more Power BI workspaces. Align this decision with your security decisions for who can access the log data.
- ✓ **Decide which workspace administrators are allowed to connect**: Determine which groups of workspace administrators can connect a Power BI workspace to a Log Analytics workspace. Set the *Azure Log Analytics connection for workspace administrators* tenant setting to align with this decision.
- ✓ **Create the Azure Log Analytics resource**: Collaborate with your Azure administrator to create each Log Analytics workspace. Verify and update the permissions that are assigned in Azure to ensure that the Power BI configuration can occur without any issues. Validate that the data stored in Azure is in the correct geographic region.
- ✓ **Set the Log Analytics connection for each Power BI workspace**: Collaborate with your Power BI workspace administrators to set up the connection to Log Analytics for each Power BI workspace. Verify that the log data is flowing correctly to the Log Analytics workspace.
- ✓ **Create queries to analyze the data**: Set up KQL queries to analyze the data in Log Analytics based on your use case and current needs.
- ✓ **Include guidance for Power BI workspace administrators**: Provide information and prerequisites to your Power BI workspace administrators for how to request a new Log Analytics workspace and how to connect to a Power BI workspace. Also, explain when it's appropriate to disconnect a Power BI workspace.

- ✓ **Provide guidance and sample queries for analyzing the data**: Create KQL queries for workspace administrators to make it easier for them to get started with analyzing the data that's been captured.
- ✓ **Monitor costs**: Collaborate with your Azure administrator to monitor Log Analytics costs on an ongoing basis.

SQL Server Profiler

You can use SQL Server Profiler (SQL Profiler) to capture Power BI semantic model events. It's a component of SQL Server Management Studio (SSMS). Connectivity to a Power BI semantic model is supported with SSMS because it's based on the Analysis Services architecture that originated in SQL Server.

You can use SQL Profiler during different stages of the lifecycle of a semantic model.

- **During data model development**: SQL Profiler can connect to a data model in Power BI Desktop as an external tool. This approach is useful for data modelers who want to validate their data model, or do performance tuning.
- **After the semantic model is published to the Power BI service**: SQL Profiler can connect to a semantic model in a Premium workspace. SSMS is one of many supported client tools that can use the XMLA endpoint for connectivity. This approach is useful when you want to audit, monitor, validate, troubleshoot, or tune a published semantic model in the Power BI service.

It's also possible to use SQL Profiler as an external tool within DAX Studio. You can use DAX Studio to start a profiler trace, parse the data, and format the results. Data modelers who use DAX Studio often prefer this approach versus using SQL Profiler directly.

> ⓘ **Note**
>
> Using SQL Profiler is a different use case to the activity of *profiling data*. You **profile data** in the Power Query Editor to gain a deeper understanding of its characteristics. While data profiling is an important activity for data modelers, it's not in scope for this article.

Consider using SQL Profiler instead of Azure Log Analytics when:

- Your organization doesn't allow you to use or create Azure Log Analytics resources in Azure.
- You want to capture events for a data model in Power BI Desktop (that hasn't been published to a Premium workspace in the Power BI service).

- You want to capture events for one semantic model for a short period of time (rather than all semantic models in a Premium workspace).
- You want to capture certain events only during a trace (such as only the *Query End* event).
- You want to start and stop traces on a frequent basis (like when you need to capture semantic model events that are occurring now).

Like Azure Log Analytics (described earlier in this article), semantic model events captured by SQL Profiler are derived from existing diagnostic logs available for Azure Analysis Services. However, there are some differences in the events that are available.

> **Tip**
>
> The use of SQL Profiler for monitoring Analysis Services is covered in many books, articles, and blog posts. Most of that information is relevant for monitoring a Power BI semantic model.

> **Important**
>
> You can also use SQL Profiler to monitor queries sent from the Power BI service to the underlying data sources (for example, to a SQL Server relational database). However, the capability to trace a relational database is deprecated. Connecting to the Analysis Services engine is **supported** and *not* deprecated. If you're familiar with **Analysis Services extended events** and you prefer to use them, connectivity from SSMS is possible for a data model in Power BI Desktop. However, it's not supported for Power BI Premium. Therefore, this section focuses only on standard SQL Profiler connectivity.

The *Allow XMLA endpoints and Analyze in Excel with on-premises semantic models* tenant setting controls which groups of users (who are also assigned to the Contributor, Member, or Admin workspace role, or the Build permission for the individual semantic model) can use the XMLA endpoint to query and/or maintain semantic models in the Power BI service. For more information about using the XMLA endpoint, see the advanced data model management usage scenario.

> **Note**
>
> You can also use SQL Profiler to help debug and troubleshoot specific DAX expressions. You can connect SQL Profiler to Power BI Desktop as an **external tool**. Look for the *DAX Evaluation Log* event class to view intermediary results of a DAX

expression. That event is generated when you use the EVALUATEANDLOG DAX function in a model calculation.

This function is only intended for development and test purposes. You should remove it from your data model calculations before publishing the data model to a production workspace.

Checklist - When planning to use SQL Profiler, key decisions and actions include:

- ✓ **Decide who can have SSMS or DAX Studio installed**: Determine whether you'll allow all the Power BI content creators in your organization to install SSMS and/or DAX Studio so they can use SQL Profiler. Decide whether these ancillary tools are installed upon request, or part of a standard set of software that's installed for approved data creators in the organization.
- ✓ **Add SQL Profiler to the External Tools menu in Power BI Desktop**: If data creators will use SQL Profiler often, ask IT to automatically add it to the External Tools menu in Power BI Desktop for these users.
- ✓ **Decide who can use the XMLA endpoint**: Determine whether all users are permitted to connect to published semantic models by using the XMLA endpoint, or whether it's limited to approved data creators only. Set the *Allow XMLA endpoints and Analyze in Excel with on-premises semantic models* tenant setting to align with this decision.
- ✓ **Provide guidance and sample queries for analyzing the data**: Create documentation for your data creators so they understand the recommended way to audit and monitor semantic models. Provide guidance for common use cases to make it easier for them to get started gathering and analyzing trace data.

Data model metadata

Because Power BI semantic models are built upon the Analysis Services engine, you have access to the tools that can query the metadata of a data model. Metadata includes everything about the data model, including table names, column names, and measure expressions.

Dynamic management views

The Analysis Services Dynamic Management Views (DMVs) can query the data model metadata. You can use the DMVs to audit, document, and optimize your data models at a point in time.

Specifically, you can:

- Audit the data sources used by a model.
- Discover which objects are consuming the most memory in a model.
- Determine how efficiently column data can be compressed.
- Find columns in a model that aren't used.
- Audit active user sessions and connections.
- Verify the structure of the model.
- Review DAX expressions used by calculated tables, calculated columns, measures, and row-level security (RLS) rules.
- Identify dependencies between objects and measures.

> **Tip**
>
> The DMVs retrieve information about the *current state* of a semantic model. Think of the data returned by DMVs as a snapshot of what's occurring at a point in time. Conversely, the semantic model event logs (described earlier in this article) retrieve information about what *activities* occurred for a semantic model while a trace connection was active.

SSMS is a tool commonly used to run DMV queries. You can also use the Invoke-ASCmd PowerShell cmdlet to create and execute XMLA scripts that query the DMVs.

Third-party tools and external tools are also popular with the Power BI community. These tools use the publicly documented DMVs to simplify access and to work with data returned by the DMVs. One example is DAX Studio, which includes explicit functionality to access the DMVs. DAX Studio also includes a built-in *View Metrics* feature, which is commonly known as *Vertipaq Analyzer*. Vertipaq Analyzer has a user interface for analyzing the structure and size of tables, columns, relationships, and partitions in a data model. You can also export (or import) the data model metadata to a .vpax file. The exported file only contains metadata about the data model structure and size, without storing any model data.

> **Tip**
>
> Consider sharing a .vpax file with someone when you need assistance with a data model. That way, you won't share the model data with that person.

You can use DMV queries during different stages of the lifecycle of a semantic model.

- **During data model development**: Your tool of choice can connect to a data model in Power BI Desktop as an external tool. This approach is useful for data modelers who want to validate their data model, or do performance tuning.
- **After the semantic model is published to the Power BI service**: Your tool of choice can connect to a semantic model in a Premium workspace. SSMS is one of many supported client tools that use the XMLA endpoint for connectivity. This approach is useful when you want to audit or validate a published semantic model in the Power BI service.

> **Tip**
>
> If you decide to write your own DMV queries (for example, in SSMS), be aware that the DMVs don't support all SQL operations. Also, some DMVs aren't supported in Power BI (because they require Analysis Services server administrator permissions that aren't supported by Power BI).

The *Allow XMLA endpoints and Analyze in Excel with on-premises semantic models* tenant setting controls which groups of users (who are also assigned to the Contributor, Member, or Admin workspace role, or the Build permission for the individual semantic model) can use the XMLA endpoint to query and/or maintain semantic models in the Power BI service.

For more information about using the XMLA endpoint, third-party tools, and external tools, see the advanced data model management usage scenario.

Best Practice Analyzer

Best Practice Analyzer (BPA) is a feature of Tabular Editor, which is a third-party tool that's achieved widespread adoption by the Power BI community. BPA includes a set of customizable rules that can help you audit the quality, consistency, and performance of your data model.

> **Tip**
>
> To set up BPA, download the set of best practice rules, which are provided by Microsoft on **GitHub**.

Primarily, BPA can help you improve consistency of models by detecting suboptimal design decisions that can reduce performance issues. It's helpful when you have self-

service data modelers distributed throughout different areas of the organization.

BPA can also help you audit and govern your data models. For example, you can verify whether a data model includes any row-level security (RLS) roles. Or, you can validate whether all model objects have a description. That's helpful when, for example, your goal is to ensure that a data model includes a data dictionary.

BPA can expose design issues that can help the Center of Excellence determine whether more training or documentation is necessary. It can take action to educate data creators on best practices and organizational guidelines.

> **Tip**
>
> Keep in mind that BPA can detect the *existence* of a characteristic (such as row-level security). However, it might be difficult to determine whether it's set up correctly. For that reason, a subject matter expert might need to conduct a review . Conversely, the *non-existence* of a particular characteristic doesn't necessarily mean a bad design; the data modeler might have a good reason for producing a particular design.

Checklist - When planning to access metadata for data models, key decisions and actions include:

- ✓ **Decide who can have SSMS installed**: Determine whether you'll allow all Power BI content creators in your organization to install SSMS so that they can connect to published semantic models. Decide whether it's installed upon request, or as part of a standard set of software that's installed for approved data creators in the organization.
- ✓ **Decide who can have third-party tools installed**: Determine whether you'll allow all Power BI content creators in your organization to install third-party tools (such as DAX Studio and Tabular Editor) so that they can monitor local data models and/or published semantic models. Decide whether they're installed upon request, or as part of a standard set of software that's installed for approved data creators in the organization.
- ✓ **Set up best practice rules**: Decide which Best Practice Analyzer rules can scan the data models in your organization.
- ✓ **Decide who can use the XMLA endpoint**: Determine whether all users are permitted to connect to semantic models by using the XMLA endpoint, or whether

it's limited to approved data creators only. Set the *Allow XMLA endpoints and Analyze in Excel with on-premises semantic models* tenant setting to align with this decision.

- ✓ **Provide guidance for content creators**: Create documentation for your data creators so that they understand the recommended way(s) to analyze semantic models. Provide guidance for common use cases to make it easier for them to start gathering and analyzing DMV results and/or using Best Practice Analyzer.

Data model and query performance

Power BI Desktop includes several tools that help data creators troubleshoot and investigate their data models. These capabilities are targeted at data modelers who want to validate their data model, and do performance tuning before publishing to the Power BI service.

Performance Analyzer

Use Performance Analyzer, which is available in Power BI Desktop, to audit and investigate performance of a data model. Performance Analyzer helps report creators measure the performance of individual report elements. Commonly, however, the root cause of performance issues is related to data model design. For this reason, a semantic model creator can benefit from using Performance Analyzer too. If there are different content creators responsible for creating reports versus semantic models, it's likely that they'll need to collaborate when troubleshooting a performance issue.

> **Tip**
>
> You can use **DAX Studio** to import and analyze the log files generated by Performance Analyzer.

For more information about Performance Analyzer, see Report-level auditing.

Query Diagnostics

Use Query Diagnostics, which are available in Power BI Desktop, to investigate the performance of Power Query. They're useful for troubleshooting, and for when you need to understand what the Power Query engine is doing.

The information you can gain from Query Diagnostics includes:

- Extra detail related to error messages (when an exception occurs).

- The queries that are sent to a data source.
- Whether query folding is or isn't occurring.
- The number of rows returned by a query.
- Potential slowdowns during a data refresh operation.
- Background events and system-generated queries.

Depending on what you're looking for, you can enable one or all the logs: aggregated, detailed, performance counters, and data privacy partitions.

You can start session diagnostics in Power Query Editor. Once enabled, query and refresh operations are collected until diagnostic tracing is stopped. The data is populated directly in the query editor as soon as the diagnostics are stopped. Power Query creates a *Diagnostics* group (folder), and adds several queries to it. You can then use standard Power Query functionality to view and analyze the diagnostics data.

Alternatively, you can enable a trace in Power BI Desktop in the *Diagnostics* section of the *Options* window. Log files are saved to a folder on your local machine. These log files are populated with the data after you close Power BI Desktop, at which time the trace is stopped. Once Power BI Desktop is closed, you can open the log files with your preferred program (such as a text editor) to view them.

Query evaluation and folding

Power Query supports various capabilities to help you understand query evaluation, including the query plan. It can also help you determine whether query folding is occurring for an entire query, or for a subset of steps in a query. Query folding is one of the most important aspects of performance tuning. It's also helpful to review the native queries sent by Power Query when you're monitoring a data source, which is described later in this article.

Premium metrics app

When troubleshooting, it can be helpful to collaborate with your Power BI Premium capacity administrator. The capacity administrator has access to the Power BI Premium utilization and metrics app. This app can provide you with a wealth of information about activities that occur in the capacity. That information can help you troubleshoot semantic model issues.

Tip

Your Premium capacity administrator can grant access to additional users (non-capacity administrators) to allow them to access the Premium metrics app.

The Premium metrics app comprises an internal semantic model and an initial set of reports. It helps you perform near-real-time monitoring of a Power BI Premium capacity (P SKU) or Power BI Embedded (A SKU) capacity. It includes data for the last two to four weeks (depending on the metric).

Use the Premium metrics app to troubleshoot and optimize semantic models. For example, you can identify semantic models that have a large memory footprint or that experience routinely high CPU usage. It's also a useful tool to find semantic models that are approaching the limit of your capacity size.

Checklist - When considering approaches to use for monitoring data model and query performance, key decisions and actions include:

- ✓ **Identify semantic model query performance targets**: Ensure that you have a good understanding of what good semantic model performance means. Determine when you'll require specific query performance targets (for example, queries to support reports must render within five seconds). If so, make sure the targets are communicated to the data creators in your organization.
- ✓ **Identify semantic model refresh performance targets**: Determine when you'll require specific data refresh targets (for example, completion of a data refresh operation within 15 minutes and prior to 5am). If so, make sure the targets are communicated to the data creators in your organization.
- ✓ **Educate your support team**: Ensure that your internal user support team is familiar with the diagnostic capabilities so they're ready to support Power BI users when they need help.
- ✓ **Connect your support team and database administrators**: Make sure that your support team knows how to contact the correct administrators for each data source (when troubleshooting query folding, for example).
- ✓ **Collaborate with your Premium capacity administrator**: Work with your capacity administrator to troubleshoot semantic models that reside in a workspace that's assigned to Premium capacity or Power BI Embedded capacity. When appropriate, request access to the Premium metrics app.
- ✓ **Provide guidance for content creators**: Create documentation for your data creators so that they understand what actions to take when troubleshooting.

✓ **Include in training materials**: Provide guidance to your data creators about how to create well-performing data models. Help them adopt good design habits early. Focus on teaching data creators how to make good design decisions.

Data source monitoring

Sometimes it's necessary to directly monitor a specific data source that Power BI connects to. For example, you might have a data warehouse that's experiencing an increased workload, and users are reporting performance degradation. Typically, a database administrator or system administrator monitors data sources.

You can monitor a data source to:

- Audit which users are sending queries to the data source.
- Audit which applications (like Power BI) are sending queries to the data source.
- Review what query statements are sent to the data source, when, and by which users.
- Determine how long it takes for a query to run.
- Audit how row-level security is invoked by the source system when it's using single sign-on (SSO).

There are many actions that a Power BI content creator might take once they analyze monitoring results. They could:

- Tune and refine the queries that are sent to the data source so they're as efficient as possible.
- Validate and tune the native queries that are sent to the data source.
- Reduce the number of columns that are imported into a data model.
- Remove high precision and high cardinality columns that are imported into a data model.
- Reduce the amount of historical data that's imported into a data model.
- Adjust the Power BI data refresh times to help spread out the demand for the data source.
- Use incremental data refresh to reduce the load on the data source.
- Reduce the number of Power BI data refreshes by consolidating multiple semantic models into a shared semantic model.
- Adjust automatic page refresh settings to increase the refresh frequency, and therefore reduce the load on the data source.
- Simplify calculations to reduce the complexity of queries sent to the data source.
- Change the data storage mode (for example, to import mode instead of DirectQuery) to reduce the consistent query load on the data source.

- Use query reduction techniques to reduce the number of queries that are sent to the data source.

System administrators might take other actions. They could:

- Introduce an intermediary data layer, such as Power BI dataflows (when a data warehouse isn't a viable option). Power BI content creators can use the dataflows as their data source instead of connecting directly to data sources. An intermediary data layer can reduce the load on a source system. It also has the added benefit of centralizing data preparation logic. For more information, see the self-service data preparation usage scenario.
- Change the data source location to reduce the impact of network latency (for example, use the same data region for the Power BI service, data sources, and gateways).
- Optimize the data source so it more efficiently retrieves data for Power BI. Several commons techniques include creating table indexes, creating indexed views, creating persisted computed columns, maintaining statistics, using in-memory or columnstore tables, and creating materialized views.
- Direct users to use a read-only replica of the data source, rather than an original production database. A replica might be available as part of a high availability (HA) database strategy. One advantage of a read-only replica is to reduce contention on the source system.

The tools and techniques that you can use to monitor data sources depend on the technology platform. For example, your database administrator can use extended events or the Query Store for monitoring Azure SQL Database and SQL Server databases.

Sometimes, Power BI accesses a data source through a data gateway. Gateways handle connectivity from the Power BI service to certain types of data sources. However, they do more than just connect to data. A gateway includes a mashup engine that performs processing and data transformations on the machine. It also compresses and encrypts the data so that it can be efficiently and securely transmitted to the Power BI service. Therefore, an unmanaged, or non-optimized, gateway can contribute to performance bottlenecks. We recommend that you talk to your gateway administrator for help with monitoring gateways.

> **Tip**
>
> Your Power BI administrator can compile a full tenant inventory (which includes lineage) and access user activities in the activity log. By correlating the lineage and user activities, administrators can identify the most frequently used data sources and gateways.

For more information about the tenant inventory and the activity log, see **Tenant-level auditing**.

Checklist - When planning to monitor a data source, key decisions and actions include:

- ✓ **Determine specific goals**: When monitoring a data source, get clarity about exactly what you need to accomplish and the goals for troubleshooting.
- ✓ **Collaborate with database administrators**: Work with your database or system administrator(s) to get their help when monitoring a specific data source.
- ✓ **Collaborate with gateway administrators**: For data sources that connect through a data gateway, collaborate with the gateway administrator when troubleshooting.
- ✓ **Connect your support team and database administrators**: Make sure that your support team knows how to contact the correct administrators for each data source (for example, when troubleshooting query folding).
- ✓ **Update training and guidance**: Include key information and tips for data creators about how to work with organizational data sources. Include information about what to do when something goes wrong.

Data refresh monitoring

A data refresh operation involves importing data from underlying data source(s) into a Power BI semantic model, dataflow, or datamart. You can schedule a data refresh operation or run it on-demand.

Service-level agreement

IT commonly uses service-level agreements (SLAs) to document the expectations for data assets. For Power BI, consider using an SLA for critical content or enterprise-level content. It commonly includes when users can expect updated data in a semantic model to be available. For example, you could have an SLA that all data refreshes must complete by 7am every day.

Semantic model logs

The semantic model event logs from Azure Log Analytics or SQL Profiler (described previously in this article) include detailed information about what's happening in a

semantic model. The captured events include semantic model refresh activity. The event logs are especially useful when you need to troubleshoot and investigate semantic model refreshes.

Premium capacity semantic models

When you have content that's hosted in a Power BI Premium capacity, you have more capabilities to monitor data refresh operations.

- The Power BI refresh summaries page in the admin portal includes a summary of the refresh history. This summary provides information about refresh duration and error messages.
- The Fabric Capacity Metrics app also includes helpful refresh information. It's useful when you need to investigate refresh activity for a Power BI Premium capacity (P SKU), Power BI Embedded (A SKU) capacity, or Fabric capacities.

Enhanced semantic model refreshes

Content creators can initiate semantic model refreshes programmatically by using enhanced refresh with the Refresh Dataset in Group Power BI REST API. When you use enhanced refresh, you can monitor the historical, current, and pending refresh operations.

Data refresh schedule monitoring

Power BI administrators can monitor data refresh schedules in the tenant to determine whether there are many refresh operations scheduled concurrently during a specific timeframe (for example, between 5am and 7am, which could be a particularly busy data refresh time). Administrators have permission to access the semantic model refresh schedule metadata from the metadata scanning APIs, which are known as the *scanner APIs*.

Power BI REST APIs

For critical semantic models, don't rely solely on email notifications for monitoring data refresh issues. Consider compiling the data refresh history in a centralized store where you can monitor, analyze, and act upon it.

You can retrieve data refresh history by using:

- The Get Refresh History in Group REST API to retrieve refresh information for a *workspace*.
- The Get Refreshables for Capacity REST API to retrieve refresh information for a *capacity*.

> **Tip**
>
> We strongly recommend that you monitor the refresh history of your semantic models to ensure that current data is available to reports and dashboards. It also helps you to know whether SLAs are being met.

Checklist - When planning for data refresh monitoring, key decisions and actions include:

- ✓ **Determine specific goals**: When monitoring data refreshes, get clarity about exactly what you need to accomplish and what the scope of monitoring should be (for example, production semantic models, certified semantic models, and others).
- ✓ **Consider setting up an SLA**: Determine whether an SLA would be useful to set expectations for data availability and when data refresh schedules should run.
- ✓ **Collaborate with database and gateway administrators**: Work with your database or system administrator(s), and gateway administrators, to monitor or troubleshoot data refresh.
- ✓ **Knowledge transfer for support team**: Make sure that your support team knows how to help content creators when data refresh issues arise.
- ✓ **Update training and guidance**: Include key information and tips for data creators about how to refresh data from organizational data sources and common data sources. Include best practices and organizational preferences for how to manage data refresh.
- ✓ **Use a support email address for notifications**: For critical content, set up refresh notifications to use a support email address.
- ✓ **Set up centralized refresh monitoring**: Use the Power BI REST APIs to compile data refresh history.

Dataflow monitoring

You create a Power BI dataflow with Power Query Online. Many of the query performance features, and the Power Query diagnostics, which were described earlier, are applicable.

Optionally, you can set workspaces to use Azure Data Lake Storage Gen2 for dataflow storage (known as *bring-your-own-storage*) rather than internal storage. When you use bring-your-own-storage, consider enabling telemetry so that you can monitor metrics for the storage account. For more information, see the self-service data preparation usage scenario, and the advanced data preparation usage scenario.

You can use the Power BI REST APIs to monitor dataflow transactions. For example, use the Get Dataflow Transactions API to check the status of dataflow refreshes.

You can track user activities for Power BI dataflows with the Power BI activity log. For more information, see Tenant-level auditing.

> **Tip**
>
> There are many best practices that you can adopt to optimize your dataflow designs. For more information, see **Dataflows best practices**.

Datamart monitoring

A Power BI datamart includes several integrated components, including a dataflow, a managed database, and a semantic model. Refer to the previous sections of this article to learn about auditing and monitoring of each component.

You can track user activities for Power BI datamarts by using the Power BI activity log. For more information, see Tenant-level auditing.

Related content

In the next article in this series, learn about tenant-level auditing.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Tenant-level auditing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This tenant-level audit planning article is primarily targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators might need to collaborate with IT, security, internal audit, and other relevant teams.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators and other relevant teams.

> **Important**
>
> We recommend that you closely follow the Power BI release plan to learn about future enhancements of the auditing and monitoring capabilities.

The purpose of a tenant-level audit solution is to capture and analyze data for all users, activities, and solutions deployed to a Power BI tenant. This tenant-level auditing data is valuable for many purposes, allowing you to analyze adoption efforts, understand usage patterns, educate users, support users, mitigate risk, improve compliance, manage license costs, and monitor performance.

Creating an end-to-end auditing solution that's secure and production-ready is a significant project that takes time. Think of it as building business intelligence on business intelligence (BI on BI). For information about why the auditing data is so valuable, see Auditing and monitoring overview.

For report-level auditing, which is targeted at report creators, see Report-level auditing.

For auditing data assets, which is targeted at data creators, see Data-level auditing.

The remainder of this article focuses on tenant-level auditing.

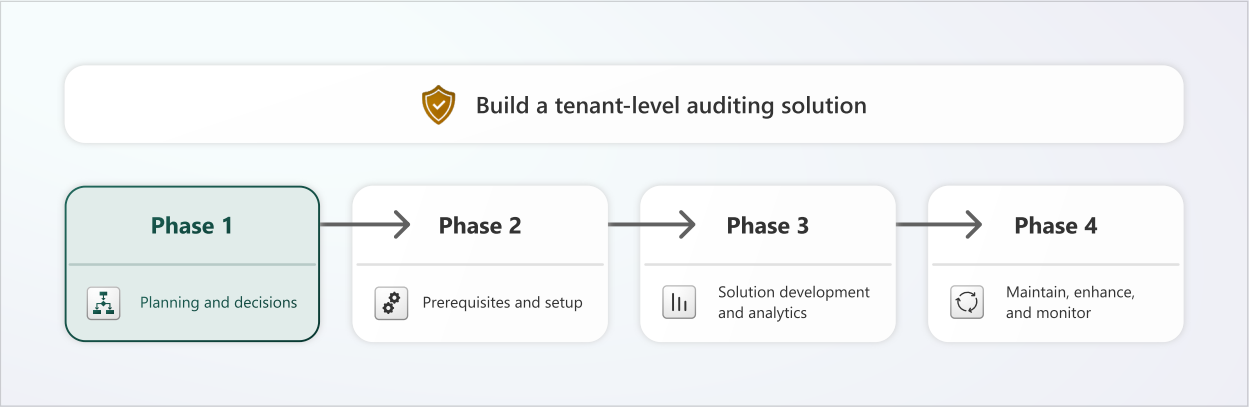
> **Tip**
>
> The primary focus of this article is to help you plan to create an end-to-end auditing solution. Because the content in this article is organized into four phases, you'll need information across multiple phases. For example, you'll find information in Phase 1 about planning to use a service principal, and information in Phase 2 about prerequisites and setup.
>
> Therefore, we recommend that you read this entire article first so that you'll be familiar with what's involved. Then you should be well-prepared to plan and build your auditing solution in an iterative manner.

When you plan to build a tenant-level auditing solution, plan to spend time on the following four phases.

- **Phase 1: Planning and decisions**
  - Requirements and priorities
  - Data needs
  - Type of auditing solution
  - Permissions and responsibilities
  - Technical decisions
  - Data source decisions
- **Phase 2: Prerequisites and setup**
  - Create storage account
  - Install software and set up services
  - Register a Microsoft Entra application
  - Set Fabric tenant settings
- **Phase 3: Solution development and analytics**
  - Extract and store the raw data
  - Create the curated data
  - Create a data model
  - Enhance the data model
  - Create analytical reports
  - Take action based on the data
- **Phase 4: Maintain, enhance, and monitor**
  - Operationalize and improve
  - Documentation and support
  - Enable alerting
  - Ongoing management

Phase 1: Planning and decisions

The first phase focuses on planning and decision-making. There are many requirements and priorities to consider, so we recommend that you spend sufficient time to understand the topics introduced in this section. The goal is to make good upfront decisions so that the downstream phases run smoothly.



Requirements and priorities

In the initial phase, the goal is to identify what's most important. Focus on what matters most, and how you're going to measure impact in your organization. Strive to define business-oriented requirements rather than technology-oriented requirements.

Here are some questions you should answer.

- **What key questions do you need to answer?** There's a large volume of auditing data you can explore. The most effective way to approach auditing is to focus on answering specific questions.
- **What are your adoption and data culture goals?** For example, perhaps you have a goal to increase the number of self-service BI content creators in the organization. In that case, you'll need to measure user activities related to creating, editing, and publishing content.
- **What immediate risks are present?** For example, you might know oversharing of content has occurred in the past. User training has since been enhanced, and you now want to audit security settings and activities on an ongoing basis.
- **Are there current key performance indicators (KPIs) or organizational goals?** For example, perhaps you have an organizational KPI that relates to digital transformation or becoming a more data-driven organization. Tenant-level auditing data can help you measure whether your Power BI implementation is aligned with these goals.

> **Tip**

Verify auditing requirements and set priorities with your **executive sponsor** and Center of Excellence. It's tempting to extract auditing data and start creating reports based on a lot of interesting data. However, it's unlikely that you'll derive high value from your auditing solution when it isn't driven by questions you need to answer and actions you intend to take.

For more ideas about ways that you can use auditing data, see Auditing and monitoring overview.

Checklist - When identifying requirements and priorities, key decisions and actions include:

- ✓ **Identify requirements**: Collect and document the key business requirements for auditing your Power BI tenant.
- ✓ **Prioritize requirements**: Set priorities for the requirements, classifying them as immediate, short-term, medium-term, and long-term (backlog).
- ✓ **Make a plan for the immediate priorities**: Put a plan in place to begin collecting data so that it's available when you need it.
- ✓ **Reassess requirements regularly**: Create a plan to reassess requirements on a regular basis (for example, twice per year). Verify whether the auditing and monitoring solution meets the stated requirements. Update action items on your plan as necessary.

Data needs

Once you've defined the requirements and priorities (described previously), you're ready to identify the data that you'll need. Four categories of data are covered in this section.

- User activity data
- Tenant inventory
- Users and groups data
- Security data

Most of the data that you'll need comes from the admin APIs, which provide a wealth of metadata about the Power BI tenant. For more information, see Choose a user API or admin API later in this article.

User activity data

Make it your first priority to obtain data about user activities. The *user activities*, which are also called *events* or *operations*, are useful for a wide variety of purposes.

Most often, this data is referred to as the *activity log* or the *event log*. Technically, there are several ways to acquire user activity data (the activity log being one method). For more information about the decisions and activities involved, see Access user activity data later in this article.

Here are some common questions that user activity data can answer.

- **Find top users and top content**
  - What content is viewed most often and by which users?
  - What are the daily, weekly, and monthly trends for viewing content?
  - Are report views trending up or down?
  - How many users are active?
- **Understand user behavior**
  - What type of security is used most often (apps, workspaces, or per-item sharing)?
  - Are users using browsers or mobile apps most often?
  - Which content creators are most actively publishing and updating content?
  - What content is published or updated, when, and by which users?
- **Identify user education and training opportunities**
  - Who is doing (too much) sharing from their personal workspace?
  - Who is doing a significant amount of exporting?
  - Who is regularly downloading content?
  - Who is publishing many new semantic models?
  - Who is using subscriptions heavily?
- **Improve governance and compliance efforts**
  - When are tenant settings changed, and by which Power BI administrator?
  - Who started a Power BI trial?
  - What content is accessed by external users, who, when, and how?
  - Who added or updated a sensitivity label?
  - What percentage of report views are based on certified semantic models?
  - What percentage of semantic models support more than one report?
  - When is a gateway or data source created or updated, and by which user?

For more information about ways to use this data, see Understand usage patterns.

> **Important**

If you're not already extracting and storing user activities data, make that an urgent priority. At a minimum, ensure that you extract the raw data and store it in a secure location. That way, the data will be available when you're ready to analyze it. History is available for 28 days or 90 days, depending on the source you use.

For more information, see Access user activity data later in this article.

Tenant inventory

Often, the second priority is to retrieve the data to create a *tenant inventory*. Sometimes it's referred to as *workspace inventory*, *workspace metadata*, or *tenant metadata*.

A tenant inventory is a snapshot at a given point in time. It describes what's published in the tenant. The tenant inventory doesn't include the actual data or the actual reports. Rather, it's metadata that describes all workspaces and their items (such as semantic models and reports).

Here are some common questions that the tenant inventory can answer.

- **Understand how much content you have and where**
  - Which workspaces have the most content?
  - What type of content is published in each workspace (particularly when you're separating reporting workspaces and data workspaces)?
  - What dependencies exist between items (such as dataflows that support many semantic models, or a semantic model that's a source for other composite models)?
  - What is the data lineage (dependencies between Power BI items, including external data sources)?
  - Are there orphaned reports (which are disconnected from their semantic model)?
- **Understand the ratio of semantic models to reports**
  - How much semantic model reuse is occurring?
  - Are there duplicate, or highly similar, semantic models?
  - Are there opportunities to consolidate semantic models?
- **Understand trends between points in time**
  - Is the number of reports increasing over time?
  - Is the number of semantic models increasing over time?
- **Find unused content**
  - Which reports are unused (or under-utilized)?
  - Which semantic models are unused (or under-utilized)?
  - Are there opportunities to retire content?

A tenant inventory helps you to use current names when analyzing historical activities. The snapshot of the items in your tenant contains the names of all items *at that point in time*. It's helpful to use current item names for historical reporting. For example, if a report was renamed three months ago, the activity log at that time records the old report name. With a properly defined data model, you can use the latest tenant inventory to locate an item by its current name (rather than its former name). For more information, see Create a data model later in this article.

For more information about ways to use a tenant inventory, see Understand published content.

> **Tip**
>
> You'll often use combine the user activity events (described in the **previous section**) and the tenant inventory to produce a complete picture. That way, you enhance the value of all of the data.

Because a tenant inventory is a snapshot at a given point in time, you'll need to decide how often to extract and store the metadata. A weekly snapshot is useful for making comparisons between the two points in time. For example, if you want to investigate security settings for a workspace, you'll need the previous tenant inventory snapshot, the activity log events, and the current tenant inventory snapshot.

There are two main ways to build a tenant inventory. For more information about the decisions and activities involved, see Access tenant inventory data later in this article.

Users and groups data

As your analytical needs grow, you'll likely determine that you'd like to include data about users and groups in your end-to-end auditing solution. To retrieve that data, you can use Microsoft Graph, which is the authoritative source for information about Microsoft Entra ID users and groups.

Data that's retrieved from the Power BI REST APIs includes an email address to describe the user, or the name of a security group. This data is a snapshot at a given point in time.

Here are some common questions that Microsoft Graph can answer.

- **Identify users and groups**
  - What's the full username (in addition to the email address), department, or location sourced from their user profile?

    - Who are the members of a security group?
    - Who's the owner of a security group (to manage members)?
- **Obtain additional user information**
    - Which licenses—Power BI Pro or Power BI Premium Per User (PPU)—are assigned to users?
    - Which users sign in most frequently?
    - Which users haven't signed in to the Power BI service recently?

> ⚠ **Warning**
>
> Some older methods (which are extensively documented online) for accessing users and groups data are deprecated and shouldn't be used. Whenever possible, use Microsoft Graph as the authoritative source of users and groups data.

For more information and recommendations about how to access data from Microsoft Graph, see Access user and groups data later in this article.

Security data

You might have a requirement to perform regular security audits.

Here are some common questions that the Power BI REST APIs can answer.

- **Identify people and applications**
    - Which reports does a user, group, or service principal have access to?
    - Which users, groups, or service principals are subscribers to receive reports with an email subscription?
- **Understand content permissions**
    - Which workspace roles are assigned to which users and groups?
    - Which users and groups are assigned to each Power BI app audience?
    - Which per-item permissions are assigned, for which reports, and for which users?
    - Which per-item permissions are assigned, for which semantic models, and for which users?
    - Which semantic models and datamarts have row-level security (RLS) defined?
    - Which items are shared to people in the entire organization?
    - Which items are published publicly on the internet?
- **Understand other permissions**
    - Who has permission to publish by using a deployment pipeline?
    - Who has permission to manage gateways and data connections?
    - Who has permission to manage a Premium capacity?

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

> **Tip**
>
> For more considerations about security, see the security planning articles.

These common questions aren't an exhaustive list; there are a wide variety of Power BI REST APIs available. For more information, see Using the Power BI REST APIs.

For more information about using the admin APIs versus user APIs (including how it affects permissions that are required for the user who's extracting the data), see Choose a user API or admin API later in this article.

Checklist - When identifying the data that's needed for auditing, key decisions and actions include:

- ✓ **Identify specific data needs for user activity data**: Validate the requirements you've collected. Identify which auditing data is necessary to meet each requirement for user activity data.
- ✓ **Identify specific data needs for tenant inventory data**: Validate the requirements you've collected. Identify which auditing data is necessary to compile a tenant inventory.
- ✓ **Identify specific data needs for users and groups data**: Validate the requirements you've collected. Identify which auditing data is necessary to meet each requirement for users and groups data.
- ✓ **Identify specific data needs for security data**: Validate the requirements you've collected. Identify which auditing data is necessary to meet each requirement for security data.
- ✓ **Verify priorities**: Verify the order of priorities so you know what to develop first.

- ✓ **Decide how often to capture activities**: Decide how frequently to capture activity events (such as once per day).
- ✓ **Decide how often to capture snapshots**: Decide what interval to capture snapshot data, such as a tenant inventory or the users and groups data.

Type of auditing solution

Tenant-level auditing is either done manually or with automated processes.

Most new auditing processes start off as a manual process, particularly during development and while testing occurs. A manual auditing process can evolve to become an automated process. However, not every auditing process needs to be fully automated.

Manual auditing processes

Manual auditing relies on scripts and processes that are run on-demand by a user (usually a Power BI administrator). When needed, the user runs queries *interactively* to retrieve auditing data.

Manual auditing is best suited to:

- New scripts that are being developed and tested.
- Occasional, one-off auditing needs.
- Exploratory auditing needs.
- Nonessential auditing processes that don't require full support.

Automated auditing processes

Auditing data that's needed on a recurring basis should be automated. It's extracted and processed on a regular schedule, and it's known as an *automated process*. Sometimes it's referred to as an *unattended process*.

The goals of an automated process are to reduce manual effort, reduce risk of error, increase consistency, and eliminate the dependency on an individual user to run it.

Automated auditing is best suited to:

- Auditing processes that run in production.
- Unattended processes that run on a regular schedule.
- Scripts that have been fully tested.
- Essential auditing processes that have other reports (or other processes) that depend on it as a data source.

- Auditing processes that are documented and supported.

The type of process—manual or automated—might impact how you handle authentication. For example, a Power BI administrator might use user authentication for a manual auditing process but use a service principal for an automated process. For more information, see Determine the authentication method later in this article.

> **Tip**
>
> If there's a regular, recurring, need to obtain auditing data that's currently handled manually, consider investing time to automate it. For example, if a Power BI administrator manually runs a script every day to retrieve data from the Power BI activity log, there's a risk of missing data should that person be ill or away on vacation.

Checklist - When considering the type of auditing solution to develop, key decisions and actions include:

- ✓ **Determine the primary purpose for each new auditing requirement**: Decide whether to use a manual or automated process for each new requirement. Consider whether that decision is temporary or permanent.
- ✓ **Create a plan for how to automate**: When it's appropriate for a particular auditing requirement, create a plan for how to automate it, when, and by whom.

Permissions and responsibilities

At this point, you should have a clear idea of what you want to accomplish and the data you'll initially need. Now's a good time to make decisions about user permissions as well as roles and responsibilities.

User permissions

As you plan to build an end-to-end auditing solution, consider user permissions for other users who will need to access the data. Specifically, decide who will be permitted to access and view auditing data.

Here are some considerations to take into account.

Direct access to auditing data

You should decide who can access the auditing data directly. There are multiple ways to think about it.

- **Who should be a Power BI administrator?** A Power BI administrator has access to all tenant metadata, including the admin APIs. For more information about deciding who should be a Power BI administrator, see Tenant-level security planning.
- **Who can use an existing service principal?** To use a service principal (instead of user permissions) to accessing auditing data, a secret must be provided to authorized users so they can perform password-based authentication. Access to secrets (and keys) should be very tightly controlled.
- **Does access need to be tightly controlled?** Certain industries with regulatory and compliance requirements must control access more tightly than other industries.
- **Are there large activity data volumes?** To avoid reaching API throttling limits, you might need to tightly control who's allowed to directly access the APIs that produce auditing data. In this case, it's helpful to ensure that there aren't duplicate or overlapping efforts.

Access to extracted raw data

You should decide who can view the raw data that's extracted and written to a storage location. Most commonly, access to raw data is restricted to administrators and auditors. The Center of Excellence (COE) might require access as well. For more information, see Determine where to store audit data later in this article.

Access to curated analytical data

You should decide who can view the curated and transformed data that's ready for analytics. This data should always be made available to administrators and auditors. Sometimes a data model is made available to other users in the organization, particularly when you create a Power BI semantic model with row-level security. For more information, see Plan to create curated data later in this article.

Roles and responsibilities

Once you've decided how user permissions work, you're in a good position to start thinking about roles and responsibilities. We recommend that you involve the right people early on, especially when multiple developers or teams will be involved in building the end-to-end auditing solution.

Decide which users or team will be responsible for the following activities.

Expand table

Role	Types of responsibilities	Role typically performed by
Communicate to stakeholders	Communication activities and requirements gathering.	COE in partnership with IT. Might also include select people from key business units.
Decide priorities, and provide direction on requirements	Decision-making activities related to the end-to-end auditing solution, including how to meet requirements.	The team that oversees Power BI in the organization, such as the COE. The executive sponsor or a data governance steering committee could become involved to make critical decisions or escalate issues.
Extract and store the raw data	Creation of scripts and processes to access and extract the data. Also involves writing the raw data to storage.	Data engineering staff, usually in IT and in partnership with the COE.
Create the curated data	Data cleansing, transformation, and the creation of the curated data in star schema design.	Data engineering staff, usually in IT and in partnership with the COE.
Create a data model	Creation of an analytical data model, based on the curated data.	Power BI content creator(s), usually in IT or the COE.
Create analytics reports	Creation of reports to analyze the curated data. Ongoing changes to reports based on new requirements and when new auditing data becomes available.	Power BI report creator(s), usually in IT or the COE.
Analyze the data and act on the results	Analyze the data and act in response to the audit data.	The team that oversees Power BI in the organization, usually the COE. Might also include other teams depending on the audit results and the action. Other teams can include security, compliance, legal, risk management, or change management.
Test and validate	Test to ensure that auditing requirements are met and that the data presented is accurate.	Power BI content creator(s), in partnership with the team that oversees Power BI in the organization.
Secure the data	Set and manage security for each storage layer, including the raw data and the curated data.	System administrator for the system that stores the data, in partnership with the

Role	Types of responsibilities	Role typically performed by
	Manage access to semantic models that are created for analyzing the data.	team that oversees Power BI in the organization.
Scheduling and automation	Operationalize a solution and schedule the process with the tool of choice.	Data engineering staff, usually in IT and in partnership with the COE.
Support the solution	Monitor the audit solution, including job runs, errors, and support for technical issues.	The team that handles Power BI system support, which is usually IT or the COE.

Many of the above roles and responsibilities can be consolidated if they're going to be performed by the same team or the same person. For example, your Power BI administrators might perform some of these roles.

It's also possible that different people perform different roles, depending on the circumstance. Actions will be contextual depending on the audit data and the proposed action.

Consider several examples.

- **Example 1**: The audit data shows that some users frequently export data to Excel. *Action taken:* The COE contacts the specific users to understand their needs and to teach them better alternatives.
- **Example 2**: The audit data shows external user access occurs in an unexpected way. *Actions taken:* An IT system administrator updates the relevant Microsoft Entra ID settings for external user access. The Power BI administrator updates the tenant setting related to external user access. The COE updates documentation and training for content creators who manage security. For more information, see Strategy for external users.
- **Example 3**: The audit data shows that several data models define the same measure differently (available from the metadata scanning APIs, if allowed by the detailed metadata tenant settings). *Action taken:* The data governance committee starts a project to define common definitions. The COE updates documentation and training for content creators who create data models. The COE also works with content creators to update their models, as appropriate. For more information about retrieving detailed metadata, see Access tenant inventory data later in this article.

> **Note**

The setup of data extraction processes is usually a one-time effort with occasional enhancements and troubleshooting. Analyzing and acting on the data is an ongoing effort that requires continual effort on a recurring basis.

Checklist - When considering permissions and responsibilities, key decisions and actions include:

- ✓ **Identify which teams are involved**: Determine which teams will be involved with the end-to-end creation and support of the auditing solution.
- ✓ **Decide user permissions**: Determine how user permissions will be set up for accessing audit data. Create documentation of key decisions for later reference.
- ✓ **Decide roles and responsibilities**: Ensure that expectations are clear for who does what, particularly when multiple teams are involved. Create documentation for roles and responsibilities, which includes expected actions.
- ✓ **Assign roles and responsibilities**: Assign specific roles and responsibilities to specific people or teams. Update individual job descriptions with Human Resources, when appropriate.
- ✓ **Create a training plan**: Establish a plan for training personnel when they require new skills.
- ✓ **Create a cross-training plan**: Ensure that cross-training and backups are in place for key roles.

Technical decisions

When you plan for a tenant-level auditing solution, you'll need to make some important technical decisions. To improve consistency, avoid rework, and improve security, we recommend that you make these decisions as early as possible.

The first planning phase involves making the following decisions.

- Select a technology to access audit data
- Determine the authentication method
- Choose a user API or admin API
- Choose APIs or PowerShell cmdlets
- Determine where to store audit data
- Plan to create curated data

Select a technology to access audit data

The first thing you need to decide is *how* to access the audit data.

The easiest way to get started is to use the pre-built reports available in the admin monitoring workspace.

When you need to access the data directly and build your own solution, you'll use an API (application program interface). APIs are designed to exchange data over the internet. The Power BI REST APIs support requests for data by using the HTTP protocol. Any language or tool can invoke Power BI REST APIs when it's capable of sending an HTTP request and receiving a JSON response.

> **Tip**
>
> The PowerShell cmdlets use the Power BI REST APIs to access the audit data. For more information, see **Choose APIs or PowerShell cmdlets** later in this article.

This section focuses on your technology choice. For considerations about the source for accessing *specific types* of audit data, see Data source decisions later in this article.

Platform options

There are many different ways to access audit data. Depending on the technology you choose, you might lean toward a preferred language. You might also need to make a specific decision on how to schedule the running of your scripts. Technologies differ widely in their learning curve and ease of getting started.

Here are some technologies you can use to retrieve data by using the Power BI REST APIs.

Expand table

Technology	Good choice for manual auditing processes	Good choice for automated auditing processes
Admin monitoring workspace	✅	
Try-it	✅	
PowerShell	✅	✅
Power BI Desktop	✅	

Technology	Good choice for manual auditing processes	Good choice for automated auditing processes
Power Automate		✅
Preferred Azure service		✅
Preferred tool/language	✅	✅
Microsoft Purview audit log search	✅	
Defender for Cloud Apps	✅	
Microsoft Sentinel		✅

The remainder of this section provides a brief introduction to each of the options presented in the table.

Admin monitoring workspace

The admin monitoring workspace contains pre-defined reports and semantic models in the Power BI service. It's a convenient way for Fabric administrators to view recent audit data and help them understand user activities.

Try-it in API documentation

Try-it is an interactive tool that allows you to experience the Power BI REST API directly in Microsoft documentation. It's an easy way to explore the APIs because it doesn't require other tools or any setup on your machine.

You can use Try-it to:

- Manually send a request to an API to determine whether it returns the information that you want.
- Learn how the URL is constructed before you write a script.
- Check data in an informal way.

> **Note**
>
> You must be a licensed and authenticated Power BI user to use Try-it. It follows the standard permissions model, meaning that the user APIs rely on user permissions, and the admin APIs require Power BI administrator permissions. You can't authenticate with Try-it by using a service principal.

Because it's interactive, Try-it is best suited to learning, exploration, and new manual auditing processes.

PowerShell and Azure Cloud Shell

You can create and run PowerShell scripts in multiple ways.

Here are several common options.

- **Visual Studio Code**: A modern, lightweight source code editor. It's a freely available open-source tool that's supported on multiple platforms, including Windows, macOS, and Linux. You can use Visual Studio Code with many languages, including PowerShell (by using the PowerShell extension).
- **Azure Data Studio**: A tool for creating scripts and notebooks. It's built on top of Visual Studio Code. Azure Data Studio is available independently, or with SQL Server Management Studio (SSMS). There are many extensions, including an extension for PowerShell.
- **Azure Cloud Shell**: An alternative to working with PowerShell locally. You can access Azure Cloud Shell from a browser.
- **Azure Functions**: An alternative to working with PowerShell locally. Azure Functions is an Azure service that lets you write and run code in a serverless environment. PowerShell is one of several languages that it supports.

> **Important**
>
> We recommend that you use the **latest version** of PowerShell (PowerShell Core) for all new development work. You should discontinue using Windows PowerShell (which can't run PowerShell Core) and instead use one of the modern code editors that support PowerShell Core. Take care when referring to many online examples that use Windows PowerShell (version 5.1) because they might not use the latest (and better) code approaches.

You can use PowerShell in several different ways. For more information about this technical decision, see Choose APIs or PowerShell cmdlets later in this article.

There are many online resources available for using PowerShell, and it's often possible to find experienced developers who can create and manage PowerShell solutions. PowerShell is a good choice for creating both manual and automated auditing processes.

Power BI Desktop

Because Power BI Desktop can connect to APIs, you might be tempted to use it to build your auditing solution. However, there are some significant drawbacks and complexities.

- **Accumulating history**: Because the Power BI activity log provides up to 28 days of data, creating your entire auditing solution involves accumulating a set of files over time that store all historical events. Accumulating historical files is simpler to accomplish with other tools.
- **Handling credentials and keys securely**: Many solutions you find online from community bloggers aren't secure because they hard-code credentials and keys in plaintext within Power Query queries. While that approach is easy, it's not recommended because anyone who obtains your Power BI Desktop file can read the values. You can't store the password (when using user authentication) or the secret (when using a service principal) securely in Power BI Desktop unless you:
  - Use a custom connector that was developed with the Power Query SDK. For example, a custom connector could read confidential values from Azure Key Vault or another service that securely stores the encrypted value. There are also various custom connector options available from the global Power BI community.
  - Use the ApiKeyName option, which works well in Power BI Desktop. However, it isn't possible to store the key value in the Power BI service.
- **Types of requests**: You might run into some limitations for what you can run in Power BI Desktop. For example, Power Query doesn't support every type of API request. For example, only GET and POST requests are supported when using the Web.Contents function. For auditing, you typically send GET requests.
- **Ability to refresh**: You need to follow specific Power Query development patterns to successfully refresh a semantic model in the Power BI service. For example, the `RelativePath` option must be present when using Web.Contents so that Power BI can properly verify the location of a data source without generating an error in the Power BI service.

In most cases, we recommend that you use Power BI Desktop only for two purposes. You should use it to:

- Build your data model based on the existing curated data (rather than using it to initially extract the auditing data).
- Create analytical reports based on your data model.

Power Automate

You can use Power Automate with Power BI in many ways. In relation to auditing, you can use Power Automate to send requests to the Power BI REST APIs and then store the extracted data in the location of your choice.

> **Tip**
>
> To send requests to the Power BI REST APIs, you can use a **custom connector** for Power Automate to securely authenticate and extract the audit data. Alternatively, you can use Azure Key Vault to reference a password or secret. Both options are better than storing passwords and secrets in plaintext within the Power Automate flow.

For other ideas on ways to use Power Automate, search for *Power BI* in the Power Automate templates.

Preferred Azure service

There are several Azure services that can send requests to the Power BI REST APIs. You can use your preferred Azure service, such as:

- Azure Functions
- Azure Automation
- Azure Data Factory
- Azure Synapse Analytics

In most cases, you can combine a compute service that handles the logic for the data extraction with a storage service that stores the raw data (and curated data, when appropriate). Considerations for making technical architecture decisions are described later in this article.

Preferred tool and/or language

You can use your preferred tool and your preferred language to send requests to the Power BI REST APIs. It's a good approach when your team has expertise with a specific tool or language, such as:

- Python
- C# on the .NET framework. Optionally, you can use the Power BI .NET SDK, which acts as a wrapper on top of the Power BI REST APIs in order to simplify some processes and increase productivity.
- JavaScript

Microsoft Purview audit search

The Microsoft Purview compliance portal (formerly the Microsoft 365 compliance center) includes a user interface that allows you to view and search the audit logs. The unified audit logs include Power BI, and all other services that support Microsoft 365 unified audit logs.

The data captured in the unified audit log is the exact same data that's contained in the Power BI activity log. When you do an audit log search in the Microsoft Purview compliance portal, it adds your request to the queue. It can take a few minutes (or longer, depending on the volume of data) for the data to be ready.

Here are some common ways to use the audit log search tool. You can:

- Select the Power BI workload to view events for a series of dates.
- Look for one or more specific activities, such as:
  - *Deleted Power BI report*
  - *Added admin access to a personal workspace in Power BI*
- View activities of one or more users.

For administrators with sufficient permissions, the audit log search tool in the Microsoft Purview compliance portal is a good option for manual auditing processes. For more information, see Microsoft Purview compliance portal later in this article.

Defender for Cloud Apps user interface

Defender for Cloud Apps is available to administrators in Microsoft Defender XDR (Microsoft 365 portal). It includes a user interface to view and search activity logs for various cloud services, including Power BI. It displays the same log events that originate in the Microsoft Purview compliance portal, in addition to other events like user sign-in activity from Microsoft Entra ID.

The audit log interface in Defender for Cloud Apps is a good option for manual auditing processes. For more information, see Defender for Cloud Apps later in this article.

Microsoft Sentinel and KQL

Microsoft Sentinel is an Azure service that allows you to collect, detect, investigate, and respond to security incidents and threats. Power BI can be set up in Microsoft Sentinel as a data connector so that audit logs are streamed from Power BI into Microsoft Sentinel Azure Log Analytics (which is a component of the Azure Monitor service). You can use the Kusto Query Language (KQL) to analyze the activity log events that are stored in Azure Log Analytics.

Using KQL to search the data in Azure Monitor is a good option for viewing part of the audit log. It's a good option for manual auditing processes, too. For more information, see Microsoft Sentinel later in this article.

Platform considerations

Here are some considerations for how you might access audit data.

- **Are you implementing a manual or automated auditing process?** Certain tools align strongly with manual processing or automated processing. For example, a user always runs the Try-it functionality interactively, so it's suited only to manual auditing processes.
- **How will you authenticate?** Not all options support every authentication option. For example, the Try-it functionality only supports user authentication.
- **How will you store credentials securely?** Different technologies have different options for storing credentials. For more information, see Determine the authentication method later in this article.
- **Which technology is your team already proficient with?** If there's a tool that your team prefers and is comfortable using, start there.
- **What is your team capable of supporting?** If a different team will support the deployed solution, consider whether that team is able to adequately support it. Also consider that your internal teams might support development that's done by consultants.
- **What tool do you have approval to use?** Certain tools and technologies might require approval. It could be due to the cost. It could also be due to IT security policies.
- **What's your preference for scheduling?** Some technologies make the decision for you. Other times it will be another decision you must make. For example, if you choose to write PowerShell scripts, there are various ways you can schedule running them. Conversely, if you use a cloud service such as Azure Data Factory, scheduling is built in.

We recommend that you review the remainder of this article before choosing a technology to access audit data.

Checklist - When deciding which technology to use to access audit data, key decisions and actions include:

- ✓ **Discuss with your IT staff**: Talk to your IT teams to find out whether there are certain tools that are approved or preferred.
- ✓ **Explore options with a small proof of concept (POC)**: Do some experimentation with a technical POC. Focus initially on learning. Then focus on your decision for what to use going forward.

Determine the authentication method

How you plan to set up authentication is a critical decision. It's often one of the most difficult aspects when you get started with auditing and monitoring. You should carefully consider available options to make an informed decision.

> ⓘ **Important**
>
> Consult with your IT and security teams on this matter. Take the time to learn the basics of how security in Microsoft Entra ID works.

Not every API on the internet requires authentication. However, all the Power BI REST APIs require authentication. When you access tenant-level auditing data, the authentication process is managed by the Microsoft identity platform. It's an evolution of the Microsoft Entra identity service that's built on industry standard protocols.

> 💡 **Tip**
>
> The Microsoft identity platform **glossary of terms** is a resource that helps you become familiar with the basic concepts.

Before you retrieve audit data, you must first *authenticate*, which is known as *signing in*. The concepts of authentication and authorization are separate, yet related. An *authentication* process verifies the identity of *who* is making the request. An *authorization* process grants (or denies) access to a system or resource by verifying *what* the user or service principal has permission to do.

When a user or service principal authenticates, a Microsoft Entra access token is issued to a resource server, such as a Power BI REST API or a Microsoft Graph API. By default, an access token expires after one hour. A refresh token can be requested, if needed.

There are two types of access tokens:

- **User tokens**: Issued on behalf of a user with a known identity.

- **App-only tokens**: Issued on behalf of a Microsoft Entra application (service principal).

For more information, see Application and service principal objects in Microsoft Entra ID.

> **Note**
>
> A Microsoft Entra application is an identity configuration that allows an automated process to integrate with Microsoft Entra ID. In this article, it's referred to as an *app registration*. The term *service principal* is also used commonly in this article. These terms are described in more detail later in this section.

> **Tip**
>
> The easiest way to authenticate is to use the Connect-PowerBIServiceAccount cmdlet from the Power BI Management module. You might see other authentication-related cmdlets in articles and blog posts online. For example, there are the `Login-PowerBI` and `Login-PowerBIServiceAccount` cmdlets, which are aliases for `Connect-PowerBIServiceAccount` cmdlet. There's also the Disconnect-PowerBIServiceAccount cmdlet that you can use to explicitly sign out at the end of a process.

The following table describes the two authentication options.

Expand table

Type of authentication	Description	Intended for	Good choice for manual auditing processes	Good choice for automated auditing processes
User authentication	Sign in as a user by using the user principal name (email address) and a password.	Occasional, interactive use	✔	
Service principal authentication	Sign in as a service principal by using a secret (key) assigned to an app registration.	Ongoing, scheduled, unattended operations		✔

Each authentication option is described in more detail in the following section.

User authentication

User authentication is useful in the following situations.

- **Manual auditing processes**: You want to run a script by using your user permissions. Permissions could be at one of two levels, depending on the type of API request:
  - **Administrator permissions for admin APIs**: You need to use your Power BI administrator permissions to send a request to an *admin API*.
  - **User permissions for user APIs**: You need to use your Power BI user permissions to send a request to a *user API*. For more information, see Choose a user API or admin API.
- **Interactive sign in**: You want to sign in interactively, which requires you to input your email address and password. For example, you must sign in interactively to use the Try-it experience, which is found in Microsoft API documentation.
- **Track who accessed tenant metadata**: When individual users and administrators send API requests, you might want to audit who those individuals are. When you authenticate with a service principal (described next), the user ID captured by the activity log is the Application ID. If multiple administrators authenticate with the same service principal, it might be difficult to know which administrator accessed the data.
- **Shared administrator account**: Some IT teams use a shared administrator account. Depending on how it's implemented and controlled, it might not be a best practice. Instead of a shared account, you should consider using Microsoft Entra Privileged Identity Management (PIM), which can grant occasional and temporary Power BI administrator rights for a user.
- **APIs that only support user authentication**: Occasionally, you might need to use an API that doesn't support authentication by a service principal. In documentation, each API notes whether a service principal can call it.

> **Important**
>
> Whenever possible, we recommend that you use service principal authentication for automated processes.

Service principal authentication

Most organizations have one Microsoft Entra tenant, so the terms *app registration* and *service principal* tend to be used interchangeably. When you register a Microsoft Entra app, there are two components.

- **App registration**: An app registration is globally unique. It's the global definition of a registered Microsoft Entra app that can be used by multiple tenants. An app registration is also known as a *client application* or the *actor*. Typically, the term *client application* implies a user machine. However, that's not the situation for app registrations. In the Azure portal, Microsoft Entra applications are found on the *App registrations* page in Microsoft Entra ID.
- **Service principal**: A service principal is the local representation of the app registration for use in your specific tenant. The service principal is derived from the registered Microsoft Entra app. For organizations with multiple Microsoft Entra tenants, the same app registration can be referenced by more than one service principal. In the Azure portal, service principals can be found on the *Enterprise applications* page in Microsoft Entra ID.

Because the service principal is the local representation, the term *service principal authentication* is the most common way to refer to sign-ins.

> **Tip**
>
> Your Microsoft Entra administrator can tell you who's allowed to **create**, and consent to, an app registration in your organization.

Service principal authentication is useful in the following situations.

- **Automated auditing processes**: You want to run an unattended process on a schedule.
- **No need to sign in to the Power BI service**: You only need to connect and extract data. A service principal can't sign in to the Power BI service.
- **Shared access to data**: You (personally) aren't a Power BI administrator, but you're authorized to use a service principal. The service principal has permission to run admin APIs to retrieve tenant-level data (or other specific permissions).
- **Use by multiple administrators**: You want to allow multiple administrators or users to use the same service principal.
- **Technical blockers**: There's a technical blocker that prevents the use of user authentication. Common technical blockers include:
  - **Multi-factor authentication (MFA)**: Automated auditing processes (that run unattended) can't authenticate by using a user account when multi-factor authentication is enabled.
  - **Password hash synchronization**: Microsoft Entra ID requires password hash synchronization for a user account to authenticate. Sometimes, IT or a cybersecurity team might disable this functionality.

App registration purpose and naming convention

Each app registration should have a clear name that describes its purpose and the target audience (who can use the app registration).

Consider implementing a naming convention such as: *<Workload> - <Purpose> - <Target audience> - <Object type>*

Here are the parts of the naming convention.

- **Workload**: Usually equivalent to a data source (for example, Power BI data or Microsoft Graph data).
- **Purpose**: Similar to the level of permissions (for example, Read versus ReadWrite). As described below, the purpose helps you to follow the principle of least privilege when you create separate app registrations that can only read data.
- **Target audience**: Optional. Depending on the workload and purpose, the target audience allows you to determine the intended users of the app registration.
- **Object type**: *EntraApp* is included for clarity.

Your naming convention can be more specific when you have multiple tenants or multiple environments (such as development and production).

The following table shows examples of app registrations that could be used to extract tenant-level auditing data:

Expand table

App registration name	Purpose	Target audience
PowerBIData-Read-AdminAPIs-EntraApp	Extract tenant-wide metadata for auditing and administration of Power BI. Admin APIs are always read-only and might not have permissions granted in Microsoft Entra ID (Power BI tenant setting only).	Power BI administrators and the Center of Excellence
PowerBIData-ReadWrite-PowerBIAdministrators-EntraApp	Manage the Power BI tenant. Requires read/write permissions to create or update resources.	Power BI administrators and the Center of Excellence
GraphData-Read-PowerBIAdministrators-EntraApp	Extract users and groups data for auditing and administration of Power BI. Requires read permissions.	Power BI administrators and the Center of Excellence

Managing multiple app registrations for different purposes involves more effort. However, you can benefit from several advantages.

- **See what the app registration is *intended* to be used for and why**: When you see the client ID from the app registration show up in your auditing data, you'll have an idea of what it was used for and why. That's a significant advantage of creating separate app registrations (rather than using only one for all purposes).
- **See who the app registration is *intended* to be used by**: When you see the client ID from the app registration show up in your auditing data, you'll have an idea of who was using it.
- **Avoid overprovisioning permissions**: You can follow the principle of least privilege by providing separate app registrations to different sets of people who have different needs. You can avoid overprovisioning permissions by using a read-only app registration when write permissions aren't necessary. For example, you might have some highly capable self-service users who want to gather data about their semantic models; they need access to a service principal with *read* permission. Whereas there might be satellite members of the Center of Excellence working for the Finance team who programmatically manage semantic models; they need access to a service principal with *write* permission.
- **Know who *should* be in possession of the secret**: The secret for each app registration should only be shared with the people who need it. When the secret is *rotated* (described later in this section), the impact is smaller when you use separate app registrations for different needs. That's helpful when you need to rotate the secret because a user leaves the organization or changes roles.

App registration permissions

There are two main ways that an app registration can access data and resources. It involves permissions and consent.

- **App-only permissions**: Access and authorization are handled by the service principal without a signed-in user. That method of authentication is helpful for automation scenarios. When using app-only permissions, it's critical to understand that permissions are *not* assigned in Microsoft Entra ID. Rather, permissions are assigned in one of two ways:
  - **For running admin APIs**: Certain Fabric tenant settings allow access to the endpoints for the admin APIs (that return tenant-wide auditing metadata). For more information, see Set Fabric tenant settings later in this article.
  - **For running user APIs**: Power BI workspace and item permissions apply. Standard Power BI permissions control what data can be returned to a service principal when running user APIs (that return auditing data based on permissions of the user or service principal that's invoking the API).
- **Delegated permissions**: Scope-based permissions are used. The service principal accesses data on behalf of the user that's currently signed in. It means that the

service principal can't access anything beyond what the signed-in user can access. Be aware that this it's a different concept from user-only authentication, described previously. Because a custom application is required to properly handle the combination of user and app permissions, delegated permissions are rarely used for auditing scenarios. This concept is commonly misunderstood, leading to many app registrations that are overprovisioned with permissions.

> **Important**
>
> In this article, the focus is only on user authentication or app-only authentication. Delegated permissions (with an interactive user and the service principal) could play an important role when programmatically embedding Power BI content. However, we strongly discourage you from setting up delegated permissions for extracting auditing data. Every API limits how frequently it can be run (with throttling in place), so it isn't practical to have different users directly accessing the raw audit data. Instead, we recommend that you extract the raw audit data once (with a centralized process), and then make the curated audit data available to authorized users downstream.

For more information, see Register a Microsoft Entra app later in this article.

App registration secrets

An app registration often has a *secret* assigned to it. The secret is used as a key for authentication, and it has an expiration date. Therefore, you need to plan how to rotate the key regularly and how to communicate the new key to authorized users.

You also need to concern yourself with where to securely store the secret. An organizational password vault, such as Azure Key Vault, is an excellent choice.

> **Tip**
>
> As an alternative approach to using a secret, you can use a ***managed identity***. A managed identity eliminates the need to manage credentials. If you intend to use services like Azure Functions or Azure Data Factory to extract the data, a managed identity is a good option to investigate.

Securely manage credentials

Regardless of whether you use user authentication or service principal authentication, one of the biggest challenges is how to securely manage passwords, secrets, and keys.

> **⊗ Caution**
>
> The first rule is one that many people violate: passwords and keys should never appear in plaintext in a script. Many articles, blogs, and examples online do that for simplicity. However, it's a poor practice that should be avoided. Anyone that obtains the script (or the file) can potentially gain access to the same data that the author has access to.

Here are several secure methods you can use to manage passwords, secrets, and keys.

- Integrate with Azure Key Vault or another organizational password vault system, whenever possible.
- Use credentials and secure strings in your PowerShell scripts. A credential works for both user authentication and service principal authentication. However, you can't use a credential when multi-factor authentication is enabled for a user account.
- Prompt for a password in your PowerShell script, or use interactive authentication with a browser.
- Use the Secret Management module for PowerShell that's published by Microsoft. It can store values by using a local vault. It can also integrate with a remote Azure Key Vault, which is useful when there are multiple administrators in your organization who work with the same scripts.
- Create a custom connector for Power BI Desktop so that it can securely handle credentials. Some custom connectors are available from community members (usually on GitHub).

> **Tip**
>
> We recommend that you consult with your IT and security teams to help choose the best method, or validate that your solution manages credentials in a secure way.

Microsoft Authentication Library

Support for Azure AD Authentication Library (ADAL) ended in December 2022. Going forward, you should use Microsoft Authentication Library (MSAL). The security team in your organization should be familiar with this significant change.

While it's not important for Power BI professionals to deeply understand the transition to MSAL, you should adhere to the following recommendations.

- Use the latest version of the Power BI Management module when you authenticate with the Connect-PowerBIServiceAccount PowerShell cmdlet. It switched from ADAL to MSAL in version 1.0.946 (February 26, 2021).
- Use the Microsoft Entra V2 endpoint when you authenticate directly in your script. The Microsoft Entra V2 endpoint uses MSAL, whereas the Microsoft Entra V1 endpoint uses ADAL.
- Discontinue the use of APIs and modules that are deprecated. For more information, see Deprecated APIs and modules later in this article.
- If you find a community solution online, be sure that it's using MSAL for authentication rather than ADAL.

Checklist – When deciding how to manage authentication, key decisions and actions include:

- ✓ **Decide which type of authentication to use**: Determine whether user authentication or service principal authentication is most suitable, depending on the type of auditing solution you plan to create.
- ✓ **Plan for what app registrations you need**: Consider your requirements, workloads, and target audience (users of each app registration). Plan for how many app registrations you need to create to support these needs.
- ✓ **Create a naming convention for app registrations**: Set up a naming convention that makes it easy to understand the intended purpose and intended users of each app registration.
- ✓ **Plan for how to securely manage credentials**: Consider ways to securely manage passwords, secrets, and keys. Consider what documentation and training might be necessary.
- ✓ **Confirm the use of MSAL**: Determine whether there are any existing manual or automated auditing solutions that rely on ADAL. If necessary, create a plan to rewrite these solutions to use the newer MSAL authentication library.

Choose a user API or admin API

When planning to retrieve auditing data, it's important to use the right type of API.

There are two types of APIs to consider.

- **User APIs**: Rely on the permissions of the signed-in user (or service principal) to send requests to the API. For example, one way to return a list of semantic models in a workspace is with a user API. Permission to read semantic models can be granted either by workspace role or per-item permissions. There are two ways to run user APIs:
  - **User authentication**: The signed-in user must have permission to access the workspace or item.
  - **Service principal authentication**: The service principal must have permission to access the workspace or item.
- **Admin APIs**: Retrieve metadata from the tenant. It's sometimes referred to as the *organizational scope*. For example, to return a list of all semantic models in the tenant, you use an admin API. There are two ways to run admin APIs:
  - **User authentication**: The signed-in user must be a Power BI administrator to run admin APIs.
  - **Service principal authentication**: The service principal must have permission to run admin APIs (without relying on security permissions like a user API does).

> **Tip**
>
> All admin APIs belong to the **Admin operation group**. Any API that has the *As Admin* suffix is an admin API; all remaining APIs are user APIs.

Consider an example where you need to obtain a list of semantic models. The following table shows six API options that you can use to do that. Four options are user APIs, and two options are admin APIs. The scope and number of items that are returned are different, depending on the API you choose.

Expand table

API name	Type of API	Scope	Number of semantic models
Get Dataset	User	Personal workspace	One
Get Datasets	User	Personal workspace	All
Get Dataset in Group	User	One workspace	One, provided the signed-in user has permission to read the semantic model
Get Datasets in Group	User	One workspace	All, provided the signed-in user has permission to read each semantic model

API name	Type of API	Scope	Number of semantic models
Get Datasets in Group as Admin	Admin	One workspace	All
Get Datasets as Admin	Admin	Entire tenant	All

> **Note**
>
> Some of the API names include the term *Group* as a reference to a workspace. That term originated from the original Power BI security model, which relied on Microsoft 365 groups. While the Power BI security model has evolved significantly over the years, the existing API names remain unchanged to avoid breaking existing solutions.

For information about tenant settings, see Set Fabric tenant settings later in this article.

Checklist - When choosing whether to use a user API or an admin API, key decisions and actions include:

- ✓ **Refer to data requirements**: Collect and document the key business requirements for an auditing solution. Based on the type of data that's needed, determine whether a user scope or organizational scope is appropriate.
- ✓ **Test the results**: Do some experimentation. Test the accuracy of the results that are returned by the different APIs.
- ✓ **Review permissions**: For any existing auditing solutions, confirm that permissions are assigned correctly for Power BI administrators and service principals. For new auditing solutions, confirm which method of authentication will be used.

Choose APIs or PowerShell cmdlets

A key decision to make is whether to use PowerShell cmdlets when they're available for the specific data that you want to retrieve. This decision is important if you've decided that PowerShell is one of the technologies you'll use to access audit data.

PowerShell is a task automation solution. The term *PowerShell* is a collective term that refers to the scripting language, a command-line shell application, and a configuration management framework. PowerShell Core is the newest version of PowerShell, which runs on Windows, Linux, and macOS.

A PowerShell *cmdlet* is a command that performs an action. A PowerShell *module* is a package that contains PowerShell members, such as cmdlets, providers, functions, workflows, variables, and aliases. You download and install modules.

A PowerShell module creates a layer of abstraction over the APIs. For example, the Get-PowerBIActivityEvent cmdlet retrieves (or *gets*) audit events for a Power BI tenant. This PowerShell cmdlet retrieves its data from the Get Activity Events REST API. Generally, it's simpler to get started by using the cmdlets because it simplifies access to the underlying APIs.

Only a subset of the APIs are available as PowerShell cmdlets. As you continue to expand your entire auditing solution, it's likely that you'll use a combination of cmdlets and APIs. The remainder of this section helps you to decide which way you'll access the data.

PowerShell modules

Microsoft has published two PowerShell modules that contain cmdlets related to Power BI. They're the Power BI Management module and the Data Gateway module. These PowerShell modules act as an *API wrapper* on top of the underlying Power BI REST APIs. Using these PowerShell modules can make it easier to write scripts.

> **Tip**
>
> In addition to the modules published by Microsoft, there are freely available PowerShell modules and scripts published (usually on GitHub) by Power BI community members, partners, and Microsoft Most Valued Professionals (MVPs).
>
> Starting with a community solution can play an important role in building your end-to-end auditing solution. If you choose to use a freely available solution, be sure to fully test it. Become familiar with how it works so you can effectively manage your solution over time. Your IT department might have criteria for using publicly available community solutions.

Power BI Management module

The Power BI Management module is a rollup module that contains specific Power BI modules for administration, capacities, workspaces, and more. You can choose to install the rollup module to obtain all of the modules, or you can install specific modules. All of the Power BI Management modules are supported on both Windows PowerShell and PowerShell Core.

> **Important**
>
> We recommend that you discontinue using Windows PowerShell (which can't run PowerShell Core). Instead, use one of the modern code editors that supports PowerShell Core. The Windows PowerShell ISE (integrated script environment) is only capable of running PowerShell version 5.1, which is no longer updated. Windows PowerShell is now deprecated, so we recommend that you use PowerShell Core for all new development work.

Here are several commonly used cmdlets that you can use to retrieve auditing data.

Expand table

Management module	Cmdlet	Purpose
Profile	Connect-PowerBIServiceAccount	Authenticate a Power BI user or service principal.
Admin	Get-PowerBIActivityEvent	View or extract Power BI activity log events for the tenant.
Workspaces	Get-PowerBIWorkspace	Compile a list of workspaces.
Reports	Get-PowerBIReport	Compile a list of reports for a workspace or the tenant.
Data	Get-PowerBIDataset	Compile a list of semantic model for a workspace or the tenant.
Profile	Invoke-PowerBIRestMethod	Send a REST API request (command). This cmdlet is a good option because it can invoke any of the Power BI REST APIs. It's also a good choice when you want to use the simpler form of authentication by using the `Connect-PowerBIServiceAccount` cmdlet and be able to invoke an API that doesn't have a corresponding PowerShell cmdlet. For more information, see Considerations for using PowerShell cmdlets later in this section.

> **Tip**
>
> There are other cmdlets available for managing and publishing content. However, the focus of this article is on retrieving auditing data.

You can download the Power BI Management module from the PowerShell Gallery. The simplest way to install modules is to use PowerShellGet.

We recommend that you install the Power BI Management rollup module. That way, all the cmdlets you might need are available. At a minimum, you need the Profile module (for authentication) and any specific modules that contain the cmdlets that you want to use.

> **Caution**
>
> Be sure you keep the modules up to date on every server, user machine, and cloud service (such as Azure Automation) where you use PowerShell. If the modules aren't updated regularly, unpredictable errors or issues could arise. Some tools (like Visual Studio Code) help you keep the modules updated. Also, be aware that PowerShellGet doesn't automatically **uninstall** older versions of a module when you install or update a newer version. It installs newer versions alongside the older versions. We recommend that you **check** for installed versions and periodically uninstall older versions.

Data Gateway module

The Data Gateway module contains cmdlets that retrieve data from an on-premises data gateway and its data sources. The Data Gateway module is supported only on PowerShell Core. It's not supported on Windows PowerShell.

Unlike the Power BI Management module (previously described), the Data Gateway module doesn't include any admin cmdlets. Many of the cmdlets (and the corresponding gateway APIs), require that the user has gateway administrator rights.

> **Warning**
>
> It's not possible to assign a service principal as gateway administrator (even when the service principal is a member of a security group). Therefore, plan to use user credentials when retrieving information about data gateways.

Here are several cmdlets from the Data Gateway module that you can use to retrieve auditing data.

Expand table

Cmdlet	Purpose
Connect-DataGatewayServiceAccount	Authenticate a Power BI user (usually requires that the user belongs to the gateway admin role).
Get-DataGatewayCluster	Compile a list of gateway clusters.
Get-DataGatewayClusterDataSource	Compile a list of data sources for a gateway cluster.
Get-DataGatewayInstaller	Verify which users are allowed to install and register gateways in the organization.

You can download the Data Gateway module from the PowerShell Gallery.

Techniques to use PowerShell

You can use PowerShell in several different ways. The decision that you make is an important one.

The following table describes three different techniques for using PowerShell.

Expand table

Technique	Description	Example
Use PowerShell cmdlets to simplify authentication, and to call APIs directly. *Recommended approach*	With this technique, there's a balance of simplicity and flexibility. The Invoke-PowerBIRestMethod cmdlet is available from the Power BI Management Profile module. This cmdlet is often referred to as a *Swiss Army Knife* because you can use it to call any of the Power BI REST APIs. The advantage of using this technique is you can simplify authentication, and then call any of the Power BI REST APIs. We strongly recommend that you use this technique.	After authenticating with the Connect-PowerBIServiceAccount cmdlet, use the Invoke-PowerBIRestMethod cmdlet to retrieve data from your preferred API (for example, Get Pipeline Users As Admin).
Use cmdlets from PowerShell as much as possible, both for	With this technique, PowerShell is used extensively. PowerShell is used to coordinate running the script, and PowerShell modules are used whenever	After authenticating with the Connect-PowerBIServiceAccount cmdlet, use a cmdlet (for example, Get-

Technique	Description	Example
authentication and for retrieving data.	possible for accessing the data. This creates a greater dependency on PowerShell from multiple aspects.	PowerBIActivityEvent) to retrieve data.
Use PowerShell only to coordinate running the process. PowerShell modules are used as little as possible.	With this technique, there's less dependency on PowerShell as a tool since its primary use is to orchestrate invoking API calls. Only one generic PowerShell module is used to access APIs (none of the modules from the Power BI Management modules are used).	After authenticating by using the Microsoft Authentication Library (MSAL), call your preferred API (for example, Get Pipeline Users As Admin) by using the generic Invoke-RestMethod cmdlet to retrieve data.

In the above table, the first technique describes an approach that balances ease of use with flexibility. This approach provides the best balance for the needs of most organizations, which is why it's recommended. Some organizations might have existing IT policies and staff preferences that influence how you decide to use PowerShell.

> **Tip**
>
> You can use the **Invoke-ASCmd** PowerShell cmdlet to create and execute **DAX**, **XMLA**, and **TMSL** scripts. However, these use cases are out of scope for this article. For more information about querying Dynamic Management Views (DMVs), see **Data-level auditing**.

Technical users (and administrators) can use the Power BI Management modules to retrieve data or automate certain aspects of content management.

- **For administrators**: Set the `-Scope` parameter to `Organization` to access entities across the entire tenant (for example, to retrieve a list of *all workspaces*).
- **For technical users**: Set the `-Scope` parameter to `Individual` (or omit the parameter) to access entities that belong to the user (for example, to retrieve a list of workspaces *that the user has permission to access*).

Consider a scenario where you need to obtain a list of semantic models. If you choose to work directly with the APIs, you must specify which API to call. However, if you choose to use the Get-PowerBIDataset cmdlet, you can set the `-Scope` parameter to retrieve a user-specific or tenant-wide list of semantic models. The choice you make depends on your decision for how to use PowerShell (covered in the previous table).

The following table describes how the parameters used in the PowerShell cmdlet translate to the API PowerShell calls.

Expand table

Cmdlet parameters	API that the cmdlet uses	Type of API	Scope	Retrieved items
`-DatasetID {DatasetID} -Scope Individual`	Get Dataset	User	Personal workspace	One semantic model
`-Scope Individual`	Get Datasets	User	Personal workspace	All semantic models
`-DatasetID {DatasetID} -GroupID {WorkspaceID}`	Get Dataset in Group	User	One workspace	One semantic model, provided the signed-in user has permission to read the semantic model
`-GroupID {WorkspaceID}`	Get Datasets in Group	User	One workspace	All semantic models, provided the signed-in user has permission to access the workspace and read the semantic model
`-GroupID {WorkspaceID} -Scope Organization`	Get Datasets in Group as Admin	Admin	One workspace	All semantic models
`-Scope Organization`	Get Datasets as Admin	Admin	Entire tenant	All semantic models

Considerations for using PowerShell cmdlets

The Power BI PowerShell cmdlets are simpler to use because they abstract some of the complexity of the REST API calls.

Here are some of the ways that the cmdlets simplify accessing auditing data.

- **Authentication**: The simplest way to authenticate in a PowerShell script is to use the `Connect-PowerBIServiceAccount` cmdlet.
- **Simplicity**: It's simpler to get started because the techniques to retrieve auditing data are straightforward. Consider that when you use the Get-PowerBIActivityEvent cmdlet, you retrieve *one day* of audit data. However, when you directly call the Get Activity Events REST API, you retrieve *one hour* of audit data. So, when you use the REST API to retrieve one day of audit data you must use a loop to call the API 24 times to extract each hour of the day.

- **Pagination of large result sets**: Large result sets are efficiently retrieved by *pagination*. Pagination involves retrieving one chunk of the results at a time. The cmdlets automatically manage pagination for you. However, when you directly use the REST APIs, your script must check a continuation token to determine whether there are any more results to come. The script should continue retrieving chunks of results until no continuation token is received. For an example of using a continuation token, see Activity Events REST API.
- **Access token expirations**: Upon authenticating, an access token is returned. By default, it expires after one hour. The cmdlets handle access token expirations for you. That way, you don't need to write logic to request a refresh token.
- **Formatting**: The data returned by a cmdlet is formatted slightly nicer than the data returned by the REST API. The output is more user-friendly. For automated auditing processes, that's not likely to be an issue. However, for manual auditing processes the nicer formatting might be helpful.

Considerations for using the REST APIs directly

Sometimes there are advantages to calling the Power BI REST APIs directly.

- **Many more APIs available**: There are more REST APIs available than PowerShell cmdlets. However, don't overlook the flexibility of the Invoke-PowerBIRestMethod cmdlet, which you can use to call any of the Power BI REST APIs.
- **Frequency of updates**: Microsoft updates the REST APIs more frequently than it updates the PowerShell modules. For example, if a new attribute is added to the Get Dataset API response, it could take some time before it becomes available in the results of the Get-PowerBIDataset cmdlet.
- **Less language/tool dependency**: When you use the REST APIs directly (instead of a cmdlet), you don't need to use PowerShell. Or, you might choose to use PowerShell only to orchestrate calling many APIs in a script. By removing (or avoiding) any dependency on PowerShell, at some time in the future you can rewrite your logic in a different language. When your IT or developer team has strong preferences for tools and languages, less dependency can be an important consideration to take into account.

Regardless of which method you choose to use, the REST APIs can change over time. When a technology evolves, APIs (and PowerShell modules) can be deprecated and replaced. Therefore, we recommend that you purposefully create scripts that are simple to maintain and resilient to change.

Checklist - When choosing whether to access the REST APIs directly or by using PowerShell cmdlets, key decisions and actions include:

- ✓ **Consult with IT on the use of PowerShell**: Contact the relevant IT team(s) to determine whether there are existing internal requirements or preferences on how PowerShell can be used.
- ✓ **Decide how you want to use PowerShell**: Determine which PowerShell techniques to use, and whether you want to intentionally increase or reduce your dependency on PowerShell. Consider whether internal communication or training is necessary.
- ✓ **Upgrade to PowerShell Core**: Research using PowerShell Core. Determine whether administrator machines need to be updated. Consider which existing scripts need to be tested. Determine whether administrators or developers need additional training to upgrade their skills.
- ✓ **Determine which PowerShell modules to use**: Consider whether the Power BI Management modules and/or the Data Gateway module will be used.
- ✓ **Decide whether community projects are permitted**: Determine whether you're allowed (or even encouraged) to use PowerShell modules available online (versus modules published by Microsoft). Consult with IT to determine whether there are criteria for community projects obtained online.
- ✓ **Clarify how to support community projects**: If PowerShell community projects are permitted, consider who'll be responsible for supporting them once they're used internally.
- ✓ **Complete a small proof of concept (POC)**: Experiment with a technical POC. Confirm your preferred client tools and method of accessing data.
- ✓ **Determine how to support advanced users**: Consider how you're going to support technical users who create automation on their own by using the user APIs.

Determine where to store audit data

When you plan to build an end-to-end auditing solution, you'll need to decide where to store the raw data (and the curated data, which is covered in the next section). Your decisions about data storage are related to your preferences for how to handle *data integration*.

When you extract raw auditing data, keep it simple. We recommend that you don't select specific attributes, perform transformations, or apply any formatting when you

initially extract the data. Instead, extract all available attributes and store the data in its original state.

Here are several reasons why storing the raw data in its original state is a best practice.

- **All data available in history**: New attributes and new event types will become available over time. Storing all the raw data is a good way to ensure that you'll always have access to whatever data was available at the time you extracted it. Even when it takes you time—which could be weeks or months—to realize that new data attributes are available, it's possible to analyze them historically because you captured them in the raw data.
- **Resilient to change**: If the raw data format changes, the process that extracts the data isn't impacted. Because some auditing data is time-sensitive, it's important to make sure that you design a data extraction process that won't fail when a change occurs in the source.
- **Roles and responsibilities**: Different team members (such as data engineers or Fabric administrators) might be responsible for creating the processes to access, extract, and store the raw audit data. Simplifying the data extraction process makes it easier for multiple teams to work together.

Here are some options for storing raw data.

- **Data lake or object storage**: A cloud service such as OneLake that specializes in storing files of any structure. It's an ideal choice for storing the raw audit data. In a data lake architecture, raw data should be stored in the bronze layer.
- **File system**: A file system (such as Windows or Linux) is a common choice for storing the raw audit data.
- **Database**: It's possible to store JSON data in a relational database, like Azure SQL Database. However, it's less common than storing the data in a data lake or file system. That's because querying and maintaining JSON data can become challenging and expensive, especially as data volumes grow.

> **Tip**
>
> When you use a data lake, object storage, or a file system, we recommend that you store the data in a way that's easy to organize and secure. It's also important to be clear about whether the data comprises events (which are typically appended) or whether it's a point-in-time snapshot (which requires selecting a date to analyze).

Consider an example involving a raw data zone of a data lake. The zone has a folder hierarchy for storing activity log data: Raw > ActivityLog > YYYY > MM. The folders are partitioned by year and month. A file stored in the month folder uses the following

format: PBIActivityLog-YYYYMMDD-YYYYMMDDTTTT.json. *YYYYMMDD* represents the actual data, and *YYYYMMDDTTTT* represents the extraction time stamp. By including the extraction time stamp, you can determine which file is the latest (in case you need to extract the data again). For example, if you extract data at 9am (UTC) on April 25, 2023 for activity on April 23, 2023, the file name would be PBIActivityLog-20230523-202305250900.

We strongly recommend that you use a technology that allows you to write the raw data to *immutable storage*. Immutable storage guarantees that once the data is written, it can't be overwritten or deleted. That distinction is important to auditors, and it allows you to trust that the raw data is reliable.

You also need to consider how to securely store the raw data. Typically, very few users require access to the raw data. Access to raw data is typically provided on a needs-basis, typically for data engineers and system administrators. Your internal auditors might also need access. The team members who are responsible for creating the curated data (described next) also require access to the raw data.

Here are some considerations to help you choose your raw data storage.

- **Is it a manual or automated auditing process?** An automated auditing process typically has stricter requirements for how and where data is stored.
- **Is the subject area particularly sensitive?** Depending on the type of data and how sensitive it is, your organization might have a requirement for how and where it's stored. Power BI audit data contains identifying user information and IP addresses, so it should be stored in a secure area.
- **Is there a preferred storage technology?** There might be an existing storage technology that's in use within the organization, so it's a logical choice.
- **Will users access the storage directly?** The security model is an important consideration. Usually, very few users are granted access to raw data. Access to the curated data is typically made available to Power BI content creators who are responsible for creating the centralized data model (the semantic model that serves as the layer for reporting).
- **Do you have data residency requirements?** Some organizations have legal or regulatory data residency requirements to store data in a specific data region.
- **How will the data be organized?** Use of the medallion architecture is a common practice, particularly in data lake and lakehouse implementations. The goal is to store your data in bronze, silver, and gold data layers. The bronze layer contains the original raw data. The silver layer contains validated and deduplicated data used for transformations. The gold layer contains the curated, highly refined data that's ready for analysis.

Checklist - When planning the location to store raw data, key decisions and actions include:

- ✓ **Consult with IT**: Contact the relevant IT team(s) to determine whether there are existing processes running to extract the raw audit data. If so, find out whether you can access the existing data. If a new data extract process is required, determine whether there are preferences or requirements for where the data should be stored.
- ✓ **Decide where to store raw data**: Determine the best storage location and structure for storing the raw data.
- ✓ **Determine data residency requirements**: Find out whether there are legal or regulatory requirements for where the data can be stored.
- ✓ **Create a folder structure and naming convention**: Determine what naming convention to use for raw data files, including the folder structure. Include these details in your technical documentation.
- ✓ **Consider how security for raw data will work**: While you design the raw data storage location, consider who'll need to access the data and how access will be granted.

Plan to create curated data

Curated data supports analysis and is designed to be user-friendly. You need to make some decisions on how and where curated data will be created. The technology you choose to store the curated data might be any of the technologies listed in the previous section.

The goal of curated data is to transform the data into a more friendly format that's optimized for analysis and reporting. It forms the data source of a Power BI data model, so that means you use a Power BI data model to analyze the usage of Power BI in your organization. Fundamental data model guidance applies: You should adopt a star schema design, which is optimized for performance and usability.

You can choose to create curated data in different ways. You need to decide how to do *data integration* and how far *upstream* to apply the transformations that prepare the data for loading into a star schema.

Data lake

You can apply transformations and create curated data files in a data lake. You should use a gold layer for curated data so that it's logically separate from the raw data that's stored in the bronze layer. Separating the data in layers also simplifies setting different user access permissions.

Use a data lake to transform and curate the data when:

- You expect that there'll be multiple Power BI data models that serve reporting (which justifies the creation of an intermediary silver layer).
- You need to support multiple self-service content creators who need access to tenant-level audit data.
- You have existing tools and processes that you want to use to transform and load data.
- You want to minimize the data preparation that needs to be done (and potentially duplicated) within one or more Power BI data models.

Power BI data model

You might be able to complete all the transformation work in Power BI. Use Power Query to access and transform the data from its raw state to the curated format.

Use a Power BI data model when:

- You want to simplify the end-to-end architecture and load the data model directly from the raw data. (Incremental refresh can help reduce refresh durations.)
- Your preference is to do all the transformation work while loading the data model.
- You expect to have one centralized data model for the tenant-level audit data. All reports (or other data models) can use the centralized Power BI semantic model as its source.
- You want to provide user access only to the centralized Power BI semantic model (and not to any of the source data).

> **Tip**
>
> When you create the curated data, set it up in a way so you can easily re-run the process for earlier date ranges. For example, if you discover that a new attribute appeared in the audit logs three months ago, you will want to be able to analyze it since it first appeared. The ability to reload the curated data history is one of several reasons why it's important to store the raw data in its original state (described earlier in this article).

For more information about what dimension tables and fact tables you might include in your star schema, see Create a data model later in this article.

Checklist - When planning how to create curated data, key decisions and actions include:

- ✓ **Decide where transformations should be done**: Consider how far upstream the transformation work should be done, and where that fits into your plan for the entire architecture.
- ✓ **Decide what tool to use to transform the data into a star schema**: Confirm which tools and services will be used for transforming the raw data into curated data.
- ✓ **Decide where curated data should be stored**: Determine the best choice for storing the raw data that's been transformed into a star schema format. Decide whether an intermediary silver layer is useful in the end-to-end architecture.
- ✓ **Create a naming convention**: Determine what naming convention to use for curated data files and folders (if applicable). Include the details in your technical documentation.
- ✓ **Consider how security for curated data will work**: While designing the curated data storage location, consider who'll need to access the data and how security will be assigned.

Data source decisions

At this point, you should be clear on requirements, data needs, and permissions. Key technical decisions have been made. You now need to make some decisions about the approach for how you'll obtain certain types of data.

Access user activity data

The Power BI user activity data, which is sometimes referred to as the *activity log* or *audit logs*, includes a wealth of information to help you understand what's happening in your Power BI tenant. For more information about identifying your data needs, see User activity data earlier in this article.

Power BI integrates its logs with the Microsoft Purview unified audit log (formerly known as the Microsoft 365 unified audit log). This integration is an advantage because it means every service within Microsoft 365 doesn't have to implement its own unique way of logging. It's enabled by default.

> **Important**
>
> If you're not already extracting user activity data, make that an urgent priority. The user activity data is available for the most recent 30 or 90 days (depending on the source). Even when you're not ready to do in-depth analytics, it's important to begin extracting and storing the data as soon as possible. That way, valuable history will be available to analyze.

You have several options to retrieve user activity data.

Expand table

Technique	Description	Default days of history available	Good choice for manual auditing processes	Good choice for automated auditing processes	Good choice to set up alerting	Good choice to get started quickly
Manual (user interface)	Microsoft Purview compliance portal	90	✓			✓
Manual (user interface)	Defender for Cloud Apps	30	✓		✓	
Programmatic	Power BI activity log (API or PowerShell cmdlet)	28		✓		✓
Programmatic	Office 365 Management Activity API	7		✓		
Programmatic	Microsoft Sentinel (Azure Monitor)	30		✓	✓	

The remainder of this section introduces each of the techniques presented in the table.

Microsoft Purview compliance portal

The audit search tool in the Microsoft Purview compliance portal (formerly known as the Microsoft 365 compliance center) is a convenient place to view activities and alerts. This

tool doesn't require you to create a script to extract and download the data. You can choose a Power BI workload to view all the audit log records for a range of time. You can also narrow down the results by selecting specific activities and users. For example, a manager asks you to find out who deleted a workspace earlier that day so they can quickly contact the person to discuss the situation.

The most recent 90 days of history are available with Audit (Standard). With Audit (Premium), long-term retention allows you to (optionally) retain audit logs longer. Since long-term retention only applies to licensed E5 users, it excludes audit records for non-E5 users and guest users. Therefore, we recommend that you only rely on the default 90-day history to ensure that all activity is captured.

There are licensing requirements to access the audit logs in the Microsoft Purview compliance portal. Elevated Exchange Online permissions are also required in order to access the audit logs.

> **Tip**
>
> By default, Power BI administrators don't have permission to access the unified audit log search in the Microsoft Purview compliance portal. Because it contains data for many Microsoft 365 services, it's a high-privilege role. In most organizations, those permissions are reserved for a small number of IT administrators. There are other options available to Power BI administrators, which are described later in this section.

The user interface in the Microsoft Purview compliance portal is useful for manual auditing processes and occasional investigations of user activities.

Defender for Cloud Apps

The portal in Defender for Cloud Apps is a convenient place to view activities and alerts without the need to create a script to extract and download the data. It also allows you to view data from the Power BI activity log and user sign-ins.

Defender for Cloud Apps includes a user interface to view and search activity logs for various cloud services, including Power BI. It displays the same log events that originate in the Microsoft Purview unified audit log, and it includes other events like user sign-in activity from Microsoft Entra ID.

Like the Microsoft Purview compliance portal (described in the previous section), Defender for Cloud Apps has a searchable user interface. However, the filter options are

different. In addition to the standard 30-day history, you can use Defender for Cloud Apps to view up to six months of activity log history (in seven-day increments).

There are licensing requirements to access Defender for Cloud Apps. Separate permissions are also required.

> **Tip**
>
> By default, Power BI administrators don't have permission to access Defender for Cloud Apps. There's a **Power BI role** in Defender for Cloud Apps. Adding Power BI administrators to this role is a good way to grant them access to a limited set of data.

The user interface in Defender for Cloud Apps is useful for manual auditing processes and one-off investigations of user activities.

Power BI activity log

The Power BI activity log originates from the unified audit log. It contains only user activities related to the Power BI service.

> **Tip**
>
> For organizations that are planning to extract user activities, we recommend they start with the Power BI activity log. It's the simplest source to programmatically access.

You have two options to access the Power BI activity log.

- Call the Get Activity Events REST API by using the tool of your choice.
- Use the Get-PowerBIActivityEvent PowerShell cmdlet. It's available from the Power BI Management Admin module.

For information about which option to use, see Choose APIs or PowerShell cmdlets earlier in this article.

> **Tip**
>
> For examples of how to access the Power BI activity log with PowerShell, see **Access the Power BI activity log**.

Power BI activity log data is available to all Fabric administrators and Power Platform administrators. The data can be accessed from the unified audit log, available to certain Exchange Online roles. For more information, see Track user activities in Power BI.

We recommend that you use the Power BI activity log when your intention is to only retrieve Power BI audit log records.

> **Note**
>
> In the audit log data, workspaces are referred to as *folders*. In the Power BI REST API, workspaces are referred to as *groups*.

Office 365 Management Activity API

You can extract data from the unified audit log for other services, such as SharePoint Online, Teams, Exchange Online, Dynamics 365, and Power BI. When your goal is to implement an auditing and monitoring solution for multiple services, we recommend that you use the Office 365 Management Activity API. Because this API returns data from many services, it's known as the *Unified Auditing API* (or the *unified audit log*). It's one of the Office 365 Management APIs.

Before you build a new solution, we recommend that you contact your IT department to determine whether existing Power BI audit records are available. It's possible that a process already extracts and stores the data. It's also possible that you might obtain permission to access this data rather than build a new solution.

You can access the data one of two ways.

- Call the Office 365 Management Activity API directly by using the tool of your choice. By default, the API returns 24 hours of data. You can retrieve a maximum of seven days of history.
- Use the Search-UnifiedAuditLog PowerShell cmdlet. It's an Exchange Online PowerShell cmdlet. You can retrieve a maximum of 90 days of history.

> **Important**
>
> The `Search-UnifiedAuditLog` cmdlet doesn't scale well and it requires you to implement a looping strategy to overcome its 5,000-row limit. For these reasons, it's suited to occasional queries, or for small organizations that experience low activity. When you only need Power BI data, we recommend that you use the **Get-PowerBIActivityEvent** cmdlet instead.

In general, getting started with the Office 365 Management Activity API is more challenging than using the Power BI activity log (described previously). That's because the API returns content blobs, and each content blob contains individual audit records. For large organizations, there could be thousands of content blobs per day. Because customers and third-party applications heavily use this API, Microsoft implements throttling to ensure that their use of the service doesn't negatively impact others (known as the *noisy neighbor* problem). Therefore, retrieving large volumes of history can be a challenge in larger organizations. For more information, see this troubleshooting article.

To use this API, you must authenticate with a service principal that's been granted permission to retrieve data from the Office 365 Management Activity API.

> **Tip**
>
> By default, Power BI administrators don't have permission to access the Office 365 Management Activity API. Because it contains data for many Microsoft 365 services, access requires the high-privilege administrator or audit roles. In most organizations, these roles are reserved for a small number of IT administrators. The Power BI activity log is intended for use by Power BI administrators.

Retrieving the audit data programmatically from the Office 365 Management Activity API is a good choice when IT needs to extract and store audit logs from various services (beyond Power BI).

Microsoft Sentinel

Microsoft Sentinel is an Azure service that allows you to collect, detect, investigate, and respond to security incidents and threats. You can set up Power BI in Microsoft Sentinel as a *data connector*. When connected, audit logs (that contain a subset of data) are streamed from Power BI into Azure Log Analytics, which is a capability of Azure Monitor.

> **Tip**
>
> Azure Log Analytics is based on the same architecture used by the workspace-level semantic model event logs. For more information about semantic model events logs, see **Data-level auditing**.

Because it's a separate Azure service, any administrator or user that you want to run Kusto Query Language (KQL) queries must be granted permissions in Azure Log Analytics (Azure Monitor). When they have permission, they can query the audit data stored in the *PowerBIActivity* table. However, keep in mind that in most cases you'll

grant users access to the curated data in the gold layer rather than the raw data in the bronze layer.

You use KQL to analyze the activity log events that are stored in Azure Log Analytics. If you have SQL experience, you'll find many similarities with KQL.

There are several ways to access the events stored to Azure Log Analytics. You can use:

- The prebuilt *Log Analytics for Power BI Semantic Models* template app.
- Power BI Desktop connector for Azure Data Explorer (Kusto).
- Web-based query experience in Azure Data Explorer.
- Any query tool that can send KQL queries.

> **⊗ Caution**
>
> Be aware that only a **subset** of the Power BI activity log data is stored in Azure Monitor. When you plan to do comprehensive analysis of Power BI usage and adoption in the organization, we recommend that you use other options (described earlier in this section) to retrieve activity data.

The period of history that you can retrieve depends on the data retention policy for the Azure Log Analytics workspace. Consider cost and access to raw data when deciding on how much data to retain.

Microsoft Sentinel and Azure Monitor are good options when IT has already made an investment with Microsoft Sentinel, the level of detail captured meets your needs, and activities such as threat detection are a priority. However, because Microsoft Sentinel doesn't capture all Power BI activity data, it doesn't support performing comprehensive analysis of Power BI usage and adoption.

User activity data considerations

Here are some considerations to help you choose your source for user activity data.

- **Will it be a manual or automated auditing process?** The user interface options work well for manual auditing processes and occasional one-off queries, especially when you need to investigate a specific activity. An automated auditing process that extracts the user activity data on a schedule will also be necessary to support historical data analysis.
- **How often do you need the user activity data?** For automated auditing processes, plan to extract data that's at least one day before the current date by using Coordinated Universal Time (UTC), rather than your local time. For example, if your

process runs on Friday morning (UTC time), you should extract data from Wednesday. There are several reasons why you should extract data with one day latency.
    - Your files will be simpler to work with when you always extract a full 24 hours at a time, which avoids partial day results.
    - You'll minimize the risk of missing some audit events. Usually, audit events arrive within 30 minutes. Occasionally some events can take up to 24 hours to arrive in the unified audit log.
- **Do you need more than Power BI data?** Consider the most efficient way to access what you need.
    - When you only need Power BI user activity data, use the Power BI activity log.
    - When you need audit logs from multiple services, use the Office 365 Management Activity API to access the unified audit log.
- **Who will develop the solution?** Plan to invest some time to build out the solution. It can take significant effort to produce a production-ready script.

Checklist - When planning how to access user activity data, key decisions and actions include:

- ✓ **Clarify scope of needs**: Determine whether you need to access only Power BI audit records, or audit data for multiple services.
- ✓ **Consult with IT**: Find out whether IT currently extracts audit logs, and whether it's possible to obtain access to the raw data so that you don't need to build a new solution.
- ✓ **Decide on a data source**: Determine the best source to use to access user activity data.
- ✓ **Complete a proof of concept**: Plan to complete a small technical proof of concept (POC). Use it to validate your decisions about how the final solution will be built.
- ✓ **Track additional data needs**: You can correlate activity log data with other sources to enhance the value of the data. Keep track of these opportunities as they arise so they can added to the scope of your project.

Access tenant inventory data

A tenant inventory is an invaluable, and necessary, part of a mature auditing and monitoring solution. It helps you understand what workspaces and content (semantic models, reports, and other items) exist in Power BI, and it's an excellent complement to

the user activity data (described in the previous section). For more information about identifying your data needs, see Tenant inventory earlier in this article.

User activities (previously described) are audited events; they're a record of actions that a user took at a specific date and time. However, the tenant inventory is different. The tenant inventory is a *snapshot* at a given point in time. It describes the current state of all published items in the Power BI service at the time you extracted it.

> **Note**
>
> Power BI REST API documentation refers to published items as *artifacts*.

> **Tip**
>
> Many organizations find it helpful to extract the tenant inventory at the same time of day every week.

To properly analyze what's happening in your Power BI tenant, you need both the user activity data and the tenant inventory. Combining them allows you to understand how much content you have and where it's located. It also allows you to find unused or under-utilized content (because there won't be any events for it in the activity log). The tenant inventory also helps you compile a list of current names for all items, which is helpful when item names change.

For more information about the value of the tenant inventory, see Tenant inventory earlier in this article.

> **Tip**
>
> You can use the **Get Unused Artifacts as Admin** API to search for items that don't have any user activity in that last 30 days. However, you can't customize that time period. For example, you might have critical content that's only used quarterly. By combining your tenant inventory with the user activity data, you can find unused items in ways that you can customize.

You can obtain tenant inventory data in two different ways.

Expand table

Technique	Description	Most suited to	Good choice for manual auditing processes	Good choice for automated auditing processes
Programmatic	The `Get Groups as Admin` API or the `Get-PowerBIWorkspace` PowerShell cmdlet can provide a list of workspaces for the entire tenant. Optionally, individual items (like semantic models and reports) per workspace can be included.	Smaller organizations or low data volume	✅	✅
Programmatic	A set of asynchronous admin APIs, collectively known as the *metadata scanning APIs* or *scanner APIs*, can return a list of workspaces and individual items. Optionally, detailed metadata (like tables, columns, and measure expressions) can be included as well.	Organizations with high data volume and/or a need to obtain detailed results		✅

The remainder of this section introduces each of the techniques presented in the table.

Groups APIs or workspaces cmdlet

To retrieve a list of workspaces, you can use one of several REST APIs, such as the Get Groups as Admin API (using the tool of your choice). The results include extra metadata such as the description and whether the workspace is stored in a Premium capacity.

> **Note**
>
> The API named includes the term *group* as a reference to a workspace. That term originated from the original Power BI security model, which relied on Microsoft 365 groups. While the Power BI security model has evolved significantly over the years, the existing API names remain unchanged to avoid breaking existing solutions.

The `Get Groups as Admin` REST API includes the helpful `$expand` parameter. This parameter allows you to optionally expand the results so that they include a list of items published to the workspace (such as semantic models and reports). You can also pass

the `users` data type to the `$expand` parameter to include the users who are assigned to a workspace role.

You can also use the Get-PowerBIWorkspace PowerShell cmdlet. It's from the Power BI Management Workspaces module. When you set the `-Scope` parameter `organization`, it functions like the `Get Groups as Admin` API. The cmdlet also accepts the `-Include` parameter (which is the equivalent of the `$expand` parameter in the REST API) to include items published in the workspace (such as semantic models and reports).

> **Tip**
>
> By passing in parameters, you can use the cmdlet in different ways. This section focuses on retrieving a tenant-wide inventory. For more information about using the `-Scope` parameter, see **Choose a user API or admin API** earlier in this article.

To help you choose which option to use, see Choose APIs or PowerShell cmdlets earlier in this article.

The `Get Groups as Admin` REST API or the `Get-PowerBIWorkspace` PowerShell cmdlet is best suited to manual auditing processes and one-off investigations. Larger organizations with high data volume typically find these options challenging to use. The REST API and cmdlet always extract a full set of data, so they can take a long time to run. So, it's also likely that larger organizations will run into limitations on the number of requests allowed per hour (known as *throttling*, which is done to protect the health of the service). The metadata scanning APIs (described next) provide a more reliable and scalable alternative.

Metadata scanning APIs

The metadata scanning APIs, commonly called the *scanner APIs*, are a set of APIs that return a list of workspaces and their Power BI items (semantic models, reports, and other items). Conceptually, they provide the same data (and more) as the Groups APIs or the workspace cmdlet, which are described in the previous section. However, the method you use to retrieve the data is different and better suited to extracting the tenant inventory.

> **Note**
>
> Take notice of how some people use the term *scanner APIs* or the phrase *scanning the tenant*. They often use those terms to mean *compiling a tenant inventory*,

distinguishing it from the user activity events. They might, or might not, be literally referring to the use of the metadata scanning APIs.

There are two primary reasons why you should consider using the metadata scanning APIs instead of the `Get Groups as Admin` REST API or the `Get-PowerBIWorkspace` cmdlet (described previously).

- **Incremental data extract**: The scanner APIs only extract data that's changed since the last time it was run. Conversely, the `Get Groups as Admin` REST API and the `Get-PowerBIWorkspace` cmdlet are synchronous operations that extract the full set of data every time they run.
- **More granular level of detail**: The scanner APIs can retrieve granular detail (like tables, columns, and measure expressions), providing permission is granted by the two tenant settings for detailed metadata. Conversely, the lowest level of detail available with the `Get Groups as Admin` REST API and the `Get-PowerBIWorkspace` cmdlet is by item type (for example, a list of reports in a workspace).

Using the scanner APIs involves more effort because you need to call a series of APIs. Also, they run as an *asynchronous* process. An asynchronous process runs in the background, and so you don't have to wait for it to finish. You might need to collaborate with IT, or a developer, when it's time to create a production-ready script that will be scheduled.

Here's the sequence of steps your process should follow when using the scanner APIs.

1. **Sign in**: Sign in to the Power BI service by using a Power BI administrator account or a service principal that has permission to run admin APIs. For more information, see Determine the authentication method earlier in this article.
2. **Get the workspace IDs**: Send a request to retrieve a list of workspace IDs. The first time it's run, you won't have a modified date, so it will return a complete list of workspaces. Usually, you'll pass the modified date to retrieve only workspaces that have changed since that point in time. The modified date must be within the last 30 days.
3. **Initiate a metadata scan**: Initiate a call to request a scan of workspace information by passing in the workspace IDs retrieved in Step 2. Note that this is a *POST* API (instead of a *GET* API, which is more common when retrieving audit data). A POST API is an HTTP request that creates or writes data for a specified resource. This query specifies the level of detail that will be extracted, such as data source details, semantic model schema, semantic model expressions, or users. When submitted, an ID for the scan is returned by the API. It also returns the date and time of the scan, which you should record as the snapshot date.

4. **Check the scan status**: Use the scan ID obtained in Step 3 to get the scan status. You might need to call this API more than once. When the returned status is *Succeeded*, can proceed to the next step. The time it takes to scan depends on how much data you request.
5. **Get the results**: Use the scan ID obtained in Step 3 to get the scan result and extract the data. You must do this step within 24 hours of completing Step 4. Keep in mind that the snapshot time stamp should be correlated with Step 3 rather than Step 5 (when there's a significant delay). That way, you'll have an accurate snapshot time stamp. Save the results in your preferred data storage location. As already described in this article, we recommend that you store the raw data in its original state.
6. **Prepare for the next process**: Record the time stamp of the scan from Step 3 so you can use it in Step 2 the next time you run the process. That way, you'll only extract data that's changed since that point in time. Moving forward, all subsequent data extracts will be incremental changes rather than full snapshots.

Checklist - When planning for access to the tenant inventory data, key decisions and actions include:

- ✓ **Confirm requirements**: Clarify needs for compiling a tenant inventory and the analytical requirements for the data.
- ✓ **Decide on the frequency for extracting tenant inventory**: Confirm how often you'll need a new tenant inventory (such as every week).
- ✓ **Decide how to extract the tenant inventory**: Confirm which method you will use to obtain the tenant inventory data (API, cmdlet, or metadata scanning APIs).
- ✓ **Complete a proof of concept**: Plan to complete a small technical proof of concept (POC). Use it to validate your decisions about how the final solution will be built.

Access user and group data

In addition to the identifying information (like an email address) that's included in the user activity data, it's valuable to retrieve additional information such as location or organizational details. You can use Microsoft Graph to retrieve data about users, groups, service principals, and licenses. Microsoft Graph comprises a set of APIs and client libraries that allow you to retrieve audit data from a wide variety of services.

Here are some details about the Microsoft Entra objects that you can access.

- **User**: An identity that exists in Microsoft Entra ID as a work, school, or Microsoft account. The term *domain user* is often used to describe organizational users, while the formal term is *user principal name* (UPN). A UPN is usually the same value as the user's email address (however, if an email address changes, the UPN doesn't change because it's immutable). There's also a unique Microsoft Graph ID for each user. Often, a user account is associated with one person. Some organizations create users that are *service accounts* that are used for automated activities or for administrative tasks.
- **Service principal**: A different type of identity, that's provisioned when you create an app registration. A service principal is intended for unattended, automated activities. For more information, see Determine the authentication method earlier in this article.
- **Group**: A collection of users and service principals. There are several types of groups that you can use to simplify permissions management. For more information, see Strategy for using groups.

> **Note**
>
> When this article refers to *users and groups*, this term includes service principals as well. This shorter term is used for brevity.

The users and groups data that you retrieve is a *snapshot* that describes the current state at a given point in time.

> **Tip**
>
> For more information about users, service principals, and groups, see **Integration with Microsoft Entra ID**.

Analytical attributes

For Power BI tenant-level auditing, you might extract and store the following attributes from Microsoft Graph.

- **Full name of users**: Many data sources only include the email address of the user that performed an activity or who's assigned to a role. Use this attribute when you want to display the full name (display name) in analytical reports. Using the full name makes reports more user-friendly.
- **Other user properties**: Other descriptive attributes about users might be available in Microsoft Entra ID. Some examples of built-in user profile attributes that have

analytical value include job title, department, manager, region, and office location.

- **Members of a security group**: Most data sources provide the name of a group (for example, the Power BI activity log records that a security group was assigned to a workspace role). Retrieving the group membership improves your ability to fully analyze what an individual user is doing or has access to.
- **User licenses**: It's useful to analyze which user licenses—free, Power BI Pro, or Power BI Premium Per User (PPU)—are assigned to users. This data can help you to identify who's not using their license. It also helps you to analyze all users (distinct users with a license) versus active users (with recent activities). If you're considering adding or expanding your Power BI Premium licenses, we recommend that you analyze the user license data together with user activity data to perform a cost analysis.
- **Members of the administrator roles**: You can compile a list of your Power BI administrators are (which includes Power Platform administrators).

For the authoritative reference of Power BI license information that you can find in the audit data from Microsoft Graph, see Product names and service plan identifiers for licensing.

> **Tip**
>
> Retrieving members from groups can be one of the most challenging aspects of obtaining audit data. You will need to do a *transitive search* to flatten out all nested members and nested groups. You can do a transitive search for **group members**. This type of search is especially challenging when there are thousands of groups in your organization. In that case, there might consider better alternatives to retrieve the data. One option is to extract all groups and group members from Microsoft Graph. However, that might not be practical when only a small number of groups is used for Power BI security. Another option is to pre-build a reference list of groups that are used by any type of Power BI security (workspace roles, app permissions, per-item sharing, row-level security, and others). You can then loop through the reference list to retrieve **group members** from Microsoft Graph.

Here are some other attributes you might extract and store.

- **User type**: Users are either members or guests. Most commonly, members are internal users and guests are external (B2B) users. Storing user type is useful when you need to analyze the activities of external users.
- **Role changes**: When you perform a security audit, it's useful to understand when a user changed roles in the organization (for example, when they transfer to a

different department). That way, you can verify whether their Power BI security settings have been—or should be—updated.

- **Disabled users**: When a user leaves the organization, usually an administrator disables their account. You can create a process to check whether disabled users are workspace administrators or semantic model owners.

> **Tip**
>
> The Power BI activity log includes an event that records when a user signs up for a trial license. You can combine that event with the user license data (sourced from Microsoft Entra ID) to produce a complete picture.

Retrieve users and groups data

You can retrieve data about users and groups in different ways.

Expand table

Technique	Description	Good choice for manual auditing processes	Good choice for automated auditing processes
Manual	Graph Explorer	✓	
Programmatic	Microsoft Graph APIs and SDKs		✓
Programmatic	Az PowerShell module		✓

The remainder of this section introduces each of the techniques presented in the table. Other techniques, which are deprecated and shouldn't be used for new solutions, are described at the end of this section.

> **Note**
>
> Be careful when you read information online because many tool names are similar. Some tools in the Microsoft ecosystem include the term *Graph* in their name, like Azure Resource Graph, GraphQL, and the Microsoft Security Graph API. Those tools aren't related to Microsoft Graph, and they're out of scope for this article.

Microsoft Graph Explorer

Microsoft Graph Explorer is a developer tool that lets you learn about Microsoft Graph APIs. It's a simple way to get started because it requires no other tools or setup on your machine. You can sign in to retrieve data from your tenant, or retrieve sample data from a default tenant.

You can use Graph Explorer to:

- Manually send a request to a Microsoft Graph API to check whether it returns the information that you want.
- See how to construct the URL, headers, and body before you write a script.
- Check data in an informal way.
- Determine which permissions are required for each API. You can also provide consent for new permissions.
- Obtain code snippets to use when you create scripts.

Use this link to open Graph Explorer.

Microsoft Graph APIs and SDKs

Use the Microsoft Graph APIs to retrieve users and groups data. You can also use it to retrieve data from services such as Microsoft Entra ID, SharePoint Online, Teams, Exchange, Outlook, and more.

The Microsoft Graph SDKs act as an *API wrapper* on top of the underlying Microsoft Graph APIs. An SDK is a *software development kit* that bundles tools and functionality together. The Microsoft Graph SDKs expose the entire set of Microsoft Graph APIs.

You can choose to send requests directly to the APIs. Alternatively, you can add a layer of simplification by using your preferred language and one of the SDKs. For more information, see Choose APIs or PowerShell cmdlets earlier in this article.

The Microsoft Graph SDKs support several languages, and there's also the Microsoft Graph PowerShell modules. Other SDKs are available for Python, C#, and other languages.

> **ⓘ Important**
>
> The Microsoft Graph PowerShell module replaces the Azure AD PowerShell modules and MSOnline (MSOL) modules, which are both deprecated. You shouldn't create new solutions with deprecated modules. The Microsoft Graph PowerShell module has many **features and benefits**. For more information, see **Upgrade from Azure AD PowerShell to Microsoft Graph PowerShell**.

You can install the Microsoft Graph PowerShell modules from the PowerShell Gallery. Because Microsoft Graph works with many Microsoft 365 services, there are many PowerShell modules that you install.

For Power BI tenant-level auditing, here are the most common PowerShell modules you'll need to install.

- Authentication (for signing in)
- Users
- Groups
- Applications (and service principals)
- Directory objects (and license details)

> **Tip**
>
> Microsoft updates the Microsoft Graph PowerShell modules regularly. Sometimes, preview modules are made available on a pre-release basis or a beta endpoint. You might want to specify the version you're interested in when you install and update the modules. Also, when you review online documentation, note that the current version number is automatically appended to the end of the URL (so be careful when saving or sharing URLs).

Az PowerShell module

You can also use the Az PowerShell module to retrieve users and groups data. It focuses on Azure resources.

> **Important**
>
> The Az PowerShell module replaces the AzureRM PowerShell modules, which are deprecated. You shouldn't create new solutions with deprecated modules.

You can use the Invoke-AzRestMethod cmdlet when there isn't a corresponding cmdlet for an API. It's a flexible approach that allows you to send a request to any Microsoft Graph API by using the Az PowerShell module.

Beginning with Az version 7, the Az cmdlets now reference the Microsoft Graph API. Therefore, the Az module acts as an *API wrapper* on top of Microsoft Graph. The Az modules have a subset of functionality that's contained in the Microsoft Graph APIs and PowerShell modules. For new solutions, we recommend that you use the Microsoft Graph PowerShell SDK.

Deprecated APIs and modules

You might find articles and blog posts online that suggest alternative options that aren't presented in this section. We strongly recommend that you ***do not*** create new solutions (and/or migrate your existing solutions) by using any of the following APIs or modules.

- **AzureRM PowerShell modules**: Deprecated and will be retired. They've been replaced with the Az PowerShell module.
- **Azure AD Graph API and Azure AD PowerShell module**: Deprecated and will be retired. This change is the result of the migration from Azure AD Graph to Microsoft Graph (note that Graph appears in both names, but *Microsoft Graph* is the future direction). All future PowerShell investments will be made in the Microsoft Graph PowerShell SDK.
- **MS Online (MSOL) PowerShell module**: Deprecated and will be retired. All future PowerShell investments will be made in the Microsoft Graph PowerShell SDK.

> ⊗ **Caution**
>
> Be sure to confirm the status of any deprecated API or module that you're currently using. For more information about the retirement of AzureRM, see **this announcement**.
>
> For more information about Microsoft Entra ID and MSOL, see **this retirement announcement post**.
>
> If you have questions or require clarification on the future direction of programmatic data access, contact your Microsoft account team.

Checklist - When planning to access users and groups data, key decisions and actions include:

- ✓ **Confirm requirements**: Clarify needs for compiling data related to users, groups, and licenses.
- ✓ **Prioritize requirements**: Confirm what the top priorities are so you know what to spend time on first.
- ✓ **Decide on the frequency for extracting data**: Determine how often you'll need a new snapshot of the users and groups data (such as every week or every month).

- ✓ **Decide how to extract data with Microsoft Graph**: Determine which method you'll use to retrieve the data.
- ✓ **Complete a proof of concept**: Plan to complete a small technical proof of concept (POC) to extract users and groups data. Use it to validate your decisions about how the final solution will be built.

Access data from Power BI REST APIs

Perhaps as a lower priority, you can also retrieve other data by using the Power BI REST APIs.

For example, you can retrieve data about:

- All apps in the organization.
- All imported semantic models in the organization.
- All deployment pipelines in the organization.
- All Premium capacities in the organization.

During a security audit, you might want to identify:

- Items that have been widely shared with the entire organization.
- Items that are available on the public internet by using publish to the web.

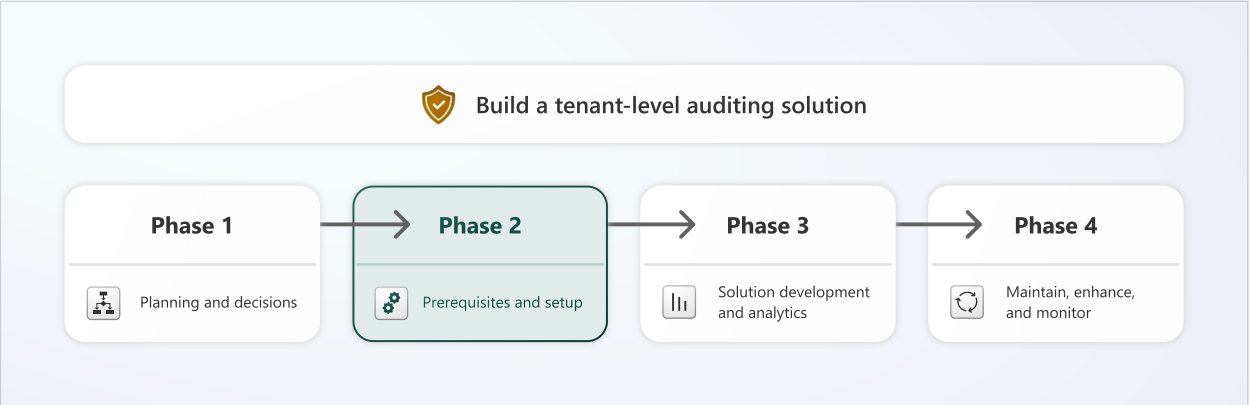
For more information about the other types of APIs, see Choose a user API or admin API earlier in this article.

Checklist - When planning for accessing data from the Power BI APIs, key decisions and actions include:

- ✓ **Obtain requirements**: Compile analytical requirements as they arise. Keep track of them in your backlog.
- ✓ **Prioritize requirements**: Set priorities for the new requirements that arise.

Phase 2: Prerequisites and setup

The second phase of planning and implementing a tenant-level auditing solution focuses on prerequisites and setup that must be done before you begin solution development.



Create storage account

At this point, you've decided on a location to store raw data and how to create curated data. Based on those decisions, you're now ready to create a storage account. Depending on your organization's processes, it might involve submitting a request to IT.

As described earlier, we recommend using a technology that allows you to write the raw data to immutable storage. Once the data is written, it can't be changed or deleted. You can then have confidence in the raw data because you know that an administrator with access can't alter it in any way. The curated data, however, doesn't (usually) need to be stored in immutable storage. Curated data might change or it can be regenerated.

Checklist – When creating a storage account, key decisions and actions include:

- ✓ **Create a storage account**: Create, or submit a request, for creating a storage account. Use immutable storage settings for the raw data, whenever possible.
- ✓ **Set permissions**: Determine which permissions should be set for the storage account.
- ✓ **Test access**: Do a small test to ensure that you can read and write to the storage account.
- ✓ **Confirm responsibilities**: Ensure that you're clear on who'll manage the storage account on an ongoing basis.

Install software and set up services

At this point, you've made your decisions about which technology to use for access audit data. Based on those decisions, you're now ready to install software and set up services.

Set up the preferred development environment for each administrator. The development environment will allow them to write and test scripts. Visual Studio Code is a modern tool for developing scripts, so it's a good option. Also, many extensions are available to work with Visual Studio Code.

If you've made the decision (previous described) to use PowerShell, you should install PowerShell Core and the necessary PowerShell modules on:

- The machine of each administrator/developer who will write or test auditing scripts.
- Each virtual machine or server that will run scheduled scripts.
- Each online service (such as Azure Functions or Azure Automation).

If you've chosen to use any Azure services (such as Azure Functions, Azure Automation, Azure Data Factory, or Azure Synapse Analytics), you should provision and set them up as well.

Checklist – When installing software and setting up services, key decisions and actions include:

- ✓ **Set up administrator/developer machines**: If applicable, install the necessary tools that will be used for development.
- ✓ **Set up servers**: If applicable, install the necessary tools on any servers or virtual machines that are present in your architecture.
- ✓ **Set up Azure services**: If applicable, provision and set up each Azure service. Assign permissions for each administrator/developer who'll be doing development work.

Register a Microsoft Entra application

At this point, you've decided how to authenticate. We recommend that you register a Microsoft Entra application (service principal). Commonly referred to as an *app registration*, it should be used for unattended operations that you'll automate.

For more information about how to create an app registration to retrieve tenant-level auditing data, see Enable service principal authentication for read-only admin APIs.

Checklist - When registering a Microsoft Entra application, key decisions and actions include:

- ✓ **Check whether an existing app registration exists**: Verify with IT whether an existing app registration is available for the purpose of running admin APIs. If so, determine whether the existing one should be used, or whether a new one should be created.
- ✓ **Create a new app registration**: Create an app registration when appropriate. Consider using an app name such as *PowerBI-AdminAPIs-EntraApp*, which clearly describes its purpose.
- ✓ **Create a secret for the app registration**: Once the app registration exists, create a secret for it. Set the expiration date based on how often you intend to rotate the secret.
- ✓ **Securely save the values**: Store the secret, app ID (client ID), and the tenant ID because you'll need them to authenticate with the service principal. Use a location that's secure, such as an organizational password vault. (If you need to request creation of an app registration from IT, specify that you need these values returned to you.)
- ✓ **Create a security group**: Create (or request) a security group that will be used for Power BI. Consider using group name such as *Power BI admin service principals*, which signifies that the group will be used to access tenant-wide metadata.
- ✓ **Add the service principal as a member of the security group**: Use the app ID (client ID) to add the new (or existing) service principal as a member of the new security group.
- ✓ **Update admin API tenant setting in Fabric**: In the Fabric admin portal, add the security group to the *Allow service principals to use read-only Power BI admin APIs* tenant setting.
- ✓ **Skip assigning permissions in Azure**: Don't delegate any permissions to the app registration (it'll gain access to the Power BI tenant-level audit data by way of the Power BI *Allow service principals to use read-only Power BI admin APIs* tenant setting).
- ✓ **Decide whether the app registration should access detailed metadata**: Determine whether you want to extract detailed information about semantic model tables, columns, and measure expressions when you build your tenant inventory.
- ✓ **Update the detailed metadata tenant settings in Power BI**: Optionally, in the Fabric admin portal, add the security group to the *Enhance admin API responses with detailed metadata* tenant setting and also the *Enhance admin API responses with DAX and mashup expressions* tenant setting.
- ✓ **Test the service principal**: Create a simple script to sign in by using the service principal and test that it can retrieve data from an admin API.

- ✓ **Create a process to manage app registration secrets**: Create documentation and a process to regularly rotate secrets. Determine how you'll securely communicate a new secret to any administrators and developers who need it.

Set Fabric tenant settings

There are several tenant settings in the Fabric admin portal that are relevant for extracting tenant-level auditing data.

Admin APIs

There are three tenant settings that are relevant for running admin APIs.

> ⓘ **Important**
>
> Because these settings grant metadata access for the entire tenant (without assigning direct Power BI permissions), you should control them tightly.

The *Allow service principals to use read-only Power BI admin APIs* tenant setting allows you to set which service principals can call admin APIs. This setting also allows Microsoft Purview to scan the entire Power BI tenant so that it can populate the data catalog.

> ⓘ **Note**
>
> You don't need to explicitly assign other Power BI administrators to the *Allow service principals to use read-only Power BI admin APIs* tenant setting because they already have access to tenant-wide metadata.

The *Enhance admin API responses with detailed metadata* tenant setting allows you to set which users and service principals can retrieve detailed metadata. Metadata is retrieved by using the metadata scanning APIs, and it includes tables, columns, and more. This setting also allows Microsoft Purview to access schema-level information about Power BI semantic models so that it can store it in the data catalog.

The *Enhance admin API responses with DAX and mashup expressions* tenant setting allows you to set which users and service principals can retrieve detailed metadata. Metadata is retrieved from the metadata scanning APIs, and it can include queries and semantic model measure expressions.

> ⓘ **Note**

The *Allow service principals to use read-only Power BI admin APIs* tenant setting is specifically about accessing *admin APIs*. Its name is very similarly to the tenant setting that's intended for accessing *user APIs* (described next). For more information about the difference between admin APIs and user APIs, see Choose a user API or admin API earlier in this article.

User APIs

There's one tenant setting that applies to calling user APIs. In this situation, Power BI permissions are also required for the service principal (for example, a workspace role).

The *Allow service principals to use Power BI APIs* tenant setting allows you to set which service principals have access to run REST APIs (excluding the admin APIs, which are granted by a different tenant setting, described above).

There are other tenant settings related to APIs, which allow access to the embedding APIs, streaming APIs, export APIs, and the *execute queries* API. However, these APIs are out of scope for this article.

For more information about the tenant settings for usage metrics, see Report-level auditing.

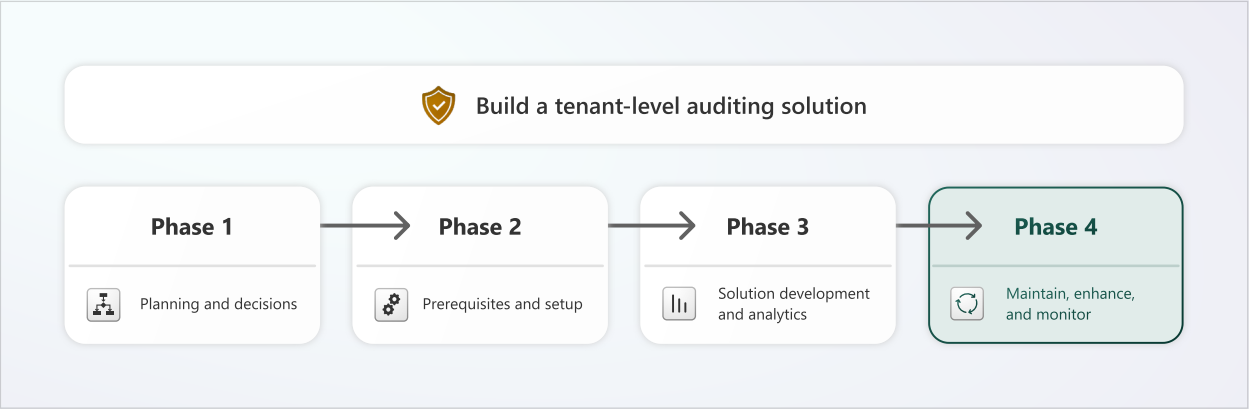
Checklist - When setting up the Fabric tenant settings, key decisions and actions include:

- ✓ **Verify that each service principal is in the correct group**: Confirm that the *Power BI admin service principals* group includes the correct service principals.
- ✓ **Update the admin API tenant setting in Power BI**: Add the security group to the *Allow service principals to use read-only Power BI admin APIs* tenant setting, which allows using the admin APIs to retrieve tenant-wide metadata.
- ✓ **Update the detailed metadata tenant settings in Power BI**: If necessary, add the security group to the *Enhance admin API responses with detailed metadata* tenant setting and the *Enhance admin API responses with DAX and mashup expressions* tenant setting.
- ✓ **Confirm which user APIs will be accessed**: Verify whether one or more user APIs will be needed (in addition using the admin APIs).
- ✓ **Update the user API tenant setting in Power BI**: Add the security group to the *Allow service principals to use Power BI APIs* tenant setting, which is intended for

user APIs.

Phase 3: Solution development and analytics

The third phase of planning and implementing a tenant-level auditing solution focuses on solution development and analytics. At this point, all the planning and decision-making has occurred, and you've met prerequisites and completed the setup. You're now ready to begin solution development so that you can perform analytics and gain insights from your auditing data.



Extract and store the raw data

At this point, your requirements and priorities should be clear. The key technical decisions have been made about how to access audit data and the location to store audit data. Preferred tools have been selected, and the prerequisites and settings have been set up. During the previous two phases, you might have completed one or more small projects (proofs of concept) to prove feasibility. The next step is to set up a process to extract and store the raw auditing data.

Like with data returned by most Microsoft APIs, auditing data is formatted as JSON. JSON-formatted data is self-describing because it's human-readable text that contains structure and data.

JSON represents data objects that consist of attribute-value pairs and arrays. For example, `"state": "Active"` is an example where the *state* attribute value is *Active*. A JSON array contains an ordered list of elements separated by a comma and which are enclosed within brackets ([ ]). It's common to find nested arrays in JSON results. Once you become familiar with the hierarchical structure of a JSON result, it's straightforward to understand the data structure, like a list (array) of semantic models in a workspace.

Here are some considerations for when you extract and store the raw data retrieved from the APIs.

- **What naming convention will be used?** For a file-based system, a naming convention is necessary for files, folders, and data lake zones. For a database, a naming convention is necessary for tables and columns.
- **What format will be used to store the raw data?** As Power BI continues to introduce new features, new activities will appear that don't exist today. For this reason, we recommend that you extract and retain the original JSON results. Don't parse, filter, or format the data while it's extracted. That way, you can use the original raw data to regenerate your curated audit data.
- **What storage location will be used?** A data lake or blob storage is commonly used to store raw data. For more information, see Determine where to store audit data earlier in this article.
- **How much history will you store?** Export the audit data to a location where you can store the history. Plan to accumulate at least two years of history. That way, you can analyze trends and changes beyond the default 28-day retention period. You might choose to store the data indefinitely, or you might decide on a shorter period, depending on the data retention policies for your organization.
- **How will you track when the process last ran?** Set up a configuration file, or the equivalent, to record the time stamps when a process starts and finishes. The next time the process runs, it can retrieve these time stamps. It's especially important that you store time stamps when you extract data by using the metadata scanning APIs.
- **How will you handle throttling?** Some Power BI REST APIs and the Microsoft Graph API implement throttling. You'll receive a 429 error (too many requests) if your API request is throttled. Throttling can be a common problem for larger organizations that need to retrieve a large volume of data. How you avoid failed attempts due to throttling depends on the technology you use to access and extract the data. One option is to develop logic in your scripts that responds to a 429 "Too many requests" error by retrying after a wait period.
- **Is the audit data needed for compliance?** Many organizations use the raw audit log records to do compliance audits or to respond to security investigations. In these cases, we strongly recommend that you store the raw data in immutable storage. That way, once the data is written, it can't be changed or deleted by an administrator or other user.
- **What storage layers are necessary to support your requirements?** The best places to store raw data are a data lake (like Azure Data Lake Storage Gen2) or object storage (like Azure Blob Storage). A file system can also be used if cloud services aren't an option.

Checklist - When extracting and storing the raw data, key decisions and actions include:

- ✓ **Confirm requirements and priorities**: Clarify the requirements and priorities for the data that you'll extract first.
- ✓ **Confirm the source of data to be extracted**: Verify the source for each type of data you need.
- ✓ **Set up a process to extract the data and load it to the raw data zone**: Create the initial process to extract and load the raw data in its original state, without any transformations. Test that the process works as intended.
- ✓ **Create a schedule to run the processes**: Set up a recurring schedule to run the extract, load, and transform processes.
- ✓ **Verify that credentials are managed securely**: Confirm that passwords, secrets, and keys are stored and communicated in secure ways.
- ✓ **Confirm security is set up correctly**: Verify that access permissions are set up correctly for the raw data. Ensure that administrators and auditors can access the raw data.

For more information about how an auditing and monitoring solution grows over time, see Operationalize and improve later in this article.

Create the curated data

At this point, the raw data is extracted and stored. The next step is to create a separate gold data layer for the curated data. Its goal is to transform and store the data files in a star schema. A star schema comprises dimension tables and fact tables, and it's intentionally optimized for analysis and reporting. The files in the curated data layer become the source of a Power BI data model (described in the next topic).

> **Tip**
>
> When you expect there to be more than one data model, investing in a centralized curated data layer is particularly useful.

Checklist - When creating the curated data layer, key decisions and actions include:

- ✓ **Confirm requirements and priorities**: If you intend to use an intermediary silver layer for the transformed data, clarify the requirements and objectives for what you

need to accomplish.

- ✓ **Set up a process to transform the raw data and load it into the curated data zone**: Create a process to transform and load the data into a star schema. Test that the process works as intended.
- ✓ **Create a schedule to run the processes**: Set up a recurring schedule to populate the curated data layer.
- ✓ **Confirm security is set up correctly**: Verify that access permissions are set up correctly for the curated data. Ensure that developers of the data model can access the curated data.

Create a data model

The topic is about setting up an analytical *data model*. A data model is query-able data resource optimized for analytics. Sometimes it's referred to as a *semantic model*, or simply a *model*. For your auditing and monitoring solution, the data model will likely be implemented as a Power BI semantic model.

In the context of auditing and monitoring, a data model sources data from the curated (gold) data layer. If you choose not to create a curated data layer, the data model sources its data directly from the raw data.

We recommend that your Power BI data model implements a star schema design. When the source data is the curated data layer, the Power BI data model star schema should mirror the curated data layer star schema.

> **Tip**
>
> For an overview of star schema design, see **Understand star schema and the importance for Power BI**.

As with any Power BI project, creating a data model is an iterative process. You can add new measures as needed. You can also add new tables and columns as new audit events become available. In time, you might even integrate new data sources.

Here are some useful dimension tables that you can include in the data model.

- **Date**: A set of date attributes to enable analysis (slicing and dicing) of data by day, week, month, quarter, year, and other relevant time periods.
- **Time**: If you need to analyze by time of day and you have a very large volume of audit data, consider storing the time part separately from the date. This approach can help to improve query performance.

- **Users**: Attributes that describe users (such as department and geographic region) that can filter many subjects of auditing data. The goal is to remove all user details from the fact tables and store them in this dimension table so that they can filter many fact tables. You can also store service principals in this table.
- **Activity events**: Attributes that group and describe the activity events (operations). To enhance your reporting, you might create a data dictionary that describes each activity event. You might also create a hierarchy that groups and classifies similar activity events. For example, you might group all item creation events, delete events, and so on.
- **Workspaces**: A list of workspaces in the tenant and workspace properties, such as type (personal or standard) and description. Some organizations record more details about workspaces (possibly using a SharePoint list). You can integrate these details into this dimension table. You need to decide whether this dimension table stores only the current state of workspace, or whether it stores versioned data that reflects significant workspace changes over time. For example, when a workspace name changes, does historical reporting show the current workspace name or the workspace name that was current at that time? For more information about versioning, see Slowly changing dimensions.
- **Item types**: A list of Power BI item types (semantic models, reports, and others).
- **Capacities**: A list of Premium capacities in the tenant.
- **Gateways**: A list of data gateways in the tenant.
- **Data sources**: A list of data sources that are used by any semantic model, dataflow, or datamart.

Here are some useful fact tables (subjects) that you can include in the data model.

- **User activities**: The fact data that's sourced from the original JSON data. Any attributes that have no analytical value are removed. Any attributes that belong in the dimension tables (above) are removed too.
- **Tenant inventory**: A point-in-time snapshot of all items published in the tenant. For more information, see Tenant inventory earlier in this article.
- **Semantic models**: Includes user activity involving semantic models (like semantic model changes), or related data sources.
- **Semantic model refreshes**: Stores data refresh operations, including details about type (scheduled or on-demand), duration, status, and which user initiated the operation.
- **Workspace roles**: A point-in-time snapshot of workspace role assignments.
- **User licenses**: A point-in-time snapshot of user licenses. While you might be tempted to store the user license in the **Users** dimension table, that approach won't support the analysis of license changes and trends over time.

- **User group memberships**: A point-in-time snapshot of users (and service principals) assigned to a security group.
- **Community activities**: Includes community-related facts such as training events. For example, you could analyze Power BI user activities compared to training attendance. This data could help the Center of Excellence identify potential new champions.

Fact tables shouldn't include columns that report creators will filter. Instead, those columns belong to related dimension tables. Not only is this design more efficient for queries, but it also promotes reuse of dimension tables by multiple facts (known as *drill across*). That last point is important to produce a useful and user-friendly data model that's extensible when you add new fact tables (subjects).

For example, the **Users** dimension table will be related to every fact table. It should be related to the **User activities** fact table (who performed the activity), **Tenant inventory** fact table (who created the published item), and all other fact tables. When a report filters by a user in the **Users** dimension table, visuals in that report can show facts for that user from any related fact table.

When you design your model, ensure that an attribute is visible once, and only once, in the model. For example, the user email address should only be visible in the **Users** dimension table. It will exist in other fact tables too (as a dimension key to support a model relationship). However, you should hide it in each fact table.

We recommend that you create your data model separate from reports. The decoupling of a semantic model and its reports results in a centralized semantic model that can serve many reports. For more information about using a shared semantic model, see the managed self-service BI usage scenario.

Consider setting up row-level security (RLS) so that other users—beyond the Center of Excellence, auditors, and administrators—can analyze and report on auditing data. For example, you could use RLS rules to allow content creators and consumers to report on their own user activities or development efforts.

Checklist - When creating the data model, key decisions and actions include:

- ✓ **Plan and create the data model**: Design the data model as a star schema. Validate relationships work as intended. As you develop the model, iteratively create

measures and add additional data based on analytical requirements. Include future improvements on a backlog, when necessary.

- ✓ **Set up RLS**: If you intend to make the data model available to other general users, set up row-level security to restrict data access. Validate that the RLS roles return the correct data.

Enhance the data model

To effectively analyze content usage and user activities, we recommend that you enrich your data model. Data model enhancements can be done gradually and iteratively over time as you discover opportunities and new requirements.

Create classifications

One way to enhance the model and increase the value of your data is to add classifications to the data model. Ensure that these classifications are used consistently by your reports.

You might choose to classify *users* based on their level of usage, or to classify *content* based on its level of usage.

User usage classification

Consider the following classifications for *user usage*.

- **Frequent user**: Activity recorded in the last week, and in nine of the last 12 months.
- **Active user**: Activity recorded in the past month.
- **Occasional user**: Activity recorded in the last nine months, but without activity in the past one month.
- **Inactive user**: No activity recorded in the last nine months.

> **Tip**
>
> It's helpful to know who your occasional or inactive users are, especially when they have Pro or PPU licenses (which involve cost). It's also helpful to know who your frequent and most active users are. Consider inviting them to join office hours or attend training. Your most active content creators might be candidates to join your champions network.

Content usage classification

Consider the following classifications for *content usage*.

- **Frequently used content**: Activity recorded in the last week, and in nine of the last 12 months.
- **Actively used content**: Activity recorded in the past month.
- **Occasionally used content**: Activity recorded in the last nine months, but without activity in the past one month.
- **Unused content**: No activity recorded in the last nine months.

User type classification

Consider the following classifications for *user type*.

- **Content creator**: Activity recorded in the past six months that created, published, or edited content.
- **Content viewer**: Activity recorded in the past six months that viewed content, but without any content creation activity.

Consider recency vs. trends

You should decide whether the usage classifications for users or content should be based only on how *recently* an activity occurred. You might want to also consider factoring in *average* or *trending* usage over time.

Consider some examples that demonstrate how simple classification logic might misrepresent reality.

- A manager viewed one report this week. However, prior to that week, the manager hadn't viewed any reports in the last six months. You shouldn't consider this manager to be a frequent user based on recent usage alone.
- A report creator publishes a new report every week. When you analyze usage by frequent users, the report creator's regular activity appears to be positive. However, upon further investigation you discover that this user has been republishing a new report (with a new report name) every time they edit the report. It would be useful for the report creator to have more training.
- An executive views a report sporadically, and so their usage classification changes frequently. You might need to analyze certain types of users, such as executives, differently.
- An internal auditor views critical reports once per year. The internal auditor might appear to be an inactive user because of their infrequent usage. Someone might take steps to remove their Pro or PPU license. Or, someone might believe that a report should be retired since it's used infrequently.

> **Tip**
>
> You can calculate averages and trends by using the DAX time intelligence functions. To learn how to use these functions, work through the **Use DAX time intelligence functions in Power BI Desktop models** learning module.

Checklist – When creating classifying usage data, key decisions and actions include:

- ✓ **Get consensus on classification definitions**: Discuss classification definitions with the relevant stakeholders. Make sure there's agreement when making the decisions.
- ✓ **Create documentation**: Ensure that the classification definitions are included in documentation for content creators and consumers.
- ✓ **Create a feedback loop**: Make sure there's a way for users to ask questions or propose changes to the classification definitions.

Create analytical reports

At this point, the auditing data has been extracted and stored, and you've published a data model. The next step is to create analytical reports.

Focus on the key information that's most relevant for each audience. You might have several audiences for your auditing reports. Each audience will be interested in different information, and for different purposes. The audiences you might serve with your reports include:

- Executive sponsor
- Center of Excellence
- Power BI administrators
- Workspace administrators
- Premium capacity administrators
- Gateway administrators
- Power BI developers and content creators
- Auditors

Here are some of the most common analytical requirements that you might want to start with when you create your auditing reports.

- **Top content views**: Your executive sponsor and leadership teams might predominantly be interested in summary information and trends over time. Your workspace administrators, developers, and content creators will be more interested in the details.
- **Top user activities**: Your Center of Excellence will be interested in who's using Power BI, how, and when. Your Premium capacity administrators will be interested in who's using the capacity to ensure its health and stability.
- **Tenant inventory**: Your Power BI administrators, workspace administrators, and auditors will be interested in understanding what content exists, where, lineage, and its security settings.

This list isn't all-inclusive. It's intended to provide you with ideas about how to create analytical reports that target specific needs. For more considerations, see Data needs earlier in this article, and Auditing and monitoring overview. These resources include many ideas for how you can use auditing data, and the types of information you might choose to present in your reports.

> **Tip**
>
> While it's tempting to present a lot of data, only include information that you're prepared to act on. Ensure that every report page is clear about its purpose, what action should be taken, and by whom.
>
> To learn how to create analytical reports, work through the **Design effective reports in Power BI** learning path.

Checklist - When planning for analytical auditing reports, key decisions and actions include:

- ✓ **Review requirements**: Prioritize creating reports based on known needs and specific questions that should be answered.
- ✓ **Confirm your audience(s)**: Clarify who'll use the auditing reports, and what their intended purpose will be.
- ✓ **Create and deploy reports**: Develop the first set of core reports. Extend and enhance them gradually over time.
- ✓ **Distribute reports in an app**: Consider creating an app that includes all your auditing and monitoring reports.

- ✓ **Verify who has access to reports**: Ensure that the reports (or the app) are made available to the correct set of users.
- ✓ **Create a feedback loop**: Make sure there's a way for report consumers to provide feedback or suggestions, or report issues.

Take action based on the data

Auditing data is valuable because it helps you to understand what's happening in your Power BI tenant. While it might seem obvious, explicitly acting on what you learn from the audit data can be easily overlooked. For that reason, we recommend that you assign someone who's responsible for tracking measurable improvements, rather than just reviewing auditing reports. That way, you can make gradual, measurable advances in your adoption and level of maturity with Power BI.

You can take many different actions based on your goals and what you learn from the auditing data. The remainder of this section provides you with several ideas.

Content usage

Here are some actions you might take based on how content is used.

- **Content is frequently used (daily or weekly)**: Verify that any critical content is certified. Confirm that ownership is clear, and the solution is adequately supported.
- **High number of workspace views**: When a high number of workspace views occur, investigate why Power BI apps aren't in use.
- **Content is rarely used**: Contact the target users to determine whether the solution meets their needs, or whether their requirements have changed.
- **Refresh activity occurs more frequently than views**: Contact the content owner to understand why a semantic model is refreshed regularly without any recent use of the semantic model or related reports.

User activities

Here are some actions you might take based on user activities.

- **First publishing action by a new user**: Identify when a user type changes from consumer to creator, which you can identify when they publish content for the first time. It's a great opportunity to send them a standard email that provides guidance and links to useful resources.
- **Engagement with the most frequent content creators**: Invite your most active creators to join your champions network, or to get involved with your community

of practice.

- **License management**: Verify whether inactive creators still need a Pro or PPU license.
- **User trial activation**: A trial license activation can prompt you to assign a permanent license to the user before their trial ends.

User training opportunities

Here are some user training opportunities that you might identify from the auditing data.

- **Large number of semantic models published by the same content creator**: Teach users about shared semantic models and why it's important to avoid creating duplicate semantic models.
- **Excessive sharing from a personal workspace**: Contact a user who's doing a lot of sharing from their personal workspace. Teach them about standard workspaces.
- **Significant report views from a personal workspace**: Contact a user who owns content that has a high number of report views. Teach them how standard workspaces are better than personal workspaces.

> **Tip**
>
> You can also improve your training content or documentation by reviewing questions answered by your **internal Power BI community** and issues submitted to the **help desk**.

Security

Here are some security situations you might want to actively monitor.

- Too many users assigned to the high-privilege Fabric administrator role.
- Too many workspace administrators (when the Member, Contributor, or Viewer workspace role would be sufficient).
- Excessive Build permissions assigned to semantic models (when Read permission would be sufficient).
- High use of per-item permissions, when Power BI app permissions or the workspace Viewer role would be a better choice for content consumers.
- How content is shared with external users.

For more information, see the Security planning articles.

Governance and risk mitigation

Here are some situations that you might encounter. Consider explicitly looking for these types of situations in your auditing reports, so you're prepared to act quickly.

- High number of views for reports (and underlying semantic models) that aren't endorsed.
- Significant use of unknown or unsanctioned data sources.
- File locations that don't align with governance guidelines for where source files should be located.
- Workspace names don't align with governance requirements.
- Sensitivity labels are used for information protection.
- Consistent data refresh failures.
- Significant and recurring use of printing.
- Unexpected or excessive use of subscriptions.
- Unexpected use of personal gateways.

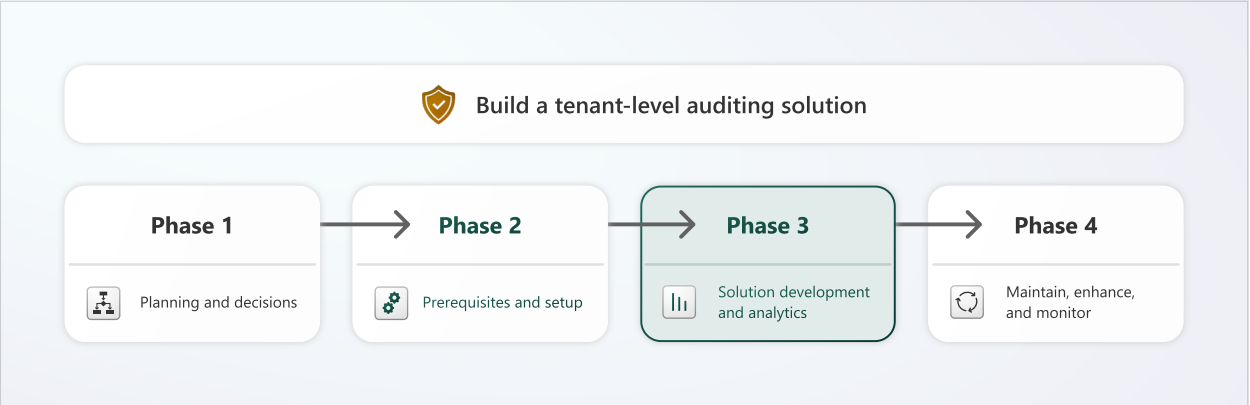
The specific actions to be taken in each situation will depend on your governance policies. For more information, see Governance in the Fabric adoption roadmap.

Checklist - When planning for potential actions based on the auditing data, key decisions and actions include:

- ✓ **Clarify expectations**: Create auditing reports with a clear set of expectations for what actions are expected.
- ✓ **Clarify who'll be responsible for actions**: Ensure that roles and responsibilities are assigned.
- ✓ **Create automation**: When possible, automate known actions that are repeatable.
- ✓ **Use a tracking system**: Use a system to track an observed situation, including contact made, next planned action, who's responsible, resolution, and status.

Phase 4: Maintain, enhance, and monitor

The fourth phase of planning and implementing a tenant-level auditing solution focuses on maintenance, enhancements, and monitoring. At this point, your auditing solution is in production use. You're now primarily concerned with maintaining, enhancing, and monitoring the solution.



Operationalize and improve

Auditing processes are typically considered to be running *in production* when initial development and testing are complete and you've automated the process. Automated auditing processes running in production have greater expectations (than manual processes) for quality, reliability, stability, and support.

A production-level auditing process has been *operationalized*. An operationalized solution commonly includes many of the following characteristics.

- **Secure**: Credentials are stored and managed securely. Scripts don't contain passwords or keys in plaintext.
- **Scheduling**: A reliable scheduling system is in place.
- **Change management**: Use of change management practices and multiple environments are used to ensure that the production environment is safeguarded. You might also work with development and test environments, or just a development environment.
- **Support**: The team that supports the solution is clearly defined. Team members have been trained, and they understand the operational expectations. Backup members have been identified and cross-training happens when appropriate.
- **Alerting**: When something goes wrong, alerts notify the support team automatically. Preferably, alerting includes both logs and email (rather than email only). A log is useful for analyzing logged errors and warnings.
- **Logging**: Processes are logged so there's a history of when the auditing data was updated. Logged information should record start time, end time, and the identity of user or app that ran the process.
- **Error handling**: Scripts and processes gracefully handle and log errors (such as whether to exit immediately, continue, or wait and try again). Alert notifications are sent to the support team when an error occurs.
- **Coding standards**: Good coding techniques that perform well are used. For example, loops are purposefully avoided except when necessary. Consistent coding

standards, comments, formatting, and syntax are used so that the solution is easier to maintain and support.

- **Reuse and modularization**: To minimize duplication, code and configuration values (like connection strings or email addresses for notifications) are modularized so that other scripts and processes can reuse them.

> **Tip**
>
> You don't have to do everything listed above all at once. As you gain experience, you can incrementally improve the solution so that it becomes complete and robust. Be aware that most examples you find online are simple, one-off script snippets that might not be production quality.

Checklist - When planning to operationalize and improve an auditing solution, key decisions and actions include:

- ✓ **Assess the level of existing solutions**: Determine whether there are opportunities to improve and stabilize existing auditing solutions that are automated.
- ✓ **Establish production-level standards**: Decide what standards you want to have for your automated auditing processes. Factor in improvements that you can realistically add over time.
- ✓ **Create a plan for improvement**: Plan to improve the quality and stability of production auditing solutions.
- ✓ **Determine whether a separate development environment is necessary**: Assess the level of risk and reliance on the production data. Decide when to create separate development and production (and test) environments.
- ✓ **Consider a data retention strategy**: Determine whether you need to implement a data retention process that purges data after a certain period of time, or upon request.

Documentation and support

Documentation and support are critical for any production-level solution. It's helpful to create several types of documentation, depending on the target audience and purpose.

Technical documentation

Technical documentation is targeted at the technical team who built the solution and who will gradually improve and expand the solution over time. It's also a useful resource for the support team.

Technical documentation should include:

- A summary of architecture components and prerequisites.
- Who owns and manages the solution.
- Who supports the solution.
- An architecture diagram.
- Design decisions, including goals, reasons why certain technical choices were made, and constraints (such as cost or skills).
- Security decisions and choices.
- Naming conventions used.
- Coding and technical standards and guidelines.
- Change management requirements.
- Deployment, setup, and installation instructions.
- Known areas of technical debt (areas that can be improved if there's opportunity to do so).

Support documentation

Depending on the level of criticality for your auditing solution, you might have a help desk or support team should urgent issues arise. They might be available all day, every day.

Support documentation is sometimes referred to as a *knowledge base* or a *runbook*. This documentation is targeted at your help desk or support team, and it should include:

- Troubleshooting guidance for when something goes wrong. For example, when there's a data refresh failure.
- Actions to take on an ongoing basis. For example, there could be some manual steps that someone needs to do regularly until an issue is resolved.

Content creator documentation

You derive more value from your auditing solution by providing usage and adoption analytics to other teams throughout the organization (with row-level security enforced, if necessary).

Content creator documentation is targeted at self-service content creators who create reports and data models that source the curated auditing data. It includes information

about the data model, including common data definitions.

Checklist – When planning for documentation and support for your auditing solution, key decisions and actions include:

- ✓ **Confirm who's expected to support the solution**: Determine who'll support auditing solutions that are considered production-level (or have downstream report dependencies).
- ✓ **Ensure support team readiness**: Verify that the support team is prepared to support the auditing solution. Identify whether there are any readiness gaps that need addressing.
- ✓ **Arrange for cross-training**: Conduct knowledge transfer sessions or cross-training sessions for the support team.
- ✓ **Clarify support team expectations**: Ensure that expectations for response and resolution are clearly documented and communicated. Decide whether anyone needs to be on call to quickly resolve issues related to the auditing solution.
- ✓ **Create technical documentation**: Create documentation about the technical architecture and design decisions.
- ✓ **Create support documentation**: Update the help desk knowledgebase to include information about how to support the solution.
- ✓ **Create documentation for content creators**: Create documentation to help self-service creators use the data model. Describe common data definitions to improve consistency of their use.

Enable alerting

You might want to raise alerts based on specific conditions in the auditing data. For example, you can raise an alert when someone deletes a gateway or when a Power BI administrator changes a tenant setting.

For more information, see Tenant-level monitoring.

Ongoing management

You need to perform ongoing management of the entire auditing solution. You might need to extend or change your auditing solution when:

- The organization discovers new data requirements.

- New audit events appear in the raw data you exact from the Power BI REST APIs.
- Microsoft makes changes to the Power BI REST APIs.
- Employees identify ways to improve the solution.

> **Important**
>
> Breaking changes are rare, but they can occur. It's important that you have someone available who can quickly troubleshoot issues or update the auditing solution when necessary.

Checklist - When planning for ongoing management of the auditing solution, key decisions and actions include:

- ✓ **Assign a technical owner**: Ensure that there's clear ownership and responsibility for the entire auditing solution.
- ✓ **Verify that a backup exists**: Make sure there's a backup technical owner who can get involved should an urgent issue arise that support can't solve.
- ✓ **Keep a tracking system**: Ensure that you have a way to capture new requests and a way to prioritize immediate priorities, and also short-term, medium-term, and long-term (backlog) priorities.

Related content

In the next article in this series, learn about tenant-level monitoring.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Tenant-level monitoring

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This tenant-level monitoring article is primarily targeted at:

- **Power BI administrators**: The administrators who are responsible for overseeing Power BI in the organization. Power BI administrators might need to collaborate with IT, information security, internal audit, and other relevant teams.
- **Center of Excellence, IT, and BI team**: The teams that are also responsible for overseeing Power BI. They might need to collaborate with Power BI administrators, information security, and other relevant teams.

The terms *auditing* and *monitoring* are closely related.

- **Auditing**: Actions you take to understand a system, its user activities, and related processes. Auditing activities can be manual, automated, or a combination of both. An auditing process might focus on one specific aspect (for example, auditing security for a workspace), or it can refer to an end-to-end auditing solution. An auditing solution consists of extracting, storing, and transforming data so that it can be analyzed and acted upon.
- **Monitoring**: Ongoing activities that inform you about what's occurring. Monitoring usually involves alerting and automation, though sometimes monitoring is done manually. Monitoring can be set up for a process you've selected to audit (for example, notifications that appear when a specific tenant setting changes).

This article describes the different ways you can monitor a Power BI tenant.

> **Note**
>
> For more information about tracking activities that users perform, see Tenant-level auditing.

Monitor and protect content

There are several monitoring aspects related to information protection and data loss prevention.

Information protection for Power BI

Information protection focuses on safeguarding organizational data, reducing the risk of sharing sensitive information, and strengthening compliance status for regulatory requirements. Information protection begins with the use of sensitivity labels.

When you classify and label content, it helps the organization to:

- Understand where sensitive data resides.
- Track external and internal compliance requirements.
- Protect content from unauthorized users.
- Educate users on how to responsibly handle data.
- Implement real-time controls to reduce the risk of data leakage.

> **Tip**
>
> For information about implementing sensitivity labels, see **Information protection for Power BI**.

Data loss prevention for Power BI

Data loss prevention (DLP) refers to activities and practices that safeguard data in the organization. The goal for DLP is to reduce the risk of data leakage, which can happen when unauthorized people share sensitive data. While responsible user behavior is a critical part of safeguarding data, DLP usually refers to policies that are automated.

DLP allows you to:

- Detect and inform administrators when risky, inadvertent, or inappropriate sharing of sensitive data occurs. Specifically, it allows them to:
  - Improve the overall security setup of your Power BI tenant, with automation and information.
  - Enable analytical use cases that involve sensitive data.
  - Provide auditing information to security administrators.
- Provide users with contextual notifications. Specifically, it supports them to:
  - Make the right decisions during their normal workflow.

- Adhere to your data classification and protection policy, without negatively affecting their productivity.

> **Tip**
>
> For information about implementing DLP, see **Data loss prevention for Power BI**.

Monitor security and threats

There are several monitoring aspects related to security and threats.

Defender for Cloud Apps for Power BI

Microsoft Defender for Cloud Apps is a cloud access security broker (CASB) that allows administrators to:

- Monitor and raise alerts based on specific activities.
- Create DLP policies.
- Detect unusual behaviors and risky sessions.
- Limit activities performed by applications (with Microsoft Entra Conditional Access app control).

Some powerful Power BI monitoring and protection capabilities are available with Defender for Cloud Apps. For example, you can:

- Prohibit all—or certain users—from downloading a file from the Power BI service when a specific sensitivity label is assigned.
- Receive an alert when an administrative activity is detected, such as when a tenant setting is updated in the Power BI service.
- Detect when suspicious or unusual behaviors occur, such as large file downloads or an unusual number of sharing operations in the Power BI service.
- Search the activity log for specific activities relating to content with a specific sensitivity label, such as data exports from the Power BI service.
- Receive notifications as risky sessions occur, such as when the same user account connects from different geographical areas in a short time period.
- Determine when someone outside of a predefined security group views specific content in the Power BI service.

For more information, see Defender for Cloud Apps for Power BI.

Microsoft Sentinel

Microsoft Sentinel is a security information and event management (SIEM) service. It's an Azure service that allows you to:

- Collect logs and security data for users, devices, applications, and infrastructure. You can capture logs and user activities from the Power BI service and many other areas across the enterprise.
- Detect potential threats. You can create rules and alerts to refine what's important to track. Automated threat intelligence exists as well, which can help reduce the manual effort.
- Analyze data and investigate the scope and root cause of incidents. You can use built-in visualization tools or Kusto Query Language (KQL) queries. You can also use Power BI to visualize and analyze the data that you collect.
- Respond to security incidents and threats. You can handle responses directly. Or, you can automate responses and remediations by using playbooks, which are workflows based on Azure Logic Apps.

Microsoft Sentinel stores its data in Azure Log Analytics (a component of Azure Monitor). It's based on the same architecture as the semantic model event logs, which capture user-generated and system-generated activities that occur for a semantic model.

You can use Microsoft Sentinel with Power BI in two different ways.

- **Use the Power BI data connector in Sentinel**: A subset of the attributes from the Power BI audit logs are streamed into Azure Log Analytics (Azure Monitor). It's one way to obtain audit logs for tracking user activities in your Power BI environment. For more information, see Tenant-level auditing.
- **Use Power BI as an analytical tool**: Power BI connects to the data that Microsoft Sentinel (and, accordingly, Azure Monitor and Azure Log Analytics) collects from a wide variety of data connectors. You can then use standard Power BI functionality to model, analyze, and visualize data. For more information, see Create a Power BI report from Microsoft Sentinel data.

> **ⓘ Important**
>
> Microsoft Sentinel, Azure Monitor, Azure Log Analytics, and Defender for Cloud Apps are separate services. They have their own pricing models and security models, which are separate from Power BI. Power BI administrators don't automatically have access to these services. We recommend that you work with your infrastructure team to plan which services are best to use.

For more information about the different options for capturing user activities, see Tenant-level auditing.

Monitor Power BI service health

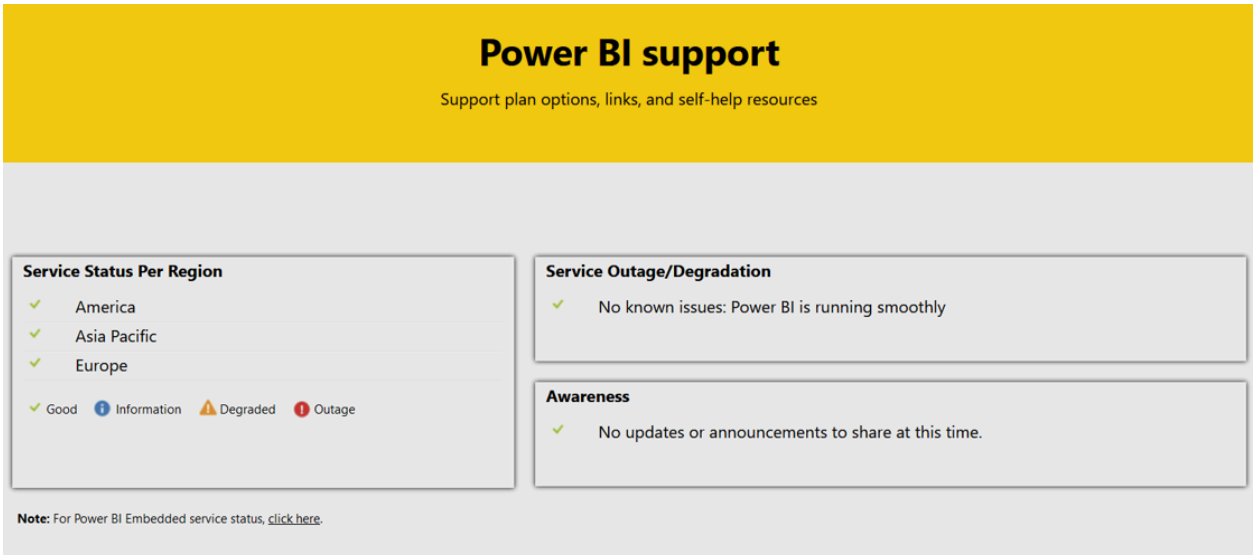
There are several ways to obtain information about service health, incidents, and issues.

> **Tip**
>
> Before you submit a Power BI support request to Microsoft, we recommend that you first check with the resources listed in this section.

Power BI support site

When there's an apparent service outage or degradation, Power BI administrators and users should refer to the Power BI support site. It's a publicly available site that displays information about high-priority issues concerning Power BI. This site shows:

- The status for the Power BI service.
- Service level outages or degradation notifications.
- Informational messages for broad awareness.



> **Note**
>
> Microsoft typically communicates issues related to the national/regional clouds in the Microsoft 365 admin center rather than the Power BI support site. If you work with national/regional clouds, work with your Microsoft 365 administrator to monitor Power BI issues.

For more information about Power BI support, see How to contact support.

For more information about how to support users in your organization, see User support.

Power BI email notifications

You can receive alert notifications by email to inform you when there's a service outage, interruption, or degradation occurring in your Power BI tenant. These notifications are available only for Premium workspaces.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

To set up email alerts, enable the *Receive email notifications for service outages or incidents* tenant setting. Because its purpose is to send email, you must assign a mail-enabled security group to this setting. We recommend that you use a group name like *Power BI System Support*. You should add your Power BI administrators, key personnel from your Center of Excellence (COE), and your help desk that handles user support to this group.

> **Tip**
>
> When you need to notify your internal users, we recommend that your COE sends a customized message that uses non-technical language. That way, the message can include additional context, and use the preferred communication platform, like a Teams channel.

For more information, see Enable service interruption notifications.

Power BI known issues

Power BI administrators and users can also monitor the Fabric known issues page. This page includes information about currently active known issues and recently closed

known issues.

Known issues can include software bugs that have been reported to Microsoft support by other customers. An issue might also include functionality that's by design, but because Microsoft Support has received a significant number of tickets, an explanation is warranted.

Microsoft 365 admin center

The Microsoft 365 admin center displays service health information, incident summaries, and advisory messages for Microsoft 365 services. Also, service incidents and certain types of major updates for the Power BI service are posted in the Microsoft 365 admin center.

> **Tip**
>
> Access to M365 is available to administrators who have sufficient **permissions**. Power BI administrators have a limited view to the M365 service health and the M365 message center.

There are two types of message.

- **Advisory message**: An issue is affecting only some customers. The service is available, however the issue could be intermittent or limited in scope and user impact.
- **Active incident**: An issue is presently causing the service, or a major function, to be unavailable or severely degraded for multiple customers.

Microsoft 365 Service health

The Microsoft 365 service health page shows notifications about advisories and incidents. It also has information about active incidents, including:

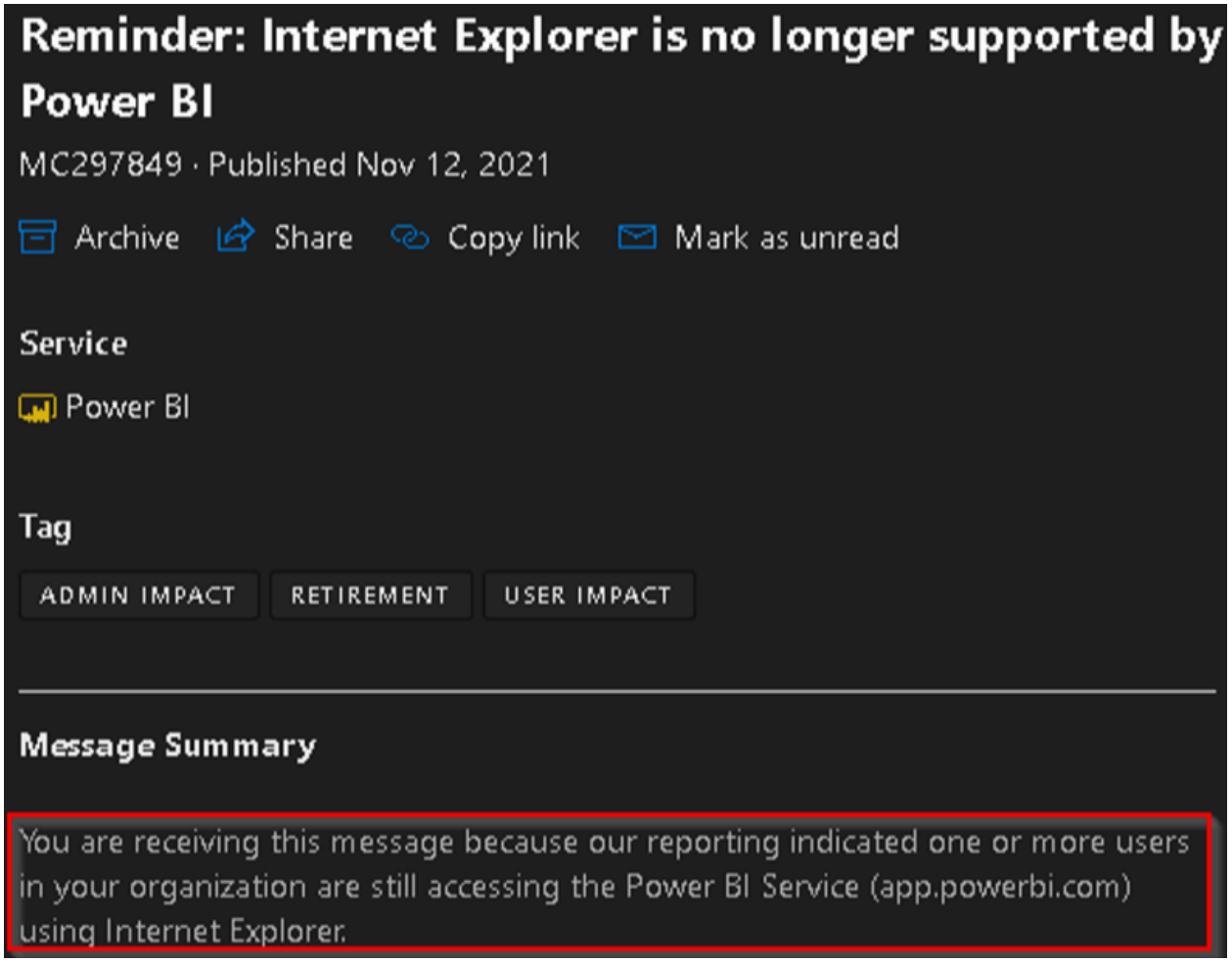
- Description
- User impact
- Status
- Duration (if closed) or estimated time to resolve (if open)
- Next status update (if open)
- Root cause (if closed)

The issue history page shows incidents and advisories that were resolved in the past 30 days.

Microsoft 365 message center

The Microsoft 365 message center publishes planned changes to Microsoft 365 services, allowing administrators to prepare in advance. For active incidents, each message links to more details in the Microsoft 365 service health page, as previously described.

Sometimes, a message is based on telemetry gathered by Microsoft. For example, Microsoft's telemetry knows which type of browser users are using to connect to the Power BI service. If Internet Explorer use is detected, you might receive a message reminding you that Power BI no longer supports that browser.



Power BI issues site

You can check the publicly available Power BI issues site. It's a place for users to publicly report issues they've encountered.

Monitor updates and fixes

By managing how Power BI Desktop is installed on user machines, you can control when updates and fixes are installed. However, you can't control when changes to the Power

BI service are released in your tenant.

We recommend that you closely monitor releases for the Power BI service and Power BI Desktop. Monitoring releases ensures that you can be prepared, test key feature changes, and announce important changes to your users.

Updates

The Power BI service is a cloud service that's updated continually and frequently. Power BI Desktop, which must be installed on a Windows machine, is usually updated monthly (except for fixes, which are described next). Microsoft posts release announcements on the Power BI blog.

For information about version numbers and links to more information for the current release, see What's new in Power BI.

For information about previous releases, see Power BI updates archive.

QFE releases

Depending on the severity, Microsoft might do a quick-fix engineering (QFE) release, which is commonly known as a *bug fix* or *hotfix*. QFE releases occur when Power BI Desktop updates are made outside of the normal monthly release cadence.

For history of previous QFE releases, see Change log for Power BI Desktop.

Monitor Power BI announcements

To effectively support Power BI in your organization, you should continually watch for announcements and new features.

Power BI release plan

You can find the public roadmap for future features including estimated dates in the release plan.

Sometimes a change that's coming is so important that you'll want to plan for it in advance. The planning cycle is divided into two semesters each year: April through September, and October through March.

Power BI blog

Subscribe to the Power BI blog to follow posts about important public announcements and new releases. Some blog posts also provide information about upcoming features that can help you to plan ahead.

> **Tip**
>
> It's particularly important to read the monthly blog post announcement. It includes a summary of new features and planned changes to the Power BI service, Power BI Desktop, and the Power BI mobile apps.

Power BI ideas

Consider routinely monitoring the Power BI ideas site. This site informs you about top ideas that other customers have requested. You can also influence the future direction of Power BI by submitting new ideas and voting for posted ideas that you want to support.

Monitor related Azure services

The Azure status page shows the status for Azure services. There are many Azure services that might potentially integrate with your Power BI tenant.

Common Azure services that integrate with Power BI include:

- **Microsoft Entra ID**: Your Power BI tenant relies on Microsoft Entra ID for identity and access management.
- **Azure Power BI Embedded**: Azure Power BI Embedded supports programmatic embedding of Power BI content in apps for your customers. Power BI Embedded is also applicable for customers who have enabled autoscale for their Power BI Premium capacity. For more information about when to use Power BI Embedded, see the embed for your customers usage scenario.
- **Azure storage accounts**: Azure Data Lake Storage Gen2 (ADLS Gen2) can be used for workspace-level data storage, including dataflows storage and semantic model backups. For more information about dataflow storage, see the self-service data preparation usage scenario.
- **Azure Log Analytics**: You can enable workspace auditing to capture semantic model event logs. For more information, see Data-level auditing.
- **Azure Files**: When the large semantic model format is enabled for a workspace, the data is stored in Azure Files.

- **Data sources**: It's likely that you have many types of data sources that Power BI connects to. Data sources could be Azure Analysis Services, Azure SQL Database, Azure Synapse Analytics, Azure storage, and others.
- **Virtual machines**: A data gateway for Power BI could run on a virtual machine (VM) in Azure. Or, a database containing data that's used as a data source for Power BI might run on a VM in Azure.
- **Virtual network data gateway**: A virtual network (VNet) data gateway could be implemented to securely access data sources in a private network.
- **Azure Key Vault**: One common way to use Azure Key Vault is for customer management of the encryption keys for data at-rest in the Power BI service. For more information, see Bring your own encryption keys for Power BI.
- **Microsoft Purview**: Used by Microsoft Purview Information Protection, or by Microsoft Purview Data Catalog to scan your Power BI tenant, to extract metadata.

Checklist - When planning for tenant-level monitoring, key decisions and actions include:

- ✓ **Educate administrators and key personnel**: Make sure that Power BI administrators and key personnel in the COE are aware of the resources available for monitoring service health, updates, and announcements.
- ✓ **Create a monitoring plan**: Determine how, and who will monitor service health, updates, and announcements. Ensure that expectations are clear for how to gather, communicate, plan, and act on the information.
- ✓ **Create a user communication plan**: Clarify which situations warrant communicating to others in the organization. Determine how, and who will be responsible for communicating to users in the organization, and in which circumstances.
- ✓ **Decide who should receive email notifications**: Determine who should receive email notifications from Microsoft when there's a Power BI issue. Update the *Receive email notifications for service outages or incidents* tenant setting to align with your decision.
- ✓ **Review administrator roles**: Review roles and permissions necessary for viewing service health in the M365 admin center.
- ✓ **Investigate information protection and DLP requirements**: Explore requirements for using sensitivity labels in Microsoft Purview Information Protection to classify data (the first building block of information protection). Consider requirements for implementing DLP for Power BI, and the associated monitoring processes.

- ✓ **Investigate Defender for Cloud Apps capabilities**: Explore requirements for using Microsoft Defender for Cloud Apps to monitor user behavior and activities.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI implementation planning: Adoption tracking

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This article helps you to plan how to track adoption of Power BI and Microsoft Fabric. This article is primarily targeted at:

- **BI and analytics directors and managers**: Decision makers who are responsible for overseeing the BI program and strategy. Successful adoption of Fabric and Power BI is essential for these decision makers to make progress towards the BI strategy.
- **Fabric administrators**: The administrators who are responsible for overseeing Fabric in the organization. Fabric administrators use the admin monitoring workspace and Fabric Capacity Metrics app to aid in adoption tracking and capacity management. Fabric administrators might need to collaborate with other administrators, like those who oversee Microsoft 365.
- **Center of Excellence (COE), IT, and BI teams**: The teams that are responsible for overseeing Power BI in the organization. These teams are responsible for promoting a healthy data culture and improving organizational adoption.
- **Subject matter experts (SMEs), content owners and content creators**: The teams and individuals that champion analytics in a team or department and who create and consume Power BI or Fabric content.

> **Tip**
>
> We recommend that you familiarize yourself with the Microsoft Fabric adoption roadmap series before you read this article about adoption tracking. The adoption roadmap introduces concepts that are important to adoption and explains how to define your organization's adoption maturity levels. After you've read the adoption roadmap, you can then use this article to plan how you'll track progress towards incrementally improving these maturity levels.

Adoption is essential to achieve successful implementation of Fabric and Power BI. Adoption isn't about whether people are using Fabric and Power BI, but instead whether they're using these tools *effectively*. You can think about adoption as empowering *more people* to use the *right data and tools* the *right way* at the *right time*.

Adoption isn't a state that you achieve. Rather, adoption is a continuous journey where you deliver sustainable and incremental progress in different areas that together empower people to work effectively with data. The result of adoption is that more people realize increased value from their data by using Fabric and Power BI. When empowered in this way, people work to achieve their key results and help the organization to accomplish its strategic business objectives.

To identify whether you're improving adoption, you perform *adoption tracking*. Like adoption itself, adoption tracking isn't only about tracking usage. Rather, adoption tracking consists of the activities and mechanisms that you use to identify and understand how people use data and tools, and whether they're using them effectively. To perform adoption tracking activities, you should identify and understand both negative user behaviors that you want to discourage and positive user behaviors that you want to replicate. You next define ways to reliably measure these behaviors and track them over time. You improve the maturity levels of your data culture and organizational adoption when you experience more positive behaviors and less negative behaviors. As you observe and measure this evolution in user behavior, you're achieving incremental progress toward your adoption goals.

The first part of this article helps you to understand the benefits and approaches to track adoption. The second part then explains how to track adoption of Fabric and Power BI.

> **ⓘ Important**
>
> Adoption tracking is essential to enable the success of both **self-service** and **enterprise BI**.

Understand why adoption tracking is important

Adoption tracking is a fundamental capability to advance organizational, user, and solution adoption.

- *Organizational adoption* tracks the state of analytics governance and data management practices.

- *User adoption* tracks the extent to which content consumers and self-service content creators are actively and effectively using analytics tools such as Fabric.
- *Solution adoption* tracks the impact of content that's been deployed. It's also concerned with measuring the level of value solutions provide.

By tracking adoption, you can identify strategic areas of improvement that you should prioritize. These priorities inform the work of your Fabric and Power BI teams, other data teams, and the business teams that they support. For this reason, it's important that you perform adoption tracking effectively. Effective adoption tracking means that you:

- Define and maintain a shared understanding for what adoption means in your organization, including the current state of adoption and the desired future state.
- Identify strategic adoption objectives and priorities, which inform the behaviors that you want to promote or encourage (or behaviors that you want to prevent and discourage).
- Define a mechanism to reliably track and measure these important behaviors and their evolution over time.
- Assign ownership of the tracking to an individual or team.
- Act upon what you learn from the tracking to steer improvement over time.

Effective adoption tracking provides both strategic and tactical perspectives of the current state of the organization and its use of data and analytics. As such, adoption tracking is closely tied to both your strategic and tactical planning.

- **Strategic**: Your strategic focus areas and objectives typically define what you want to achieve with your adoption, and which behaviors are important to make progress toward those objectives. What you learn from adoption tracking helps inform long-term strategic decisions.
- **Tactical**: Your key results often require measures that you obtain and monitor by performing adoption tracking. What you learn from adoption tracking helps inform short-term tactical actions to help you achieve your strategic objectives.

> **Tip**
>
> Adoption and adoption tracking are important for your BI strategy. For more information about what a BI strategy is, how it relates to your business strategy, and the steps you can take to define a BI strategy, see **BI strategy overview**.

The benefits of adoption tracking

Adoption tracking yields benefits for many different teams in your organization. To summarize, adoption tracking aids in:

- Visibility and understanding of how people in the organization use Fabric and Power BI. This visibility helps you to preemptively mitigate risk and identify opportunities to help people use data and tools more effectively.
- Clarity about which activities should be prioritized to make progress toward your objectives. This prioritization also helps you to plan and allocate resources.
- Measurements that inform feedback loops and demonstrate a return on investment (ROI) in your implemented analytics solutions and initiatives. This return on investment can help you to obtain executive sponsorship and secure more budget for future data and analytics projects.

> **Note**
>
> To help you understand the importance and benefits of effective adoption tracking, consider the following analogy.
>
> *If your BI strategy is a moving vehicle that your organization drives, then adoption tracking is the windshield of that vehicle.*
>
> - A *clean windshield* is what you get with effective adoption tracking. A clean windshield allows you to see where you are and where you're going. You see through this windshield to understand your motion in relation to what's around you. With a clean windshield, you can avoid obstacles and steer toward opportunities.
> - A *dirty windshield* is what you get when you neglect or improperly perform adoption tracking. A dirty windshield prevents you from seeing where you are and where you're going, and you're a danger to yourself and to others. With a dirty windshield, you're less likely to safely reach your desired destination.

Different approaches to perform adoption tracking

Adoption tracking requires that you gather data to measure and track indicators over time by using different solutions. You can perform adoption tracking by using either custom or ready-made solutions, or a combination of the two to meet the needs of your organization.

Custom solutions

You can build your own custom solution for adoption tracking. Custom solutions are similar to a tenant-level auditing solution. A custom adoption tracking solution requires the highest effort to create and maintain. That's because you must gather the data and pass it through an end-to-end analytics pipeline before you can effectively use it. However, once completed, a custom solution has the greatest potential to address your specific needs and questions.

> **ⓘ Important**
>
> Building a custom adoption tracking solution can involve significant effort because it's an end-to-end BI project. Ensure that you first evaluate the available ready-made solutions before you decide to develop your own solution.

When you build a custom solution, you gather data from various sources such as the following.

- **Audit log**: Provides a detailed view of all of the Microsoft Fabric user activities from the Microsoft 365 admin portal in an audit log. You can also use the Power BI activity log, which shows only the Power BI activities, and is based on the Power BI REST APIs. For information about how to get access to this activity log, see Access the Power BI activity log. You can access the activity log programmatically by using the ActivityEvents REST API.
- **Gateway logs**: Provide detailed information about data gateway configuration and operations. You can gather the logs from the on-premises data gateway app for on-premises data gateways, and the Fabric portal for virtual network (VNet) data gateways.
- **Azure Log Analytics**: Provides detailed information about telemetry, such as query performance, refresh operations, high levels of activity, or unusual activity. Workspace administrators can work with an Azure administrator to integrate Azure Log Analytics with a Fabric workspace to gather this detailed information.
- **REST APIs**: You can gather information by using various Power BI and Fabric REST APIs.
  - **Power BI REST APIs**: Get information about Power BI items (such as Dataflow refresh history), capacities, gateways, and more. The ActivityEvents REST API is particularly important to get user activities from the activity log. Activities are often an excellent data source to view and measure behaviors in your tenant.
  - **Fabric REST APIs**: Get information about Fabric items (such as Notebooks), domains, tenant settings, and more.
  - **Metadata scanning APIs**: Get detailed information about workspaces and their items. There are separate metadata scanning APIs for both Power BI and Fabric.

- **Microsoft Graph API**: Get user and group data, such as user attributes and licenses.

> **Note**
>
> For more detailed guidance about how to gather this type of data for a custom solution, see data source decisions from the tenant-level auditing article.

> **Tip**
>
> You can also enrich the **admin monitoring workspace** ready-made solution (described in the following section) with this information to extend its usefulness. For example, you can create a composite model by combining the activity information in the admin monitoring workspace shared semantic model with user and group data from Microsoft Graph.

Ready-made solutions

You can benefit from ready-made solutions for adoption tracking. These solutions are provided by Microsoft in your Fabric tenant, and you can use them with minimal effort and set up time. However, except for the content in the admin monitoring workspace, you have limited ability to customize and adapt ready-made solutions for your specific needs. For instance, most of these ready-made solutions provide a limited data history, such as the last 30 days.

> **Tip**
>
> Except for the **Fabric Capacity Metrics app**, all ready-made solutions are available for any tenant. The Fabric Capacity Metrics app is available only when your organization uses Fabric capacity.

Here are the ready-made solutions that you can use to support adoption tracking activities.

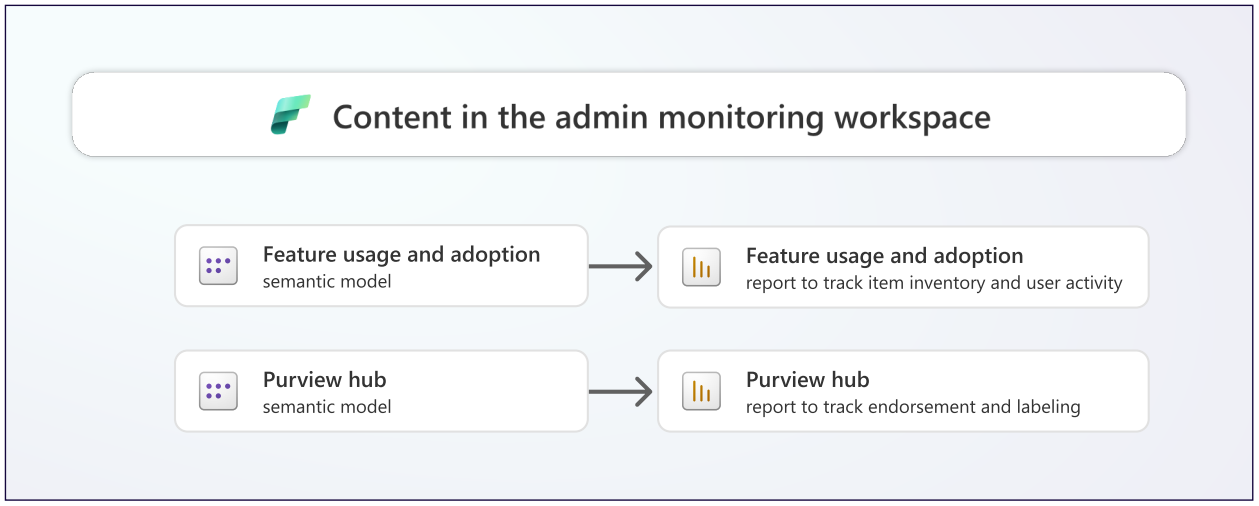
- **Monitoring hub**: Track information about data items that you have access to in your tenant. For example, this ready-made solution is one way to track average (or longest) refresh times, refresh failures, and refresh frequency by data item. You could use the Monitoring hub to identify the longest data item refresh and then work with the owner of that content to optimize and reduce Fabric capacity usage.

- **Power Platform admin center**: Track information about on-premises data gateway management. For example, this ready-made solution is one way to track how many gateways are installed, and who installed them. You could use the Power Platform admin center to identify unauthorized gateways, or gateways without an owner, so that they can be retired and archived.
- **Usage metrics reports**: Track information about report usage and workspace activities. For example, this ready-made solution is one way to track how many people are using your report, whether usage is rising or falling, and who the top users are. You could use the usage metrics report to identify whether usage of a critical report is falling so that you can plan office hours to support its users and drive better solution adoption.
- **Fabric Capacity Metrics app**: Track information about resource usage in your capacity to optimize and reduce the Capacity Units (CUs) used by different operations. The capacity metrics app is an essential tool not only to ensure the health of your capacity, but also for adoption tracking. When people use Power BI effectively, they avoid over-utilization of available capacity resources. You could use the capacity metrics app to identify resource-intensive items and operations and then work with the content owners to educate them about how to work toward more efficient solutions.
- **Admin monitoring workspace**: Track detailed information about user activities and inventories of the Fabric and Power BI items in your tenant. The admin monitoring workspace tracks a broader range of metrics and items than other ready-made solutions. Fabric administrators can also share this content with other users who aren't administrators. Furthermore, these administrators can grant Build permission to the underlying semantic models to allow others users to perform *ad hoc* analysis and to create new reports or composite semantic models that add new data or calculations.

> **Tip**
>
> This article includes the various ready-made solutions for completeness. To benefit most from this article, we recommend that you focus your time and effort on the admin monitoring workspace.

In this article, we focus on how to use the admin monitoring workspace to perform adoption tracking. With the admin monitoring workspace, organizations have a ready-to-use solution that they can also extend with other data. By using the admin monitoring workspace, teams and organizations of any size can immediately begin to conduct adoption tracking.

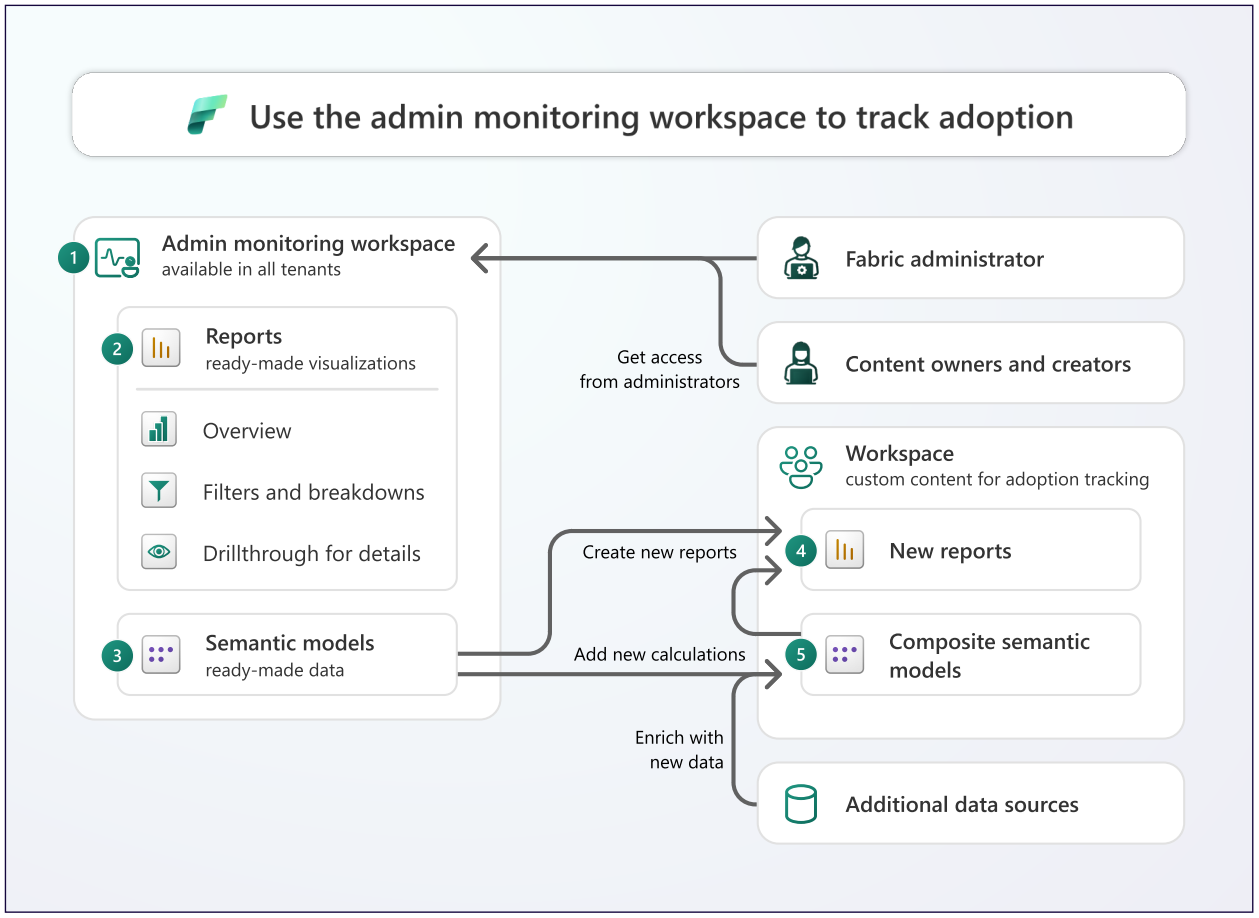
The following diagram and table provide an overview of the content in the admin monitoring workspace.



Expand table

Content	Description	Examples of questions that you can answer with this content
Feature usage and adoption	Provides insight into user activities from the activity log, and an inventory of the items in your tenant.	• How many items are there in a tenant or workspace, and which workspace has the most content? • How many data versus reporting items are there in a tenant or workspace, or how many items of a specific type (like dataflows)? • Which user is most frequently exporting reports to Excel? • Which items (or users) are inactive (they have no activity in the last 30 days)? • Which items are redundant across or within workspaces? • To what extent are Fabric items and features being used throughout the organization? Who's using these items, and where do they reside? • What items do external guest users have access to (from B2B sharing)?
Preview hub	Provides insight into content endorsement and sensitivity labels that have been applied.	• How many items are endorsed as Promoted or Certified? • Which workspaces contain endorsed or labeled content? • Which sensitivity labels are used for content, and how frequently?

The following diagram depicts how you can use the admin monitoring workspace to perform adoption tracking.



The diagram depicts the following components and processes.

Expand table

Item	Description
1	The admin monitoring workspace contains ready-made content available for all tenants to help with tracking and monitoring activities. Fabric administrators have access to this workspace and can grant workspace or item access to other content owners and creators, where necessary.
2	The admin monitoring workspace has reports that Fabric administrators and content owners and creators use for adoption tracking. These reports contain ready-made visualizations for overviews, filters, and breakdowns of different areas. The reports also support drillthrough to other pages to reveal details for a particular area-of-interest.
3	Reports have an underlying semantic model that Fabric administrators can connect to. These semantic models contain ready-made data and calculations. Fabric administrators can grant Build permission to content owners and creators so that they can also create new or custom tracking and monitoring content.
4	New reports created from the admin monitoring workspace semantic models are published to other workspaces that contain custom content for adoption tracking.

Item	Description
5	Composite semantic models integrate with other data to extend the usefulness of the original semantic models. They can also enrich the data with new calculation logic. The composite semantic models inform deeper, tailored insights into tracking and monitoring. Fabric administrators or content owners and creators can also create and publish new reports from these composite models.

To summarize, when you use the admin monitoring workspace to perform adoption tracking and monitoring of your tenant, you typically follow a stepwise process.

1. Get an overview of the situation from the reports.
2. Use filters and breakdowns to narrow the focus to important areas.
3. Drillthrough on categories of interest to investigate specific details.
4. Connect to the underlying semantic models to create your own reports.
5. Create composite models to extend the original semantic models with other data or calculation logic to meet your needs.

> **Tip**
>
> Consider your use of the admin monitoring workspace content analogous to how a user typically conducts self-service analytics. Use this content to better understand the user experience and flow from the perspective of a self-service user in your organization. In this scenario, *you* are the self-service user who consumes centralized reports and semantic models distributed by Microsoft.

The following section describes steps that you can take to track adoption in Fabric and Power BI.

Track adoption of Fabric and Power BI

To successfully track adoption, you first define your long-term objectives and the behaviors that reinforce them. You next identify indicators that can reliably measure these behaviors and can inform your short-term actions and activities. Importantly, you perform this tracking at a consistent cadence, regularly reevaluating your objectives and targets as behaviors change over time.

The following sections elaborate on the steps you take to track adoption of Fabric.

Step 1: Evaluate maturity to define your adoption objectives

To begin adoption tracking, you should first have clearly defined objectives that you want to achieve. That way, you can track progress towards them. One way to identify your adoption objectives is by performing a maturity assessment. In a maturity assessment, you evaluate your organization's current maturity level for different capability areas that are important to adoption (such as executive sponsorship or change management). Your maturity levels should accurately reflect how the organization uses Fabric and Power BI today. Performing a maturity assessment helps you to identify your objectives and where you'll need to prioritize your efforts to improve adoption.

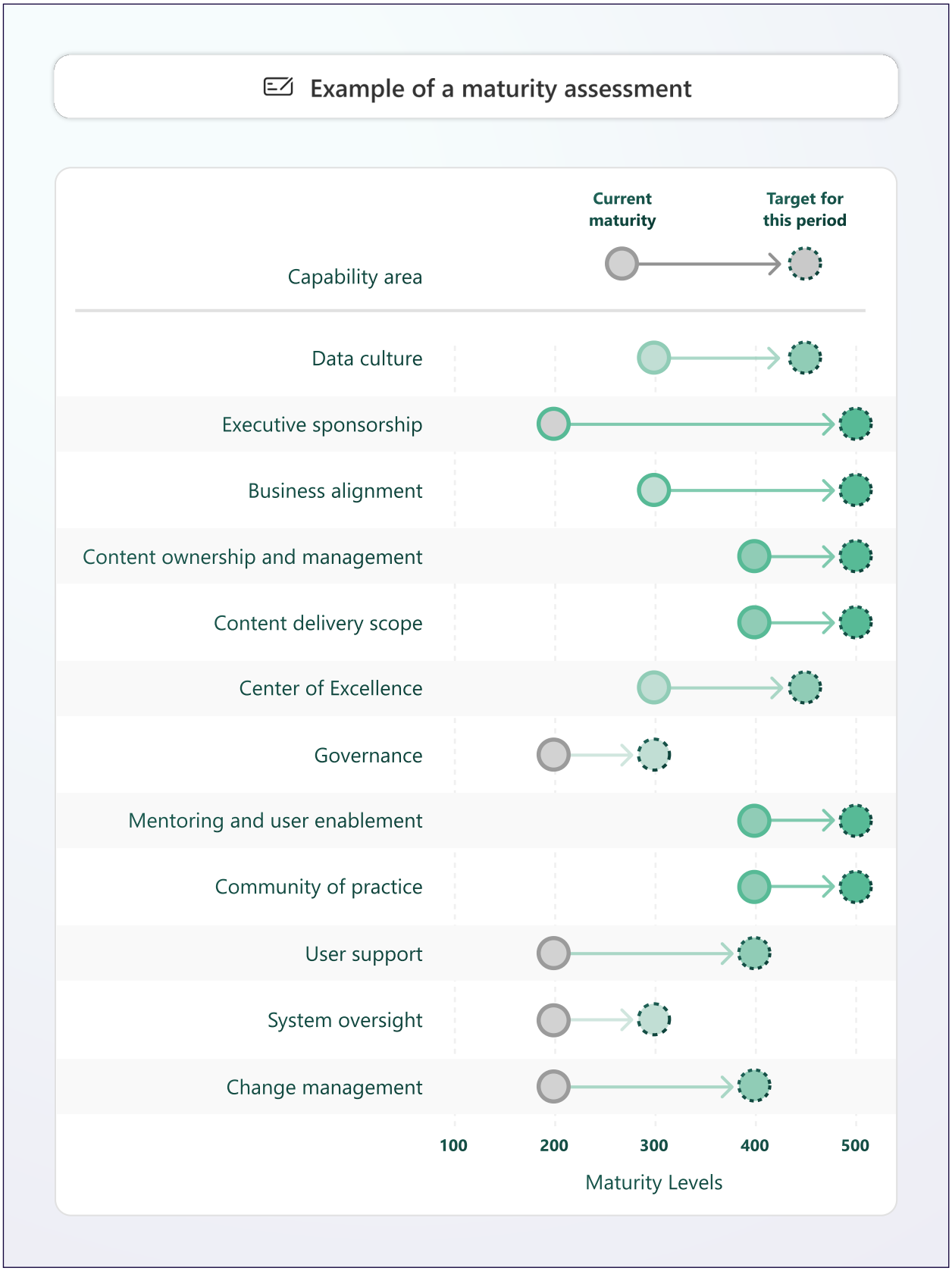
As you perform your maturity assessment, consider the following points.

- **Define what adoption means for your organization.** Consider how people in the organization should effectively use the data and tools available to them.
- **Define how you'll tie adoption to business metrics.** Consider how improving adoption will advance business strategic interests, and how improving adoption will lead to improvement in business metrics that decision makers use to evaluate progress.
- **Identify the biggest pain points and problems.** Consider which areas are most pressing and where improved adoption could have the biggest positive impact (or conversely, where poor adoption is having the biggest negative impact).

> **Tip**
>
> Before you complete your maturity assessment, you can update the maturity level descriptions so that they are directly relevant to your organization. For example, you could include other considerations that are relevant for your success, or update descriptions to use language or terminology that's specific to your organization.
>
> Also, a maturity assessment is something you can do as part of your **strategic planning**. However, you don't need to work through a full strategic planning exercise to start adoption tracking. Like adoption itself, we recommend that you focus on how to achieve sustainable, incremental progress.

Based on your current maturity levels, you can identify your strengths and set targets for each area. Targets help you to define your priorities and which objectives you want to focus on within a specific period (such as the next quarter).

The following diagram depicts an example. It shows how you can visualize your adoption maturity levels after you've conducted an initial assessment and then defined your targets. This example depicts an assessment for each of the capability areas of the Microsoft Fabric adoption roadmap.



In this example, the organization has a significant level of decentralized self-service analytics. They have identified many gaps, the largest of which is in their executive sponsorship. For this capability area, they aim for a 500 maturity level, yet they're currently achieving a 200 level. In this case, the organization might choose to prioritize executive sponsorship first.

Evaluating your maturity level and defining your adoption objectives is an iterative process. Ideally, you should repeat an evaluation each period, comparing your reassessment of the maturity levels to your targets (see Step 6).

Step 2: Identify which behaviors are important

Once you understand your adoption objectives, you next define which behaviors you want to promote or discourage. These behaviors describe the way that people create, use, or manage content. Behaviors should have a clear link to your objectives. Positive behaviors should advance progress toward your objectives, while negative behaviors slow your progress or negate it.

When you define what behaviors are important to track, consider the following points.

- Describe why desired behaviors are correct or preferred, and how more of this behavior will benefit users and the organization.
- Consider whether some behaviors are standard for the tool or specific to your organizational context. For example, reducing exports to Excel is a commonly desired behavior for analytics tools. However, other organizations might require exports for auditing purposes and might seek to specifically reduce ad hoc exports.
- Survey the organization for individuals or teams that experience significantly more (or less) success with analytics than others. When observing these teams, identify key behaviors that contribute to their successes (or struggles).

> **Note**
>
> The behaviors that you identify as important will likely be specific to your organization, its objectives, and its pattern for **content ownership and management** and **content delivery scope**. For instance, enterprise teams that distribute reports and semantic models might be more concerned with usage of these centralized items. By contrast, in a self-service environment where users make their own semantic models and reports, Fabric administrators might be more interested in identifying refresh failures and over-sharing of items or item types.

Here are some examples of positive behaviors that you might want to encourage.

- Use of strategically important reports.
- Regular reuse of shared semantic models.
- Separation of development and production workspaces (instead of publishing directly to the workspace from which content is shared with consumers).
- Sharing reports and dashboards from apps instead of workspaces or items.

- Active engagement among a community of practice.
- Use of Analyze in Excel or live connected tables in Excel.

Here are some examples of negative behaviors that you might want to discourage.

- Export to Excel from reports.
- Refresh failures of data items.
- Unowned or unused items and workspaces. An unowned item or workspace is also often referred to as *orphaned*, meaning that it's owned by someone who left the organization, such as when a workspace has no admin.
- Published redundant versions of the same item (duplication).
- Published semantic models that connect to local files.

Step 3: Define how to measure the behaviors

After you identify the behaviors that are important, you must then define how you'll measure them. This task is an essential step that leads to adoption tracking. Without measurement, you're unable to quantify and track the behaviors to understand their frequency and trend. Understanding trends is especially critical because it allows you to identify improvement over time.

> **Tip**
>
> The measures you use in adoption tracking can also be part of the key results that you use in tactical planning. For more information about how to define key results, and how you can effectively measure a target and use indicators, see **BI tactical planning**.

> **Note**
>
> Some behaviors are more difficult to monitor because they occur outside of your tenant. For instance, engagement and activity among a community of practice is important. They stimulate a healthy, self-sustaining data culture. However, gathering information to measure them might be difficult, for example, when individuals meet in person, or use private chats to communicate.
>
> Therefore, try to focus your adoption tracking on behaviors that occur within your tenant. They will be easier to measure with the **ready-made solutions** that you have available. For other behaviors and activities, consider alternative forms of measurement, like surveys.

Here are some examples of how you might measure the positive behaviors that you want to encourage.

- **Use of strategically important reports**: Use the usage metrics report to identify important reports within a workspace. Alternatively, you can also use the *ViewReport* activity from the audit log by user and report to track report usage. Observe whether the frequency of their usage is consistent or increasing over time.
- **Regular reuse of shared semantic models**: Use the *Feature usage and adoption* report to identify the ratio of semantic models to reports. Observe whether there are more reports than semantic models.
- **Separate development and production workspaces**: Use the *Feature usage and adoption* report to filter to a single report to determine whether it exists in only one workspace. Or, identify workspaces that use proper naming conventions to indicate development, test, and production stages.

Here are some examples of how you might measure the negative behaviors that you want to discourage.

- **Reduction of export to Excel**: Use the *Feature usage and adoption* report to filter to the *ExportArtifact* activity. Then determine which users are exporting to Excel the most, and whether it's rising or falling in frequency over time.
- **Oversharing or inappropriate sharing of items**: Use the *Feature usage and adoption* report to identify the most shared items, or which items external users have access to. Then evaluate whether specific items are being shared too frequently, or whether external users have access to items that they shouldn't.
- **Unowned or unused items**: Use the *Feature usage and adoption* report to identify low user activity by item. These items can become candidates for retirement and archiving.

Once you can measure the behaviors important for your adoption tracking, you should use these measurements to evaluate and understand the current situation. The outcome of this task establishes a baseline for future improvement. During the initial exploration of the data, you should also critically review each measure to ensure that it reflects the behaviors and processes that you want to improve.

Step 4: Determine whether you need other data

Depending on the measurements you'll track, you might require other data in addition to what's available from the content in the admin monitoring workspace. You can use this data to extend the semantic models in the admin monitoring workspace by creating a composite model. After you publish your composite model (to different workspace),

you can create new, custom reports that address more questions. You can also enrich the semantic model with calculations.

> **Tip**
>
> For more information about other data that you can use to track adoption, see **Custom solutions** earlier in this article.

Step 5: Assign ownership of the adoption tracking

Adoption tracking creates value only when you regularly act on the tracking you're doing. Therefore, you should assign ownership to the specific adoption tracking metric to a team or individual. This team or individual is then responsible for performing the tracking and taking (or motivating) an action based on the learnings from that information.

Consider the following points when you're ready to assign ownership.

- Clearly define the responsibilities of the ownership team or individual by describing what ownership means.
- Define possible actions that owners should take, and when they should take that action.
- Define the time period when the action should be taken (for instance, the following day, week, or month).
- Clarify the desired (or expected) outcome that the action should achieve.
- Plan recurring meetings with these teams and individuals to provide support and ensure accountability.

Step 6: Reassess and track progress over time

You should perform adoption tracking regularly to identify whether there are measurable changes in the frequency of user behaviors, and to determine whether you're making progress toward your goals. For example, adoption tracking can be a valuable addition to your iterative planning processes, such as tactical planning. This planning activity is typically where you decide on your short-term priorities and activities. During planning, you should reevaluate your maturity levels for the different capability areas and compare them to the targets that you set in Step 1.

> **Important**

> Adoption tracking isn't the result of an irregular or one-off audit of your environment, or how your organization uses Fabric. Instead, you should perform adoption tracking *consistently*. That way, you can identify improvement over time.

Here are some other reasons why you should perform adoption tracking over time.

- Measuring change over time is the best way to identify whether you're experiencing success and making progress toward your adoption objectives. For instance, you might seek to improve content ownership and management by reducing the number of unused or orphaned workspaces and items in a tenant. You'll only know whether you're successful when you continue measuring and observe this number decreasing over time.
- User behaviors will change over time as their business objectives and capabilities evolve. For instance, you might aim to improve system oversight by reducing the frequency of refresh failures of semantic models in a Fabric capacity. Over time, you might observe that more users are making Direct Lake semantic models, where refresh failures less likely to occur (since data transformation doesn't occur in the semantic model refresh). You might then decide to focus on a different metric like Capacity Unit (CU) usage from each semantic model, instead of (or in addition to) refresh failures.
- You might take different actions as you get closer to your adoption objectives, or you might need to change what you're tracking, or how you track it. For instance, you might seek to improve governance by reducing export to Excel activity. At first, you focus on reducing the total exports over time. Then, over time, you observe a decrease in the total number of exports. You then decide to shift your focus to key users who export regularly, or exports from specific data items that contain sensitive or confidential information.

Checklist - When conducting adoption tracking, key decisions and actions include:

- **Read the Microsoft Fabric adoption roadmap**: Understand what adoption is, why it's important, and the different areas that constitute effective adoption in an organization. Also, understand the difference between organizational adoption, user adoption, and solution adoption.
- **Understand what adoption means for your organization**: Consider meeting with key stakeholders in interactive meetings to hear their perspective.

- **Build a shared understanding of adoption within the organization**: Communicate what adoption means within your organization through channels like a communication hub, your centralized portal, or the community of practice.
- **Evaluate your maturity levels**: Conduct a maturity level assessment for each of the areas defined in the Microsoft Fabric adoption roadmap to understand the current state of adoption in your organization.
- **Define your objectives**: Identify the maturity levels that you want to attain. Ensure that you set realistic objectives. Then prioritize key areas that will help you achieve progress.
- **Identify behaviors that you want to discourage or replicate**: Survey the organization and identify any behaviors related to effectively working with data or performing analytics. Identified behaviors could lead to success or challenges.
- **Define how to measure these behaviors**: Determine how you'll measure these behaviors to identify improvement over time, which should lead to more positive behaviors and less negative behaviors.
- **Decide whether you'll use custom or ready-made solutions**: Evaluate what adoption tracking solution you'll use to measure behaviors. Explore the ready-made solutions available in your tenant before you commit to a custom solution. Remember that you can enrich the admin monitoring workspace models with other data and calculations.
- **Assign ownership and define actions**: Clarify who has ultimate responsibility for certain indicators, and what actions these indicators will motivate (and who's responsible for taking these actions).
- **Plan regular meetings to assess progress and ensure accountability**: Make sure that these meetings aren't about enforcing ownership. Instead, focus on the adoption tracking itself, and whether it's happening consistently and effectively, or not.

Related content

For more considerations, actions, decision-making criteria, and recommendations to help you with Power BI implementation decisions, see Power BI implementation planning.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

The Power BI ecosystem is diverse and can be implemented in different ways. In this series of articles, common usage scenarios are provided to illustrate different ways that Power BI can be deployed and utilized by creators and consumers. Understanding how these usage scenarios are used in your organization, and by whom, can influence the implementation strategies you decide to take.

> **Note**
>
> The most prevalent components of Power BI are identified in each scenario based on how Power BI is intended to be used for that scenario. The objective is *not* to call out every possible option for each usage scenario. Rather, each scenario diagram depicts the primary features that are most relevant for that scenario.

How to use the scenarios

Use the scenarios to help you with Power BI architecture planning and implementation decisions. Here are some suggestions:

- Initially read the scenarios in the order they're documented. Become familiar with the concepts and how the scenarios build upon each other.
- Focus on the scenarios that align well with your data culture. Also consider how content ownership and management is handled, as well as content delivery scope when determining which usage scenarios are a good fit.
- Consider which areas of your BI operations could be strengthened in your organization. For example, if your goal is to reduce the level of data duplication, focus on the managed self-service BI scenario. If your goal is to improve efficiency of data preparation efforts, focus on the self-service data preparation scenario.
- Determine if there are ways to use Power BI that will bring additional value or reduce risk for your organization. For example, if your goal is to achieve a balance

between centralization and decentralization (described further in the content ownership and management articles), consider the customizable managed self-service BI scenario.
- After understanding the areas of your BI operations that you want to implement or strengthen, create a project plan that defines tactical steps to arrive at your desired future state.

> **Tip**
>
> You might need to mix and match the ideas described in the usage scenarios to create a Power BI implementation strategy that fits your circumstances. To support the needs of users from different departments and business units, expect to draw from multiple Power BI implementation methods simultaneously. That way, you'll be able to support diverse content creators and various solutions.

Content collaboration and delivery scenarios

The following usage scenarios are about *content collaboration and delivery*. These initial four scenarios focus primarily on content ownership and management, and content delivery scope. They are inter-related, building upon each other in a way that aligns with how business intelligence teams evolve and grow over time. They can be thought of as the building blocks that other scenarios build upon—particularly the self-service BI scenarios that are described in the next section. Therefore, it's a good idea to review those scenarios first.

- **Personal BI**: The content creator has a lot of freedom and flexibility to create content for individual usage. This scenario describes using a personal workspace for private usage.
- **Team BI**: The primary focus is on informal collaboration among team members who work closely together on a team. This scenario describes using a workspace for both collaboration and distribution. It also showcases the value of using Microsoft Teams for collaboration between Power BI creators and consumers.
- **Departmental BI**: There's a focus on distributing content to a larger number of users within a department or business unit. This scenario describes using a Power BI app for distributing content.
- **Enterprise BI**: The primary focus is on content distribution at scale. This scenario describes using Premium capacity to distribute content to a larger number of read-only consumers who have a Fabric free license.

> **Note**
>
> Additional information about content ownership and management and content delivery scope are described in the Fabric adoption roadmap.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Self-service BI scenarios

Four usage scenarios focus on supporting *self-service BI* activities, in which analytical responsibilities are handled by people throughout many areas of the organization. The content collaboration and delivery scenarios (described in the previous group of scenarios) also include aspects of self-service BI but from a slightly different viewpoint. The intention of this set of scenarios is to focus on several important aspects to plan for in a Power BI implementation.

The self-service BI scenarios presented here primarily emphasize the use of *managed self-service BI* in which data management is centralized. Reusability of this centralized data is one of the primary goals. Business users take responsibility for creation of reports and dashboards.

- **Managed self-service BI**: The goal is for many report creators to reuse shared semantic models. This scenario describes decoupling the report creation process from the semantic model creation process. To encourage report authors to find and reuse an existing shared semantic model, it should be endorsed and made discoverable in the OneLake catalog in the Fabric portal.
- **Customizable managed self-service BI**: The focus is on the semantic model creator customizing or extending an existing semantic model to satisfy new requirements. This scenario describes publishing a customized data model where some tables are new while others are dependent on the existing shared semantic model.

- **Self-service data preparation**: The focus is on centralizing data preparation activities to improve consistency and reduce effort. This scenario describes creating Power BI dataflows to avoid repeating data preparation Power Query logic in many different Power BI Desktop files. A dataflow can be consumed as a data source by numerous semantic models.

- **Advanced data preparation**: The focus is on improving the reach and reusability of dataflows for multiple users, teams, and use cases. This scenario describes use of multiple workspaces based on purpose: staging, cleansed, and final.

- **Self-service real-time analytics**: The focus is on how a business analyst can produce real-time Power BI reports.

- **Prototyping and sharing**: Prototyping techniques are very useful for validating requirements for visuals and calculations by subject matter experts. Prototyping solutions might be temporary, short-lived solutions, or they might ultimately evolve into a solution that's fully validated and released. This scenario describes using Power BI Desktop during an interactive prototyping session. It's followed by sharing in the Power BI service when additional feedback is needed from a subject matter expert.

  > **Note**
  >
  > Additional information about content ownership and management, and content delivery scope, which affect self-service BI activities and decisions, are described in the Fabric adoption roadmap.

Content management and deployment scenarios

The following *content management and deployment* scenarios describe approaches for how content creators and owners use methodical and disciplined lifecycle management processes to reduce errors, minimize inconsistencies, and improve the user experience for consumers.

- **Self-service content publishing**: The focus is on ensuring that content is stable for consumers. This scenario describes using a Power BI deployment pipeline to publish content through development, test, and production workspaces. It also describes how (optionally) Premium per user license mode can be used for development and test workspaces, and Premium per capacity license mode for the production workspace.

- **Enterprise content publishing**: The focus is on using more sophisticated and programmatic techniques to publish content through development, test, and production workspaces. In this scenario, it describes how you can use Azure DevOps to orchestrate collaboration and content publication.
- **Advanced data model management**: The focus is on empowering creators with advanced data modeling and publishing capabilities. This scenario describes managing a data model by using Tabular Editor, which is a third-party tool. Data modelers publish their models to the Power BI service by using the XMLA endpoint, which is available with Power BI Premium.

Embedding and hybrid scenarios

There are two *embedding and hybrid* scenarios: content embedding and on-premises reporting. They describe ways to deploy and distribute content that can be used in addition to, or instead of, the Power BI service.

- **Embed for your organization**: The focus is on making analytical data easier for business users to access by integrating visuals within the tools and applications they use every day. This scenario describes using the Power BI REST APIs to embed content in a custom application for users who have permission and appropriate licenses to access Power BI content in your organization.
- **Embed for your customers**: This scenario describes using the Power BI REST APIs to embed content in a custom application for users who don't have permission or appropriate licenses to access Power BI content in your organization. The custom application requires an embedding identity that has permission and an appropriate license to access Power BI content. The custom application could be a multitenancy application.
- **On-premises reporting**: The focus is on using a basic portal for publishing, sharing, and consuming business intelligence content within your organizational network. This scenario describes using Power BI Report Server for this purpose.

Download the scenario diagrams

Each of the usage scenario articles contains a scenario diagram. We encourage you to download the scenario diagrams if you'd like to embed them in your presentations, documentation, or blog posts—or print them out as wall posters. Because they're Scalable Vector Graphics (SVG) images, you can scale them up or down without any loss of quality.

Related content

In the next article in this series, learn about enabling private analytics for an individual with the personal BI usage scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Personal BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

As described in the Fabric adoption roadmap, *personal BI* is about enabling an individual to gain analytical value. It's also about allowing them to perform business tasks more efficiently with the use of data, information, and analytics. Personal BI is sometimes thought of as the entry point for *self-service BI*.

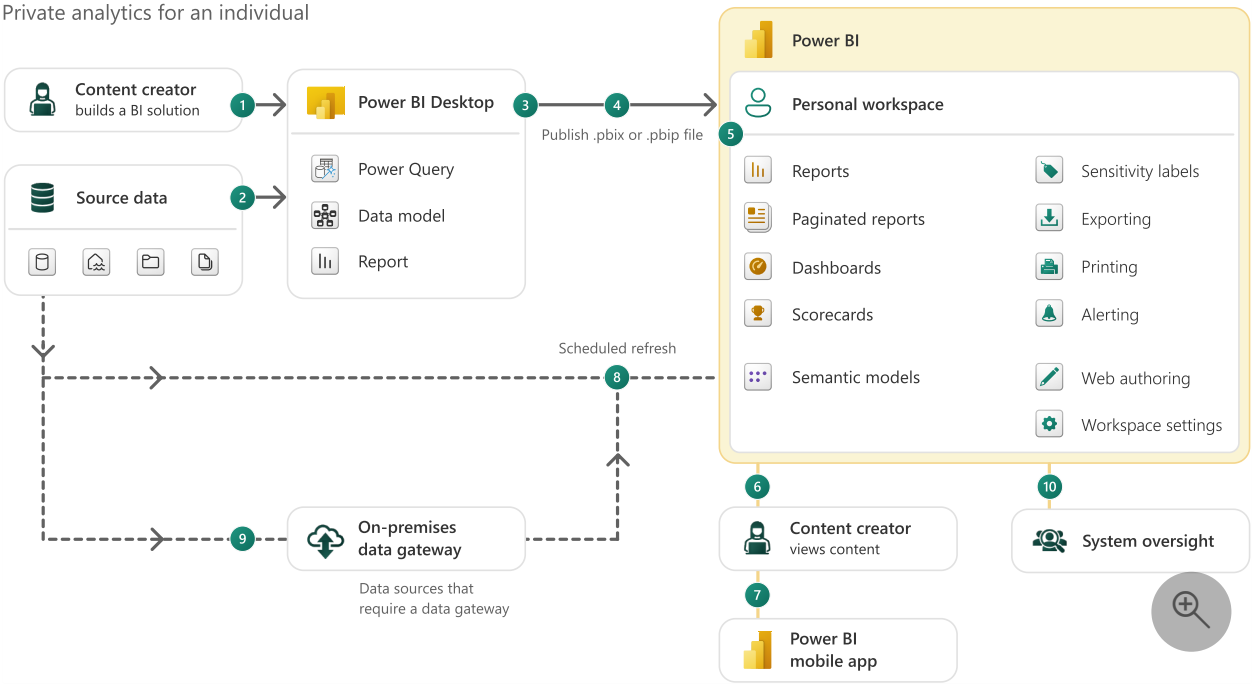
In personal BI scenarios, the content creator has a lot of freedom and flexibility to create content for individual usage. Simplicity and speed are usually high priorities. There's no sharing or collaboration in this usage scenario—those topics are covered in the team BI, departmental BI, and enterprise BI scenario articles.

> **Note**
>
> There are four *content collaboration and delivery* scenarios that build upon each other. The personal BI scenario is the first of the four scenarios. A list of all scenarios can be found in the Power BI usage scenarios overview article.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support personal BI. The focus is on private analytics for an individual.

Personal BI
Private analytics for an individual



> **Tip**
>
> We encourage you to download the scenario diagram if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	The Power BI content creator develops a BI solution using Power BI Desktop.
2	Power BI Desktop connects to data from one or more data sources. Queries and data mashups, which combine multiple sources, are developed in the Power Query Editor.
3	Data model development and report creation are done in Power BI Desktop. In a personal BI solution, the primary intention is typically data exploration and analysis.
4	When ready, the content creator publishes the Power BI Desktop file (.pbix) of Power BI project file (.pbip) to the Power BI service.
5	Since the primary intention is personal usage, the content is published to the content creator's personal workspace. Some advantages of using the Power BI service (instead of remaining solely in Power BI Desktop) include scheduled data refresh, dashboard alerts, and the ability to consume content using a mobile app. The content creator can also edit

Item	Description
	reports and models in their personal workspace with web authoring (if they enable the workspace setting).
6	The content creator views and interacts with the content published. One option is to sign in to the Power BI service using a web browser.
7	The content creator can also use a Power BI mobile app to view published content.
8	Scheduled data refresh can be set up in the Power BI service to keep imported data up to date.
9	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
10	Power BI administrators oversee and monitor activity in the Power BI service. Personal workspaces are usually governed to a much lesser extent than workspaces that are intended for collaboration and distribution.

Key points

The following are some key points to emphasize about the personal BI scenario.

Choice of authoring tools

Power BI Desktop is the authoring tool to develop queries, models, and Power BI reports. It's possible to use different tools to create Excel reports and Power BI paginated reports (not depicted in the scenario diagram).

Reliance on personal workspace

Use of the personal workspace can be thought of like an *analytical sandbox*. For many organizations, personal content is subject to little governance or formal oversight. However, it's still wise to educate content creators on guidelines to be successful with personal BI. Use of the sharing feature available within a personal workspace isn't depicted in this usage scenario since the focus is individual analytics.

> **Important**
>
> Limit the use of personal workspaces and ensure no mission-critical content is stored in them. Although a Power BI administrator can access and govern a user's

personal workspace, storing critical content in personal workspaces does represent risk to the organization.

Use of Fabric free license

For personal use, which by definition means there's no sharing or collaboration with others, only certain capabilities of the Power BI service are available to a user with a Fabric free license. When using a free license, most activities to create and publish content to the Power BI service are limited to their personal workspace.

> **Tip**
>
> The **enterprise BI** scenario describes how users with a Fabric free license can view content when it's hosted in a Premium capacity.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

> **Note**
>
> A data gateway in *personal mode* is most frequently installed on the machine of an individual user. Therefore, a data gateway in personal mode is best-suited to personal BI usage scenarios. Your organization can restrict individuals from

installing data gateways, in which case the content creator can use a data gateway in standard mode (typically set up and managed by IT).

Information protection

Information protection policies can be applied to content in the Power BI service. Some organizations have a mandatory label policy that requires a sensitivity label be assigned, even within a personal workspace.

System oversight

The activity log records user activities that occur in the Power BI service, and it extends to personal workspaces. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and detect risky activities. Auditing and governance requirements are typically less stringent for personal BI scenarios.

Related content

In the next article in this series, learn about small team collaboration with the team BI usage scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Team BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

Once a valuable BI solution is created, it's time to collaborate with colleagues. The goal is to deliver additional value beyond what can be achieved with the personal BI scenario.

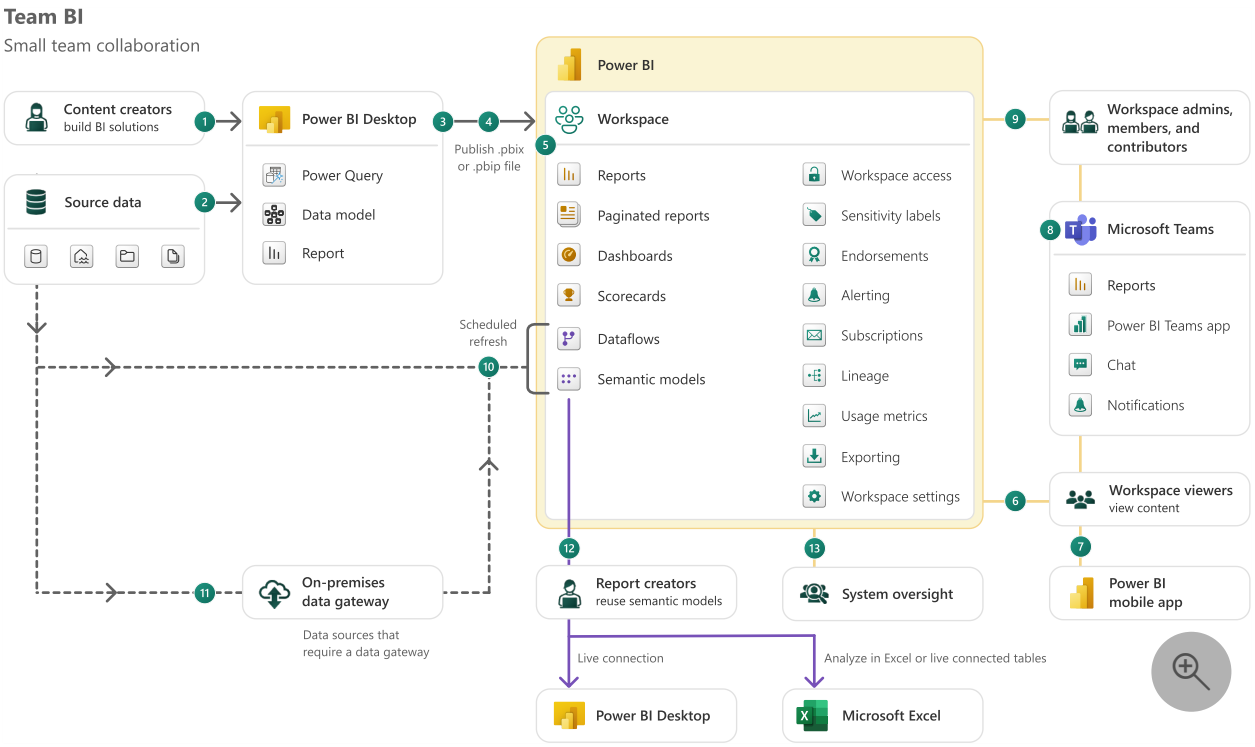
As described in the Fabric adoption roadmap, *team BI* focuses on a small team of people who work closely together. Collaborating and sharing content with each other in an informal way is usually a key objective of team BI (more formal delivery of content is covered in the departmental BI and enterprise BI scenarios).

Sometimes when working with close colleagues, collaboration for small teams can be done simply within a workspace. A workspace can be thought of as a way to informally view content (without the formality of publishing a Power BI app, which is covered in the departmental BI scenario) by members of a small team.

> **Note**
>
> There are four *content collaboration and delivery* usage scenarios that build upon each other. The team BI scenario is the second of the four scenarios. A list of all scenarios can be found in the Power BI usage scenarios article.
>
> The managed self-service BI scenario introduces an important concept about decoupling semantic model and report development. For simplicity, this concept isn't explicitly discussed in this article. You're encouraged to apply the concepts discussed in the managed self-service BI scenario whenever possible.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support team BI. The primary focus is small team collaboration.



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Power BI content creators develop BI solutions using Power BI Desktop. In a team BI scenario, it's common for creators to work within a decentralized team, department, or business unit.
2	Power BI Desktop connects to data from one or more data sources. Queries and data mashups, which combine multiple sources, are developed in the Power Query Editor.
3	Data model development and report creation are done in Power BI Desktop. In a team BI solution, the purpose is to help team members understand the meaning and significance of data by placing it in a visual context.
4	When ready, content creators publish their Power BI Desktop file (.pbix) or Power BI project file (.pbip) to the Power BI service.
5	The content is published to a workspace. Its primary purpose is to provide information and enable collaboration for a small team. Content creators can also create or edit some

Item	Description
	content in a workspace.
6	All users assigned to a workspace role (viewer or higher) view and interact with content in the workspace. One option is to sign in to the Power BI service using a web browser.
7	The Power BI mobile apps are also available for viewing published content.
8	Users who frequently work in Microsoft Teams might find it convenient to manage or view Power BI content directly in Teams. They can use the Power BI app for Microsoft Teams or view reports that are embedded within a team channel. Users can also have private chats with each other and receive notifications directly in Teams.
9	Users assigned to the administrator, member, or contributor workspace role can publish and manage workspace content.
10	Scheduled data refresh can be set up in the Power BI service to keep imported data—in semantic models or dataflows—up to date.
11	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
12	Other self-service content creators can author new reports using an existing semantic model. They can choose to use Power BI Desktop, Excel, or Power BI Report Builder (not depicted in the scenario diagram). The reuse of existing semantic models in this manner is highly encouraged.
13	Power BI administrators oversee and monitor activity in the Power BI service. Team BI solutions could be subject to more governance requirements than personal BI, but fewer than departmental BI and enterprise BI solutions.

Key points

The following are some key points to emphasize about the team BI scenario.

Source file storage

Power BI Desktop is the authoring tool to develop queries, models, and interactive reports. Because collaboration is a high priority for team BI, it's important to store the source Power BI Desktop file in a secure, shared location. Locations such as OneDrive for work or school or SharePoint (not depicted in the scenario diagram) are useful due to built-in version history and automatic file synchronization. A shared library is securable, easily accessible by colleagues, and has built-in versioning capabilities.

When the co-management of a BI solution involves multiple people with different skillsets, consider decoupling the model and reports into separate Power BI Desktop

files (described in the managed self-service BI scenario). This approach encourages reuse of the semantic model and is more efficient than continually alternating between the people who are editing the Power BI Desktop file. That's particularly helpful when, for instance, one person works on the semantic model while another person works on the reports.

Workspaces

A Power BI workspace serves as a logical container in the Power BI service for storing related Power BI items, such as semantic models and reports. In a team BI scenario, it's practical and simple to use the workspace for collaboration as well as for the viewing of reports by a small number of users. The distribution of content as a Power BI app is described in the departmental BI scenario.

Workspace access and sharing

In addition to organizing content, a workspace forms a security boundary. Assign users to workspace roles when a team member needs to edit or view all items published to a workspace. The four workspace roles (administrator, member, contributor, and viewer) support productivity for self-service content creators and consumers, without over-provisioning permissions.

> **Note**
>
> Alternatively, workspace users can share individual reports and dashboards (not depicted in the scenario diagram). Sharing can grant read-only access to someone who isn't assigned to a workspace role. However, try to limit sharing because it can be tedious to setup for many items or many users.

Power BI user licenses

When collaborating in a workspace, all users must have a Power BI Pro or Power BI Premium Per User (PPU) license.

> **Note**
>
> There's one exception to the requirement of a Power BI Pro or PPU license: When the workspace is assigned to Premium capacity or Fabric F64 or greater capacity,

Fabric free license users (with proper permissions) can view the workspace (and/or Power BI app) content. This approach is described in the **enterprise BI** scenario.

Reuse existing semantic models

The reuse of existing semantic models is important for team collaboration. It helps to promote a *single version of the truth*. It's particularly important when a small number of semantic model creators support many report creators. A Power BI Desktop live connection can connect a report to an existing semantic model, avoiding the need to create another semantic model. Alternatively, when users prefer to create an Excel report, they can use the Analyze in Excel feature. This type of connectivity is preferred to exporting data to Excel because it:

- Avoids creating duplicate semantic models.
- Reduces the risk of inconsistent data and calculations.
- Supports all slicing, dicing, and pivoting capabilities within the visuals while remaining connected to the semantic model that's stored in the Power BI service.

To access an existing semantic model, the content creator must have Build permission for the semantic model. It can be granted directly or indirectly when the user is assigned to a workspace role (contributor or higher) or granted when publishing a Power BI app or sharing a Power BI item. The managed self-service BI scenario explores the reuse of shared semantic models further.

Power BI integration with Microsoft Teams

Using a modern collaboration tool like Microsoft Teams engages users to make data-driven decisions. Microsoft Teams supports collaborative discussions about data while viewing Power BI content within a natural workflow. To learn about more collaboration options, see Collaborate in Microsoft Teams with Power BI.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

Note

For team, departmental, and enterprise BI scenarios, a centralized data gateway in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements.

Related content

In the next article in this series, learn about distributing content to a larger number of viewers in the departmental BI usage scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Departmental BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

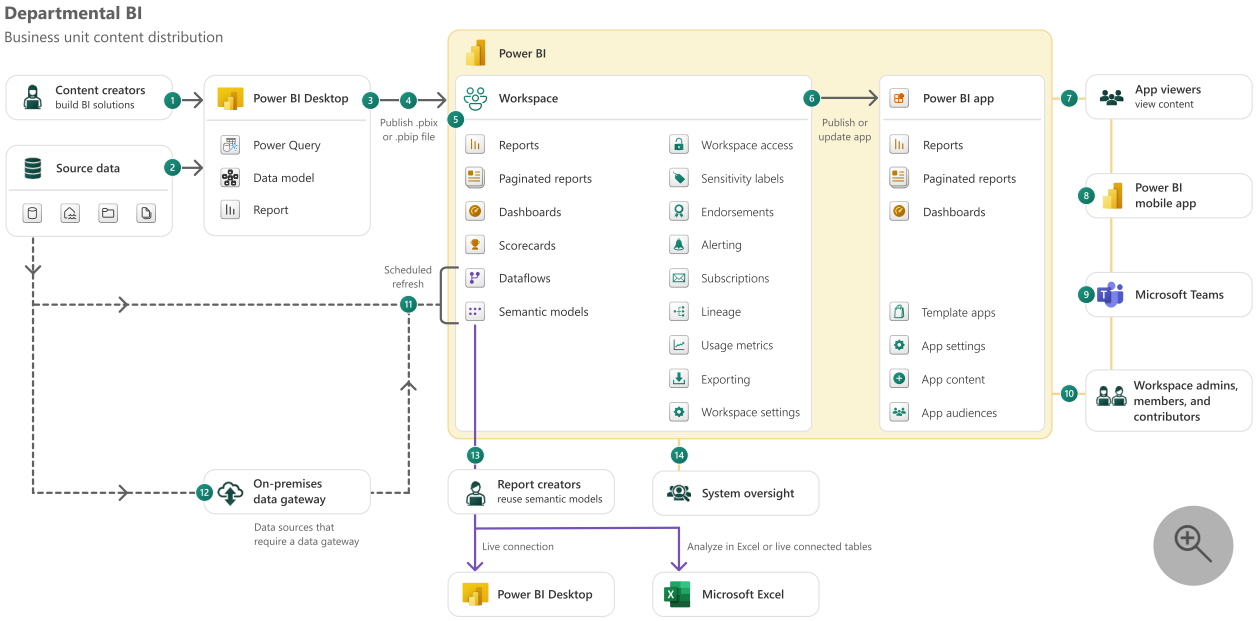
As described in the Fabric adoption roadmap, *departmental BI* focuses on distributing content to a larger number of users. These users are typically members of a department or business unit.

When teams grow larger, it becomes impractical to use a workspace effectively for the distribution of all reports (as described in the team BI scenario). A more effective way to handle larger departmental BI scenarios is to use the workspace for collaboration and distribute workspace content as an app to consumers.

> **Note**
>
> There are four *content collaboration and delivery* usage scenarios that build upon each other. The departmental BI scenario is the third of the four scenarios. A list of all scenarios can be found in the Power BI usage scenarios article.
>
> The managed self-service BI scenario introduces an important concept about decoupling semantic model and report development. For simplicity, this concept isn't explicitly discussed in this article. You're encouraged to apply the concepts discussed in the managed self-service BI scenario whenever possible.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support departmental BI. The primary focus is on using a Power BI app for content distribution to a large consumer audience.



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Power BI content creators develop BI solutions using Power BI Desktop. In a departmental BI scenario, it's common for creators to work within a decentralized team, department, or business unit.
2	Power BI Desktop connects to data from one or more data sources. Queries and data mashups, which combine multiple sources, are developed in the Power Query Editor.
3	Data model development and report creation are done in Power BI Desktop. In a departmental BI solution, the purpose is to help colleagues understand the meaning and significance of data by placing it in a visual context.
4	When ready, content creators publish their Power BI Desktop file (.pbix) or Power BI project file (.pbip) to the Power BI service.
5	The content is published to a workspace. Its primary purpose is to provide a collaboration area for people who are responsible for creating, managing, and validating content. Content creators can also create or edit some content in the workspace.

Item	Description
6	Some, or all, reports and dashboards are published as a Power BI app. The purpose of the app is to provide a set of related content for consumers to view in a user-friendly way.
7	Power BI app users are added to app audiences, which are assigned read-only permissions. App settings, app content, and app audiences are managed separately from the workspace. Some teams may install and use template apps.
8	The Power BI mobile apps are also available for viewing apps and workspace content.
9	Users who frequently work in Microsoft Teams might find it convenient to manage or view Power BI content directly in Teams.
10	Users assigned to the administrator, member, or contributor workspace roles can publish and manage workspace content.
11	Scheduled data refresh is set up in the Power BI service to keep imported data—in semantic models or dataflows—up to date.
12	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
13	Other self-service content creators can author new reports using an existing semantic model. They can choose to use Power BI Desktop, Excel, or Power BI Report Builder (not depicted in the scenario diagram). The reuse of existing semantic models in this manner is highly encouraged.
14	Power BI administrators oversee and monitor activity in the Power BI service. Departmental BI solutions could be subject to more governance requirements than team BI solutions, but fewer requirements than enterprise BI solutions.

Key points

The following are some key points to emphasize about the departmental BI scenario.

Source file storage

Power BI Desktop is the authoring tool to develop queries, models, and interactive reports. For departmental BI, it's important to store the source Power BI Desktop file in a secure, shared location. Locations such as OneDrive for work or school or SharePoint (not depicted in the scenario diagram) are useful. A shared library is securable, easily accessible by colleagues, and has built-in versioning capabilities.

When the co-management of a BI solution involves multiple people with different skillsets, consider decoupling the model and reports into separate Power BI Desktop

files (described in the managed self-service BI scenario). This approach encourages reuse of the semantic model and is more efficient than continually alternating between the people who are editing the Power BI Desktop file. That's particularly helpful when, for instance, one person works on the semantic model while another person works on the reports.

Workspaces

A Power BI workspace serves as a logical container in the Power BI service for storing related Power BI items, such as semantic models and reports. Although this scenario depicts one workspace, multiple workspaces are commonly required to satisfy all workspace planning requirements.

The managed self-service BI scenario describes the use of separate workspaces.

Power BI app publication

For departmental BI, a Power BI app works well for content distribution to consumers (rather than direct workspace access, which is described in the team BI scenario). A Power BI app provides the best experience for consumers because it presents a set of related content with a user-friendly navigation experience. A Power BI app is particularly useful in situations where there's a larger and more diverse number of consumers, or when the content developer doesn't work closely with the app consumers.

Power BI app permissions

Power BI app users are granted read-only permission to the app, and these permissions are managed separately from the workspace. This additional level of flexibility is useful for managing who can view the content.

For departmental BI, it's a best practice to limit workspace access to those who are responsible for content authoring, development, and quality assurance activities. Typically, only a small number of people genuinely require workspace access. Consumers can access the content by opening the Power BI app, rather than opening the workspace.

Power BI user licenses

All content creators and consumers of the workspace or the Power BI app must have a Power BI Pro or Power BI Premium Per User (PPU) license.

> **Note**
>
> There's one exception to the requirement of a Power BI Pro or PPU license: When the workspace is assigned to Premium capacity or Fabric F64 or greater capacity, Fabric free license users (with proper permissions) can view the workspace (and/or app) content. This approach is described in the **enterprise BI** scenario.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Reuse existing semantic models

The reuse of existing semantic models is important for team collaboration. It helps to promote a *single version of the truth*. It's particularly important when a small number of semantic model creators support many report creators. A Power BI Desktop live connection can connect a report to an existing semantic model, avoiding the need to create another semantic model. Alternatively, when users prefer to create an Excel report, they can use the Analyze in Excel feature. Retaining connectivity to the semantic model is preferred to exporting data to Excel because it:

- Avoids creating duplicate semantic models.
- Reduces the risk of inconsistent data and calculations.
- Supports all slicing, dicing, and pivoting capabilities within the visuals while remaining connected to the semantic model that's stored in the Power BI service.

To access an existing semantic model, the content creator must have Build permission for the semantic model. It can be granted directly or indirectly when the user is assigned to a workspace role (contributor or higher) or granted when publishing a Power BI app or sharing a Power BI item. The managed self-service BI scenario explores the reuse of shared semantic models further.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

> **Note**
>
> For team, departmental, and enterprise BI scenarios, a centralized **data gateway** in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements.

Related content

In the next article in this series, learn about organization-wide content distribution at scale in the enterprise BI scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Enterprise BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

As described in the Fabric adoption roadmap, *enterprise BI* is characterized by having a significantly larger number of content consumers, compared to a much smaller number of authors who create and publish content.

The distinction between the enterprise BI and the departmental BI scenarios is the use of Microsoft Fabric capacity or Power BI Premium capacity, which allow content to be widely distributed to consumers who have a Fabric free license. Consumers can include users within the organization, as well as guest users who are external to the organization.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

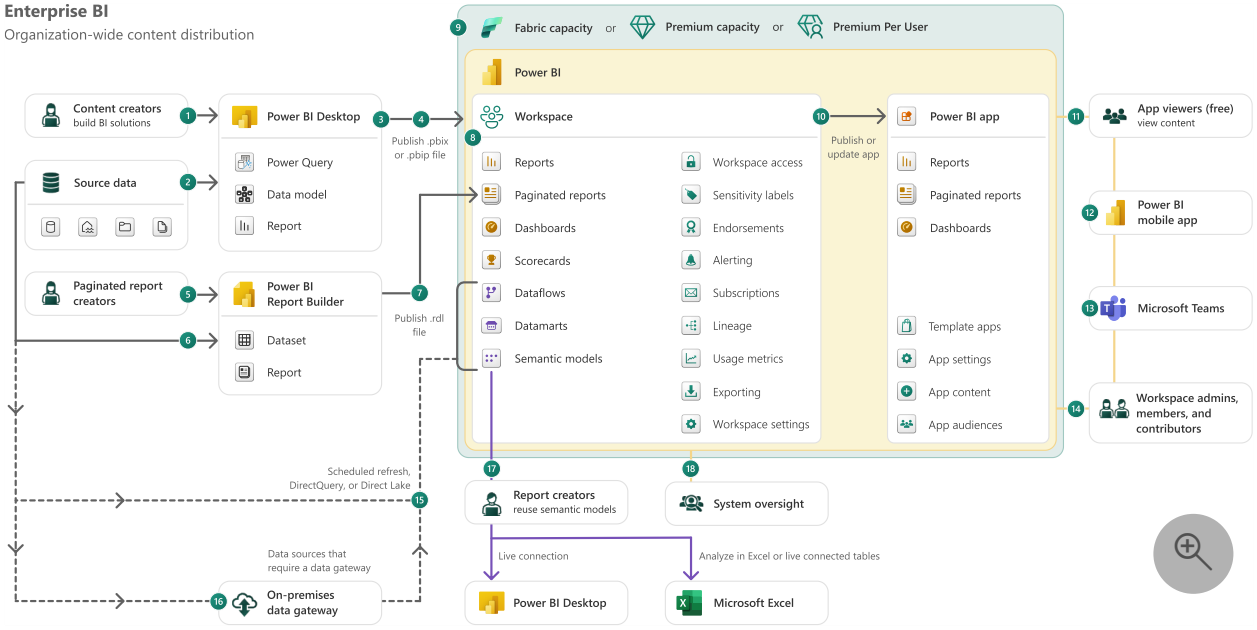
Large enterprise BI implementations often employ a centralized approach. Enterprise Power BI content is commonly maintained by a centralized team, for use broadly throughout the organization. The centralized team responsible for content management is usually IT, BI, or the Center of Excellence (COE).

> **Note**
>
> There are four *content collaboration and delivery* usage scenarios that build upon each other. The enterprise BI scenario is the fourth scenario. A list of all scenarios can be found in the Power BI usage scenarios article.

The managed self-service BI scenario introduces an important concept about decoupling semantic model and report development. For simplicity, this concept isn't explicitly discussed in this article. You're encouraged to apply the concepts discussed in the managed self-service BI scenario whenever possible.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support enterprise BI. The primary focus is on organization-wide content distribution at scale including the use of Power BI Premium capacity. This scenario also depicts developing Power BI paginated reports.



Tip

We encourage you to download the scenario diagram if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Power BI content creators develop BI solutions using Power BI Desktop. In an enterprise BI scenario, it's common that creators are members of a centralized team (such as IT, BI, or

Item	Description
	the COE) that supports users across organizational boundaries.
2	Power BI Desktop connects to data from one or more data sources. Queries and data mashups, which combine multiple sources, are developed in the Power Query Editor.
3	Data model development and report creation are done in Power BI Desktop. The purpose is to help colleagues understand the meaning and significance of data by placing it in a visual context.
4	When ready, content creators publish their Power BI Desktop file (.pbix) or Power BI project file (.pbip) to the Power BI service.
5	Report creators develop paginated reports by using Power BI Report Builder.
6	Power BI Report Builder queries data from one or more data source types.
7	When ready, report creators publish their Power BI Report Builder file (.rdl) to the Power BI service.
8	Multiple Power BI item types can be published to a workspace that has Fabric capacity, Premium capacity, or Premium Per User license mode.
9	In the enterprise BI scenario, use of Fabric capacity or Premium capacity (rather than Premium Per User) is depicted. This choice is made to support content delivery to many content viewers who have a free Fabric license.
10	Some, or all, reports and dashboards are published as a Power BI app. The purpose of the app is to provide a set of related content for consumers to view in a user-friendly way.
11	Power BI app users are added to app audiences, which are assigned read-only permission. App settings, app content, and app audiences are managed separately from the workspace. In an enterprise BI scenario, users with any type of Power BI license (Fabric (free), Power BI Pro, or Premium Per User) can be assigned as a viewer of the app. This feature applies only when the workspace is assigned a license mode of **Fabric capacity** or **Premium capacity** (free users cannot access workspace content when it's assigned a license mode of **Premium per user** or **Embedded**).
12	The Power BI mobile apps are also available for viewing app and workspace content.
13	Users who frequently work in Microsoft Teams might find it convenient to manage or view Power BI content directly in Teams.
14	Users assigned to the administrator, member, or contributor workspace roles can publish and manage workspace content.
15	Scheduled data refresh is set up in the Power BI service to keep imported data—in semantic models or dataflows—up to date.

Item	Description
16	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
17	Other self-service content creators can author new reports using an existing semantic model. They can choose to use Power BI Desktop, Excel, or Power BI Report Builder. The reuse of existing semantic models in this manner is highly encouraged. The managed self-service BI scenario explores semantic model reuse further.
18	Fabric administrators oversee and monitor activity in the Fabric portal. Enterprise BI solutions are often subject to stricter governance requirements than team BI or departmental BI solutions.

Key points

The following are some key points to emphasize about the enterprise BI scenario.

Choice of report authoring tools

Power BI Desktop is a tool to develop highly interactive reports, whereas Power BI Report Builder is a tool to develop paginated reports. For more information about when to use paginated reports, see When to use paginated reports in Power BI.

Excel reports can also be published to the Power BI service (not depicted in the scenario diagram) when a PivotTable or PivotChart better meets reporting requirements.

Source file storage

For enterprise BI, it's important to store the source Power BI Desktop files and Power BI Report Builder files in a secure, shared location. Locations such as OneDrive for work or school or SharePoint (not depicted in the scenario diagram) are useful. A shared library is securable, easily accessible by colleagues, and has built-in versioning capabilities.

When the co-management of a BI solution involves multiple people with different skillsets, consider decoupling the model and reports into separate Power BI Desktop files (described in the managed self-service BI scenario). This approach encourages reuse of the semantic model, and is more efficient than continually alternating between the people who are editing the Power BI Desktop file. That's particularly helpful when, for instance, one person works on the semantic model while another person works on the reports.

Workspaces

A Power BI workspace serves as a logical container in the Power BI service for storing related Power BI items, such as semantic models and reports. Although this scenario depicts one workspace, multiple workspaces are commonly required to satisfy all workspace planning requirements.

The managed self-service BI scenario describes the use of separate workspaces.

Workspace license mode

A workspace license mode can be assigned to **Pro**, **Premium per user** (PPU), **Premium per capacity**, or **Embedded**. This choice impacts feature availability, as well as which users can access the content in the workspace and the associated Power BI app. An enterprise BI scenario often involves many consumers of the content. So, it can be cost effective to use the **Premium per capacity** license mode to distribute content to users with a free license.

Power BI app publication

For enterprise BI, a Power BI app works well for content distribution to consumers (rather than direct workspace access, which is described in the team BI scenario). A Power BI app provides the best experience for consumers because it presents a set of related content with a user-friendly navigation experience. A Power BI app is particularly useful in situations where there's a larger and more diverse number of consumers, or when the content developer doesn't work closely with the app consumers.

Power BI app permissions

Power BI app users are granted read-only permission to the app, and these permissions are managed separately from the workspace. This additional level of flexibility is useful for managing who can view the content.

For enterprise BI, it's a best practice to limit workspace access to those who are responsible for content authoring, development, and quality assurance activities. Typically, only a small number of people genuinely require workspace access. Consumers can access the content by opening the Power BI app, rather than opening the workspace.

Distribute content to Fabric free license users

Users with a Fabric free license (or Power BI Pro or PPU license) can view content when granted app access or are added to a workspace role—provided the workspace is assigned to Premium capacity. This ability to distribute content to users with a free license is not available for any of the other workspace license modes, including **Pro**, **Premium per user**, or **Embedded**.

Power BI Premium capacity license

Use of a P SKU (such as P1, P2, P3, P4, or P5) is described in this scenario. A P SKU is required for typical production scenarios and is appropriate for the enterprise BI scenario described in this article.

Manage lifecycle of content

Generally, enterprise BI solutions require stability for production content. One aspect is controlling when and how content is deployed to production. Use of deployment pipelines is described in the self-service content publishing scenario.

Reuse existing semantic models

The reuse of existing semantic models is important for team collaboration. It helps to promote a *single version of the truth*. It's particularly important when a small number of semantic model creators support many report creators. A Power BI Desktop live connection can connect a report to an existing semantic model, avoiding the need to create another semantic model. Alternatively, when users prefer to create an Excel report, they can use the Analyze in Excel feature. Retaining connectivity to the semantic model is preferred to exporting data to Excel because it:

- Avoids creating duplicate semantic models.
- Reduces the risk of inconsistent data and calculations.
- Supports all slicing, dicing, and pivoting capabilities within the visuals while remaining connected to the semantic model that's stored in the Power BI service.

To access an existing semantic model, the content creator must have Build permission for the semantic model. It can be granted directly or indirectly when the user is assigned to a workspace role (contributor or higher) or granted when publishing a Power BI app or sharing a Power BI item. The managed self-service BI scenario explores the reuse of shared semantic models further.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

> **Note**
>
> For team, departmental, and enterprise BI scenarios, a centralized data gateway in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements.

Related content

In the next article in this series, learn more about the importance of reusing semantic models in the managed self-service BI scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Managed self-service BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

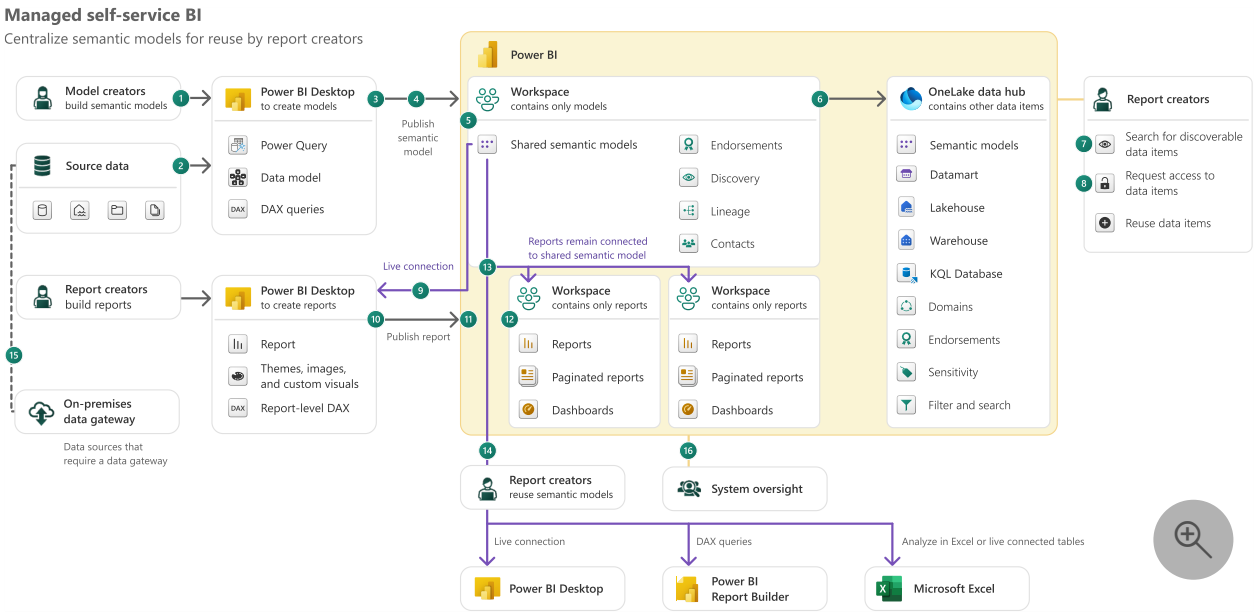
As described in the Fabric adoption roadmap, *managed self-service BI* is characterized by a blended approach that emphasizes *discipline at the core* and *flexibility at the edge*. The data architecture is usually maintained by a single team of centralized BI experts, while reporting responsibility belongs to creators within departments or business units.

Usually, there are many more report creators than semantic model creators. These report creators can exist in any area of the organization. Because self-service report creators often need to quickly produce content, a blended approach allows them to focus on producing reports that support timely decision-making without the additional effort of creating a semantic model.

> **Note**
>
> The managed self-service BI scenario is the first of the self-service BI scenarios. For a complete list of the self-service BI scenarios, see the Power BI usage scenarios article.
>
> For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support managed self-service BI. The primary objective is for many report creators to reuse centralized shared semantic models. To accomplish that, this scenario focuses on decoupling the model development process from the report creation process.



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Semantic model creators develop models by using Power BI Desktop. For semantic models that are intended for reuse, it's common (but not required) for creators to belong to a centralized team that supports users across organizational boundaries (such as IT, enterprise BI, or the Center of Excellence).
2	Power BI Desktop connects to data from one or more data sources.
3	Data model development is done in Power BI Desktop. Additional effort is made to create a well-designed and user-friendly model because it will be used as a data source by many self-service report creators. Model creators can use DAX queries to develop and explore the model during development.
4	When ready, semantic model creators publish their Power BI Desktop file (.pbix) or Power BI project file (.pbip) *that contains only a model* to the Power BI service.
5	The semantic model is published to a workspace dedicated to storing and securing shared semantic models. Since the semantic model is intended for reuse, it's endorsed (certified or promoted, as appropriate). The semantic model is also marked as discoverable to further encourage its reuse. The lineage view in the Power BI service can be used to track

Item	Description
	dependencies that exist between Power BI items, including reports connected to the semantic model.
6	Semantic model discovery in the OneLake catalog is enabled because the semantic model is marked as discoverable. Discoverability allows the existence of a semantic model to be visible in the OneLake catalog by other content creators who are looking for data.
7	Report creators use the OneLake catalog in the Fabric portal to search for discoverable data items, like semantic models.
8	If report creators don't have permission, they can request Build permission on the data items. This starts a workflow to request Build permission from an authorized approver. When approved, the report creator can reuse data items to create new reports.
9	Report creators create new reports by using Power BI Desktop. Reports use a live connection to a shared semantic model.
10	Report creators develop reports in Power BI Desktop. In addition to the report, the report creators can use themes, images, and custom visuals, and they can create report-level measures.
11	When ready, report creators publish their Power BI Desktop file to the Power BI service.
12	Reports are published to a workspace dedicated to storing and securing reports and dashboards.
13	Published reports remain connected to the shared semantic models that are stored in a different workspace. Any changes to the shared semantic model affect all reports connected to it.
14	Other self-service report creators can author new reports using the existing shared semantic model. Report creators can choose to use Power BI Desktop, Power BI Report Builder, or Excel.
15	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
16	Power BI administrators oversee and monitor activity in the Power BI service.

Key points

The following are some key points to emphasize about the managed self-service BI scenario.

Shared semantic model

The key aspect of making *managed self-service BI* work is to minimize the number of semantic models. This scenario is about shared semantic models that help achieve a *single version of the truth*.

> **Note**
>
> For simplicity, the scenario diagram depicts just one shared semantic model. However, it's not usually practical to model all organizational data in a single semantic model. The other extreme is to create a new semantic model for every report, as less experienced content creators often do. The goal of managed self-service BI is to find the right balance, leaning toward relatively few semantic models and creating new semantic models when it makes sense to do so.

Decouple semantic model and reports

When the semantic model is decoupled from reports, it facilitates the separation of effort and responsibility. A shared semantic model is commonly maintained by a centralized team (like IT, BI, or Center of Excellence), while reports are maintained by subject matter experts in the business units. However, that's not required. For example, this pattern can be adopted by any content creator that wants to achieve reusability.

> **Note**
>
> For simplicity, dataflows aren't depicted in the scenario diagram. To learn about dataflows, see the self-service data preparation scenario.

Semantic model endorsement

Because shared semantic models are intended for reuse, it's helpful to endorse them. A *certified* semantic model conveys to report creators that the data is trustworthy and meets the organization's quality standards. A *promoted* semantic model highlights that the semantic model owner believes the data is valuable and worthwhile for others to use.

> **Tip**
>
> It's a best practice to have a consistent, repeatable, rigorous process for endorsing content. Certified content should indicate that data quality has been validated. It should also follow change management rules, have formal support, and be fully

documented. Because certified content has passed rigorous standards, the expectations for trustworthiness are higher.

Semantic model discovery

The OneLake catalog helps report creators find, explore, and use semantic models across the organization. In addition to semantic model endorsement, enabling semantic model discovery is critical for promoting its reuse. A discoverable semantic model is visible in the OneLake catalog for report creators who are searching for data.

> **Note**
>
> If a semantic model isn't configured to be discoverable, only Power BI users with Build permission can find it.

Request semantic model access

A report creator might find a semantic model in the OneLake catalog that they want to use. If they don't have Build permission for the semantic model, they can request access. Depending on the request access setting for the semantic model, an email will be submitted to the semantic model owner or custom instructions will be presented to the person who is requesting access.

Live connection to the shared semantic model

A Power BI Desktop live connection connects a report to an existing semantic model. Live connections avoid the need to create a new data model in the Power BI Desktop file.

> **Important**
>
> When using a live connection, all data that the report creator needs must reside within the connected semantic model. However, the **customizable managed self-service BI** scenario describes how a semantic model can be extended with additional data and calculations.

Publish to separate workspaces

There are several advantages to publishing reports to a workspace different from where the semantic model is stored.

First, there's clarity on who's responsible for managing content in which workspace. Second, report creators have permissions to publish content to a reporting workspace (via workspace admin, member, or contributor roles). However, they only have Read and Build permissions for specific semantic models. This technique allows row-level security (RLS) to take effect when necessary for users assigned to the viewer role.

> **Important**
>
> When you publish a Power BI Desktop report to a workspace, the RLS roles are applied to members who are assigned to the viewer role in the workspace. Even if viewers have Build permission to the semantic model, RLS still applies. For more information, see **Using RLS with workspaces in Power BI**.

Dependency and impact analysis

When a shared semantic model is used by many reports, those reports can exist in many workspaces. The lineage view helps identify and understand the downstream dependencies. When planning a semantic model change, first perform impact analysis to understand which dependent reports might require editing or testing.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model.

> **Note**
>
> For managed self-service BI scenarios, a centralized **data gateway** in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements. With a managed self-service BI scenario, it's particularly helpful to track usage of shared semantic models. A high report-to-semantic model ratio indicates good reuse of semantic models.

Related content

In the next article in this series, learn about ways to customize and extend a shared semantic model to meet additional types of requirements.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Customizable managed self-service BI

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

As described in the Fabric adoption roadmap, *managed self-service BI* is characterized by a blended approach that emphasizes *discipline at the core* and *flexibility at the edge*. The data architecture is usually maintained by a single team of centralized BI experts, while reporting responsibility belongs to creators within departments or business units.

However, when the core data architecture doesn't include all data required, semantic model creators can extend, personalize, or customize existing shared semantic models. New specialized semantic models can be created that meet business requirements not met by existing centrally delivered semantic models. Importantly, there's no duplication of core data. This usage scenario is called *customizable managed self-service BI*.

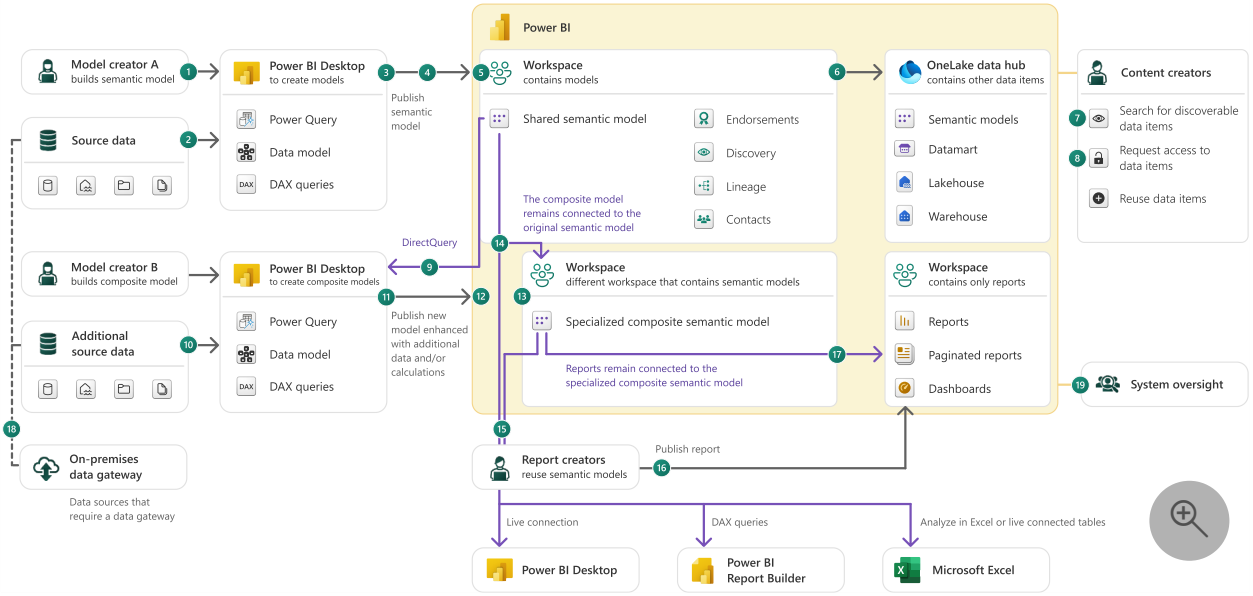
> **Note**
>
> This customizable managed self-service BI scenario is the second of the self-service BI scenarios. This scenario builds upon what can be done with a centralized shared semantic model (that was introduced in the managed self-service BI scenario). A list of all scenarios can be found in the Power BI usage scenarios article.
>
> For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components to support customizable managed self-service BI. The primary focus is on providing content creators in the business units with the ability to create a specialized data model by extending an existing shared semantic model. The goal is to

achieve reusability whenever possible and to allow flexibility to meet additional analytical requirements.

Customizable managed self-service BI
Create composite semantic models that extend and customize other semantic models



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Semantic model creator A develops a model using Power BI Desktop. For a semantic model that's intended for reuse, it's common (but not required) for the creator to belong to a centralized team that supports users across organizational boundaries (such as IT, enterprise BI, or the Center of Excellence).
2	Power BI Desktop connects to data from one or more data sources.
3	Data model development is done in Power BI Desktop. Additional effort is made to create a well-designed and user-friendly model so it can be used as a data source by many self-service report creators. Model creators can use DAX queries to develop and explore the model during development.
4	When ready, model creator A publishes their Power BI Desktop file (.pbix) or Power BI project file (.pbip) *that contains only a model* to the Power BI service.

Item	Description
5	The semantic model is published to a workspace dedicated to storing and securing shared semantic models. Since the semantic model is intended for reuse, it's endorsed (certified or promoted, as appropriate). The semantic model is also marked as discoverable to further encourage its reuse. The lineage view in the Power BI service can be used to track dependencies that exist between Power BI items.
6	Data discovery in the OneLake catalog is enabled because the semantic model is marked as discoverable. Discoverability allows the existence of a semantic model to be visible in the OneLake catalog by other Power BI content creators who are looking for data.
7	Content creators use the OneLake catalog in the Fabric portal to search for discoverable data items, like semantic models.
8	If content creators have permission, they can request Build permission on data items. This starts a workflow to request Build permission from an authorized approver. Once they have permission, content creators can reuse the data items to build new solutions.
9	In Power BI Desktop, model creator B creates a live connection to the original shared semantic model that's located in the Power BI service. Since the intention is to extend and customize the original semantic model, the live connection is converted to a DirectQuery model. This action results in a local model in the Power BI Desktop file.
10	Power BI Desktop connects to data from additional data sources. The goal is to augment the shared semantic model so that additional analytical requirements are met by the new specialized composite semantic model.
11	Relationships are created in Power BI Desktop between the existing tables (from the shared semantic model, also known as the remote model) and new tables just imported (stored in the local model). Additional calculations and modeling work is done in Power BI Desktop to complete the design of the specialized composite model.
12	When ready, semantic model creator B publishes their .pbix or .pbip file to the Power BI service.
13	The new specialized composite semantic model is published to a workspace dedicated to storing and securing semantic models that are owned and managed by the department.
14	The specialized semantic model remains connected to the original Power BI shared semantic model. Any changes to the original shared semantic model will affect downstream specialized composite semantic models that have dependency on it.
15	Other self-service report creators can author new reports connected to the specialized composite semantic model. Report creators can choose to use Power BI Desktop, Power BI Report Builder, or Excel.
16	Reports are published to a workspace dedicated to storing and securing reports and dashboards.
17	Published reports remain connected to the specialized semantic model that's stored in a different workspace. Any changes to the specialized semantic model affect all reports

Item	Description
	connected to it.
18	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
19	Fabric administrators oversee and monitor activity in the Fabric portal.

Key points

The following are some key points to emphasize about the customizable managed self-service BI scenario.

Shared semantic model

The key aspect of making *managed self-service BI* work is to minimize the number of semantic models. This scenario depicts a shared semantic model that contributes towards achieving a *single version of the truth*.

> **Note**
>
> For simplicity, the scenario diagram depicts just one shared semantic model. However, it's not usually practical to model all organizational data in a single semantic model. The other extreme is to create a new semantic model for every report, as less experienced content creators often do. The goal is to find the right balance, leaning toward relatively few semantic models and creating new semantic models when it makes sense to do so.

Augment the initial shared semantic model

Sometimes self-service creators need to augment an existing semantic model with, for instance, additional data that's specific to their department. In this case, they can use DirectQuery connections to Power BI semantic models. This feature allows for an ideal balance of self-service enablement while taking advantage of the investment in centrally managed data assets. The scenario diagram depicts a DirectQuery connection. The act of converting a live connection to a DirectQuery connection creates a local model that allows new tables to be added. Relationships can be created between tables from the original shared semantic model (the remote model) and new tables just added (the local model). Additional calculations and data modeling can be done to customize the new data model.

> **Tip**
>
> This scenario highlights reusing a shared semantic model. However, sometimes there are situations when data modelers want to limit the creation of downstream data model. In that case, they can enable the **Discourage DirectQuery connections** property in the Power BI Desktop settings.

Semantic model endorsement

Because shared semantic models are intended for reuse, it's helpful to endorse them. A *certified* semantic model conveys to report creators that the data is trustworthy and meets the organization's quality standards. A *promoted* semantic model highlights that the semantic model owner believes the data is valuable and worthwhile for others to use.

> **Tip**
>
> It's a best practice to have a consistent, repeatable, rigorous process for endorsing content. Certified content should indicate that data quality has been validated. It should also follow change management rules, have formal support, and be fully documented. Because certified content has passed rigorous standards, the expectations for trustworthiness are higher.

Semantic model discovery

The OneLake catalog helps report creators find, explore, and use semantic models across the organization. In addition to semantic model endorsement, enabling semantic model discovery is critical for promoting its reuse. A discoverable semantic model is visible in the OneLake catalog for report creators who are searching for data.

> **Note**
>
> If a semantic model isn't configured to be discoverable, only Power BI users with Build permission can find it.

Request semantic model access

A report creator might find a semantic model in the OneLake catalog that they want to use. If they don't have Build permission for the semantic model, they can request access. Depending on the request access setting for the semantic model, an email will be submitted to the semantic model owner or custom instructions will be presented to the person who is requesting access.

Publish to separate workspaces

There are several advantages to publishing reports to a workspace different from where the semantic model is stored.

First, there's clarity on who's responsible for managing content in which workspace. Second, report creators have permissions to publish content to a reporting workspace (via workspace admin, member, or contributor roles). However, they only have Read and Build permissions for specific semantic models. This technique allows row-level security (RLS) to take effect when necessary for users assigned to the viewer role.

Dependency and impact analysis

When a shared semantic model is used by other semantic models or reports, those dependent objects can exist in many workspaces. The lineage view helps identify and understand the downstream dependencies. When planning a semantic model change, first perform impact analysis to understand which semantic models or reports should be edited or tested.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model.

> **Note**
>
> For customizable managed self-service BI scenarios, a centralized **data gateway** in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements. With a customizable managed self-service BI scenario, it's particularly helpful to track usage of the original shared semantic model as well as dependent semantic models.

Related content

In the next article in this series, learn about reusing data preparation work with dataflows in the self-service data preparation scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Self-service data preparation

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

Data preparation (sometimes referred to as ETL, which is an acronym for *Extract, Transform, and Load*) often involves a significant amount of work depending on the quality and structure of source data. The *self-service data preparation* usage scenario focuses on the reusability of data preparation activities by business analysts. It achieves this goal of reusability by relocating the data preparation work from Power Query (within individual Power BI Desktop files) to Power Query Online (using a Power BI dataflow). The centralization of the logic helps achieve a *single source of the truth* and reduces the level of effort required by other content creators.

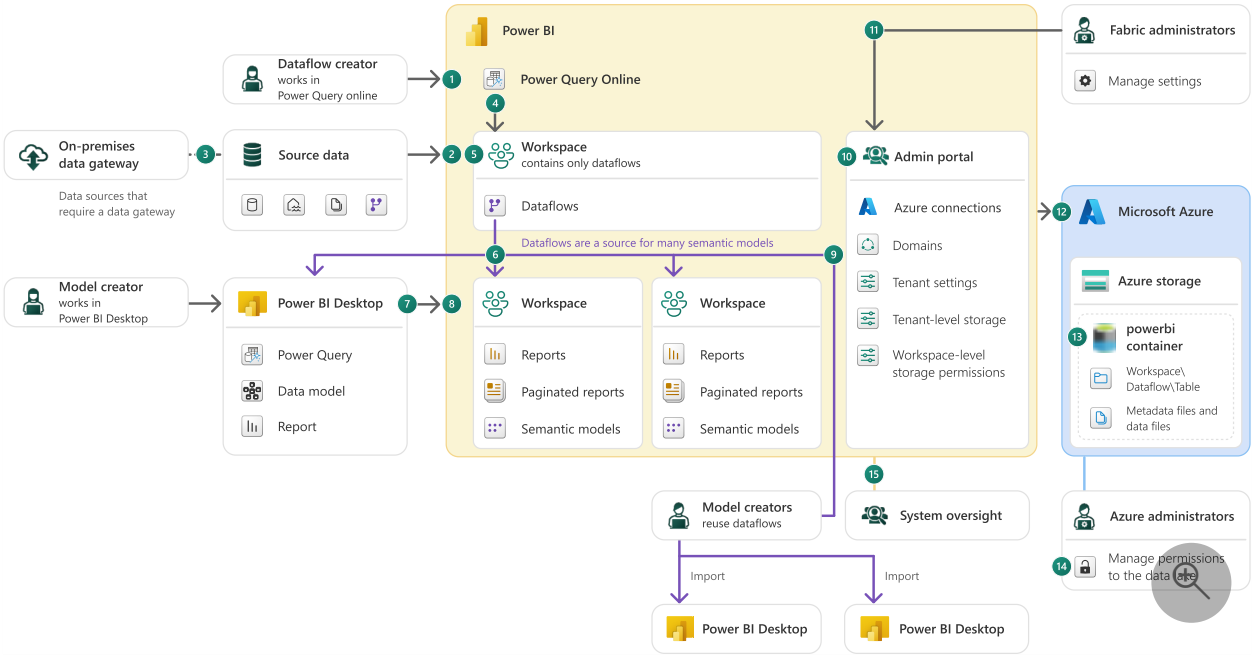
Dataflows are created by using Power Query Online in one of several tools: the Power BI service, Power Apps, or Dynamics 365 Customer Insights. A dataflow created in Power BI is referred to as an *analytical* dataflow. Dataflows created in Power Apps can either be one of two types: *standard* or *analytical*. This scenario only covers using a Power BI dataflow that's created and managed within the Power BI service.

> **Note**
>
> The self-service data preparation scenario is one of the self-service BI scenarios. For a complete list of the self-service scenarios, see the Power BI usage scenarios article.
>
> For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support self-service data preparation. The primary focus is on creating a dataflow in Power Query Online that becomes a source of data for multiple semantic models. The goal is for many semantic models to leverage the data preparation that's done once by the dataflow.

Self-service data preparation
Dataflows for centralizing data cleaning and transformation work



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	The dataflow creator develops a collection of tables within a Power BI dataflow. For a dataflow that's intended for reuse, it's common (but not required) for the creator to belong to a centralized team that supports users across organizational boundaries (such as IT, enterprise BI, or the Center of Excellence).
2	The dataflow connects to data from one or more data sources.

Item	Description
3	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network. These gateways are used both for authoring the dataflow in Power Query Online, which is a web-based version of Power Query, and refreshing the dataflow.
4	Dataflows are developed using Power Query Online. The familiar Power Query interface in Power Query Online makes the transition from Power BI Desktop simple.
5	The dataflow is saved as an item in a workspace that's dedicated to storing and securing dataflows. A dataflow refresh schedule is required to keep the data current (not depicted in the scenario diagram).
6	The dataflow can be reused as a data source by content creators, and by other semantic models that could reside in different workspaces.
7	The semantic model creator develops a new data model by using Power BI Desktop. The semantic model creator can use the full capabilities of Power Query within Power BI Desktop. They can optionally apply other query steps to further transform the dataflow data or merge the dataflow output.
8	When ready, the semantic model creator publishes the Power BI Desktop file (.pbix) that contains the data model to the Power BI service. Refresh for the semantic model is managed separately from the dataflow (not depicted in the scenario diagram).
9	Other self-service semantic model creators can create new data models in Power BI Desktop by using the dataflow as a data source.
10	In the Admin portal, Power BI administrators can set up Azure connections to store dataflow data in their Azure Data Lake Storage Gen2 (ADLS Gen2) account. Settings include assigning a tenant-level storage account and enabling workspace-level storage permissions.
11	Power BI administrators manage settings in the Admin portal.
12	By default, dataflows store data using internal storage that's managed by the Power BI service. Optionally, data output by the dataflow can be stored in the organization's ADLS Gen2 account. This type of storage is sometimes called *bring your own data lake*. A benefit of storing dataflow data in the data lake is that it can be accessed and consumed by other BI tools.
13	Dataflow data in ADLS Gen2 is stored within a Power BI-specific container known as *filesystem*. Within this container, a folder exists for each workspace. A subfolder is created for each dataflow, as well as for each table. Power BI generates a snapshot each time the dataflow data is refreshed. Snapshots are self-describing, comprising metadata and data files.
14	Azure administrators manage permissions for the organization's ADLS Gen2 account.

Item	Description
15	Power BI administrators oversee and monitor activity in the Power BI service.

> **Tip**
>
> We recommend that you review the **advanced data preparation** usage scenario too. It builds upon concepts introduced in this scenario.

Key points

The following are some key points to emphasize about the self-service data preparation scenario.

Dataflows

A dataflow comprises a collection of tables (also known as *entities*). All work to create a dataflow is done in Power Query Online. You can create dataflows in multiple products, including Power Apps, Dynamics 365 Customer Insights, and Power BI.

> **Note**
>
> You can't create dataflows in a personal workspace in the Power BI service.

Support semantic model creators

The scenario diagram depicts using a Power BI dataflow to provide prepared data to other self-service semantic model creators.

> **Note**
>
> Semantic model use the dataflow as a data source. A report can't connect directly to a dataflow.

Here are some advantages of using Power BI dataflows:

- Semantic model creators use the same familiar Power Query interface found in Power BI Desktop.

- Data preparation and data transformation logic defined by a dataflow can be reused many times because it's centralized.
- When data preparation logic changes are made to the dataflow, it might not require updating dependent data models. Removing or renaming columns, or changing column data types, will require updating dependent data models.
- Pre-prepared data can easily be made available to Power BI semantic model creators. Reuse is particularly helpful for commonly used tables—especially dimension tables, like date, customer, and product.
- The level of effort required by semantic model creators is reduced because the data preparation work has been *decoupled* from the data modeling work.
- Fewer semantic model creators need direct access to source systems. Source systems can be complex to query and could require specialized access permissions.
- The number of refreshes executed on source systems is reduced because semantic model refreshes connect to dataflows, and not to the source systems from which dataflows extract data.
- Dataflow data represents a snapshot in time, and promotes consistency when used by many semantic models.
- Decoupling data preparation logic into dataflows can help improve semantic model refresh success. If a dataflow refresh fails, semantic models will refresh using the last successful dataflow refresh.

> **Tip**
>
> Create dataflow tables by applying **star schema** design principles. A star schema design is well-suited to creating Power BI semantic models. Also, refine the dataflow output to apply friendly names and use specific data types. These techniques promote consistency in dependent semantic models and helps reduce the amount of work that semantic model creators need to do.

Semantic model creator flexibility

When a semantic model creator connects to a dataflow in Power BI Desktop, the creator isn't limited to using the exact dataflow output. They still have the full functionality of Power Query available to them. This functionality is useful if additional data preparation work is required, or the data requires further transformation.

Dataflow advanced features

There are many design techniques, patterns, and best practices for dataflows that can take them from self-service to enterprise-ready. Dataflows in a workspace that has its

license mode set to **Premium per user**, **Premium capacity**, or **Fabric capacity** can benefit from advanced features.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

> **Note**
>
> One of the **advanced features** is incremental refresh for dataflows. Although incremental refresh for *semantic models* is a Power BI Pro feature, incremental refresh for dataflows is a Premium feature.
>
> To learn more about dataflow advanced features, see the **advanced data preparation** usage scenario.

Dataflow and semantic model refresh

As previously mentioned, a dataflow is a source of data for semantic models. In most cases, multiple data refresh schedules are involved: one for the dataflow and one for each semantic model. Alternatively, it's possible to use DirectQuery from the semantic model to the dataflow, which is a Premium feature (not depicted in the scenario diagram).

Azure Data Lake Storage Gen2

In Microsoft Azure, an ADLS Gen2 account is a specific type of Azure Storage account that has the hierarchical namespace enabled. ADLS Gen2 has performance, management, and security advantages for operating analytical workloads. By default, Power BI dataflows use internal storage, which is a built-in data lake account managed by the Power BI service. Optionally, organizations can *bring their own data lake* by connecting to their organization's ADLS Gen2 account.

Here are some advantages of using the organization's data lake account:

- The data stored by a Power BI dataflow can (optionally) be accessed from the data lake by other users or processes. That's helpful when dataflow reuse occurs beyond Power BI. For example, the data could be accessed by Azure Data Factory.
- The data in the data lake can (optionally) be managed by other tools or systems. In this case, Power BI could consume the data rather than manage it (not depicted in the scenario diagram).

Tenant-level storage

The Azure connections section of the Admin portal includes a setting to configure a connection to an ADLS Gen2 account. Configuring this setting enables *bring your own data lake*. Once set up, you can set workspaces to use that data lake account.

> **Important**
>
> Setting **Azure connections** does not mean that all dataflows in the Power BI tenant are stored in this account by default. In order to use an explicit storage account (instead of internal storage), each workspace must be specifically connected.
>
> It's critical to set the workspace Azure connections *prior to creating any dataflows* in the workspace. The same Azure storage account is used for **Power BI semantic model backups**.

Workspace-level storage

A Power BI administrator can configure a setting to allow workspace-level storage permissions (in the Azure connections section of the Admin portal). When enabled, this setting allows workspace administrators to use a different storage account than the one defined at the tenant-level. Enabling this setting is particularly helpful for decentralized business units who manage their own data lake in Azure.

> **Note**
>
> The **workspace-level storage permission** in the Admin portal applies to all workspaces in the Power BI tenant.

Common Data Model format

The data in an ADLS Gen2 account is stored in the *Common Data Model (CDM) structure*. The CDM structure is a metadata format that dictates how the self-describing schema, as well as the data, is stored. The CDM structure enables semantic consistency in a format that's standardized for sharing data across numerous applications (not depicted in the scenario diagram).

Publish to separate workspaces

There are several advantages to publishing a dataflow to a workspace that's separate from where the dependent semantic models are stored. One advantage is clarity on who's responsible for managing which types of content (if you have different people handling different responsibilities). Another advantage is that specific workspace permissions can be assigned for each type of content.

> **Note**
>
> You can't create dataflows in a personal workspace in the Power BI service.
>
> The **advanced data preparation** usage scenario describes how to set up multiple workspaces to provide better flexibility when supporting enterprise-level self-service creators.

Gateway setup

Typically, an On-premises data gateway is required for connecting to data sources that reside within a private organizational network or a virtual network.

A data gateway is required when:

- Authoring a dataflow in Power Query Online that connects to private organizational data.
- Refreshing a dataflow that connects to private organizational data.

> **Tip**
>
> Dataflows require a centralized **data gateway** in *standard mode*. A gateway in ***personal mode*** isn't supported when working with dataflows.

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements. With a self-service data preparation scenario, it's particularly helpful to track usage of dataflows.

Related content

In the next article in the series, learn about the advanced data preparation usage scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Advanced data preparation

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

Data preparation (sometimes referred to as ETL, which is an acronym for *Extract, Transform, and Load*) activities often involve a large effort. The time, skill, and effort involved with collecting, cleaning, combining, and enriching data depends on the quality and structure of source data.

Investing time and effort in centralized data preparation helps to:

- Enhance reusability and gain maximum value from data preparation efforts.
- Improve the ability to provide consistent data to multiple teams.
- Reduce the level of effort required by other content creators.
- Achieve scale and performance.

The *advanced data preparation* usage scenario expands on the self-service data preparation scenario. Advanced data preparation is about increasing dataflow reuse by multiple users across various teams and for various use cases.

Separate workspaces, organized by dataflow purpose, are helpful when dataflow output is provided to multiple semantic model creators, especially when they are on different teams in the organization. Separate workspaces are also helpful for managing security roles when the people who create and manage dataflows are different from the people consume them.

> **Note**
>
> The advanced data preparation scenario is the second of the data preparation scenarios. This scenario builds upon what can be done with centralized dataflows as described in the self-service data preparation scenario.
>
> The advanced data preparation scenario is one of the self-service BI scenarios. However, a centralized team member can use the techniques in a similar way to

what's described in the managed self-service BI scenario. For a complete list of the self-service scenarios, see the Power BI usage scenarios article.

For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

Scenario diagram

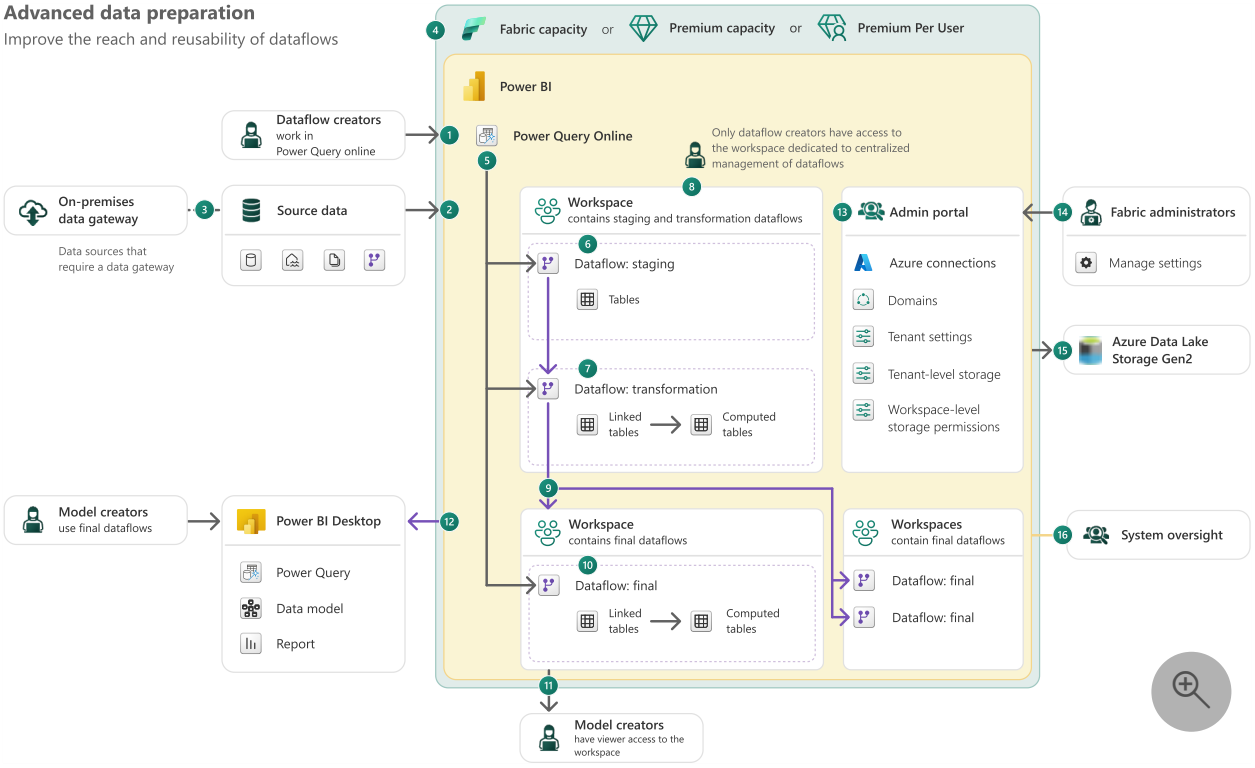
> **Tip**
>
> We recommend that you review the self-service data preparation usage scenario if you're not familiar with it. The advanced self-service data preparation scenario builds upon that scenario.

The focus of this advanced data preparation scenario is on:

- The use of separate dataflows based on purpose: staging, transformation, or final. We recommend using *composable building blocks* to obtain greater reuse, in various combinations, to support specific user requirements. Composable building blocks are described later in this article.
- The use of separate workspaces that support dataflow *creators* or dataflow *consumers*. Data modelers, who consume dataflows, might be on different teams and/or have different use cases.
- The use of linked tables (also known as *linked entities*), computed tables (also known as *computed entities*), and the enhanced compute engine.

> **Note**
>
> Sometimes the terms *semantic model* and *data model* are used interchangeably. Generally, from a Power BI service perspective, it's referred to as *semantic model*. From a development perspective, it's referred to as a *data model* (or *model* for short). In this article, both terms have the same meaning. Similarly, a semantic model creator and a data modeler have the same meaning.

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support the advanced data preparation scenario.



> **Tip**
>
> We encourage you to download the scenario diagram if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	The dataflow creator develops a collection of tables within a dataflow. For a dataflow that's intended for reuse, it's common (but not required) that the creator belongs to a centralized team that supports users across organizational boundaries (such as IT, enterprise BI, or the Center of Excellence).
2	The dataflow connects to data from one or more data sources.
3	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network. These gateways are used both for authoring the dataflow in Power Query Online and refreshing the dataflow.
4	All of the workspaces involved have their license mode set to **Fabric capacity**, **Premium capacity**, **Premium Per User**, or **Embedded**. These license modes allow for the use of linked tables and computed tables across workspaces, which are required in this scenario.

Item	Description
5	Dataflow creators develop dataflows by using Power Query Online, which is a web-based version of Power Query.
6	A *staging dataflow* is created in a workspace that's dedicated to the centralized management of dataflows. A staging dataflow copies the raw data as-is from the source. Few, if any, transformations are applied.
7	A *transformation dataflow* (also known as a *cleansed dataflow*) is created in the same workspace. It sources data by using linked table(s) to the staging dataflow. Computed table(s) include transformation steps that prepare, cleanse, and reshape the data.
8	Dataflow creators have access to manage content in the workspace that's dedicated to the centralized management of dataflows.
9	One or more other workspaces exist that are intended to provide access to the final dataflow, which delivers production-ready data to data models.
10	The *final dataflow* is created in a workspace available to data modelers. It sources data by using linked table(s) to the transformation dataflow. Computed table(s) represent the prepared output that's visible to data modelers who are granted the workspace **viewer** role.
11	Semantic model creators (who consume the dataflow output) have viewer access to the workspace that contains the final dataflow output. Dataflow creators also have access to manage and publish content in the workspace (not depicted in the scenario diagram).
12	Semantic model creators use the *final dataflow* as a data source when developing a data model in Power BI Desktop. When ready, the semantic model creator publishes the Power BI Desktop file (.pbix) that contains the data model to the Power BI service (not depicted in the scenario diagram).
13	Fabric administrators manage settings in the Admin portal.
14	In the Admin portal, Power BI administrators can set up Azure connections to store dataflow data in their Azure Data Lake Storage Gen2 (ADLS Gen2) account. Settings include assigning a tenant-level storage account and enabling workspace-level storage permissions.
15	By default, dataflows store data by using internal storage that's managed by the Power BI service. Optionally, data output by the dataflow can be stored in the organization's ADLS Gen2 account.
16	Fabric administrators oversee and monitor activity in the Fabric portal.

Key points

The following are some key points to emphasize about the advanced data preparation scenario.

Dataflows

A dataflow comprises a collection of tables (also known as *entities*). Each table is defined by a query, which contains the data preparation steps required to load the table with data. All work to create a dataflow is done in Power Query Online. You can create a dataflow in multiple products, including Power Apps, Dynamics 365 Customer Insights, and Power BI.

> **Note**
>
> You can't create dataflows in a personal workspace in the Power BI service.

Types of dataflows

Use of *composable building blocks* is a design principle that allows you to manage, deploy, and secure system components, and then use them in various combinations. Creating modular, self-contained dataflows that are specific to a purpose is a best practice. They help to achieve data reuse and enterprise scale. Modular dataflows are also easier to manage and test.

Three types of dataflows are shown in the scenario diagram: *staging dataflow*, *transformation dataflow*, and *final dataflow*.

Staging dataflow

A staging dataflow (sometimes called a *data extraction dataflow*) copies raw data as-is from the source. Having the raw data extracted with minimal transformation means that downstream transformation dataflows (described next) can use the staging dataflow as their source. This modularity is useful when:

- Access to a data source is restricted to narrow time windows and/or to a few users.
- Temporal consistency is desired to ensure that all downstream dataflows (and related semantic models) deliver data that was extracted from the data source at the same time.
- Reducing the number of queries submitted to the data source is necessary due to source system restrictions or its ability to support analytical queries.

- A copy of the source data is useful for reconciliation processes and data quality verifications.

Transformation dataflow

A transformation dataflow (sometimes called a *cleansed dataflow*) sources its data from linked tables that connect to the staging dataflow. It's a best practice to separate out transformations from the data extraction process.

A transformation dataflow includes all the transformation steps required to prepare and restructure the data. However, there's still a focus on reusability at this layer to ensure the dataflow is suitable for multiple use cases and purposes.

Final dataflow

A final dataflow represents the prepared output. Some additional transformations might occur based on the use case and purpose. For analytics, a star schema table (dimension or fact) is the preferred design of the final dataflow.

Computed tables are visible to data modelers that are granted the workspace **viewer** role. This table type is described in the types of dataflow tables topic below.

> **Note**
>
> Data lakes often have *zones*, like bronze, silver, and gold. The three types of dataflows represent a similar design pattern. To make the best possible data architecture decisions, give thought to who will maintain the data, the expected use of the data, and the skill level required by people accessing the data.

Workspaces for dataflows

If you were to create all dataflows in a single workspace, it would significantly limit the extent of reusability. Using a single workspace also limits the security options available when supporting multiple types of users across teams and/or for different use cases. We recommend using multiple workspaces. They provide better flexibility when you need to support self-service creators from various areas of the organization.

The two types of workspaces shown in the scenario diagram include:

- **Workspace 1**: It stores centrally managed dataflows (sometimes referred to as a *backend workspace*). It contains both the staging and transformation dataflows

because they're managed by the same people. Dataflow creators are often from a centralized team, such as IT, BI, or the Center of Excellence. They should be assigned to either the workspace **admin**, **member**, or **contributor** role.

- **Workspace 2**: It stores and delivers the final dataflow output to consumers of the data (sometimes referred to as a *user workspace*). Semantic model creators are often self-service analysts, power users, or citizen data engineers. They should be assigned to the workspace **viewer** role because they only need to consume the output of the final dataflow. To support semantic model creators from various areas of the organization, you can create numerous workspaces like this one, based on use case and security needs.

> **Tip**
>
> We recommend reviewing ways to **support semantic model creators** as described in the **self-service data preparation** usage scenario. It's important to understand that semantic model creators can still use the full capabilities of Power Query within Power BI Desktop. They can choose to add query steps to further transform the dataflow data or merge the dataflow output with other sources.

Types of dataflow tables

Three types of dataflow tables (also known as *entities*) are depicted in the scenario diagram.

- **Standard table**: Queries an external data source, such as a database. In the scenario diagram, standard tables are depicted in the staging dataflow.
- **Linked table**: References a table from another dataflow. A linked table doesn't duplicate the data. Rather, it allows the reuse of a standard table multiple times for multiple purposes. Linked tables aren't visible to workspace viewers since they inherit permissions from the original dataflow. In the scenario diagram, linked tables are depicted twice:
  - In the transformation dataflow for accessing the data in the staging dataflow.
  - In the final dataflow for accessing the data in the transformation dataflow.
- **Computed table**: Performs additional computations by using a different dataflow as its source. Computed tables allow customizing the output as needed for individual use cases. In the scenario diagram, computed tables are depicted twice:
  - In the transformation dataflow for performing common transformations.
  - In the final dataflow for delivering output to semantic model creators. Since computed tables persist the data again (after the dataflow refresh), data

modelers can access the computed tables in the final dataflow. In this case, data modelers should be granted access with the workspace **viewer** role.

> ⓘ **Note**
>
> There are many design techniques, patterns, and best practices that can take dataflows from self-service to enterprise-ready. Also, dataflows in a workspace that has its license mode set to **Premium per user** or **Premium capacity** can benefit from advanced features. Linked tables and computed tables (also known as *entities*) are two advanced features that are essential for increasing the reusability of dataflows.

Enhanced compute engine

The enhanced compute engine is an advanced feature available with Power BI Premium.

> ⓘ **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

The enhanced compute engine improves performance of linked tables (within the same workspace) that reference (link to) the dataflow. To get maximum benefit from the enhanced compute engine:

- Split out the staging and transformation dataflows.
- Use the same workspace to store the staging and transformation dataflows.
- Apply complex operations that can query fold early in the query steps. Prioritizing foldable operations can help to achieve the best refresh performance.
- Use incremental refresh to reduce refresh durations and resource consumption.
- Perform testing early and frequently during the development phase.

Dataflow and semantic model refresh

A dataflow is a source of data for semantic models. In most cases, multiple data refresh schedules are involved: one for each dataflow and one for each semantic model. Alternatively, it's possible to use DirectQuery from the semantic model to the dataflow, which requires Power BI Premium and the enhanced compute engine (not depicted in the scenario diagram).

Azure Data Lake Storage Gen2

An ADLS Gen2 account is a specific type of Azure storage account that has the hierarchical namespace enabled. ADLS Gen2 has performance, management, and security advantages for operating analytical workloads. By default, Power BI dataflows use internal storage, which is a built-in data lake account managed by the Power BI service. Optionally, organizations can *bring their own data lake* by connecting to an ADLS Gen2 account in their organization.

Here are some advantages of using your own data lake:

- Users (or processes) can directly access the dataflow data stored in the data lake. That's helpful when dataflow reuse occurs beyond Power BI. For example, Azure Data Factory could access the dataflow data.
- Other tools or systems can manage the data in the data lake. In this case, Power BI could consume the data rather than manage it (not depicted in the scenario diagram).

When using linked tables or computed tables, make sure that each workspace is assigned to the same ADLS Gen2 storage account.

> ⓘ **Note**
>
> Dataflow data in ADLS Gen2 is stored within a Power BI-specific container. This container is depicted in the **self-service data preparation** usage scenario diagram.

Admin portal settings

There are two important settings to manage in the Admin portal:

- **Azure connections**: The Azure connections section of the Admin portal includes a setting to set up a connection to an ADLS Gen2 account. This setting allows a Power BI administrator to *bring your own data lake* to dataflows. Once configured, workspaces can use that data lake account for storage.

- **Workspace-level storage**: A Power BI administrator can set workspace-level storage permissions. When enabled, the setting allows workspace administrators to use a different storage account to the one set at tenant-level. Enabling this setting is helpful for decentralized business units that manage their own data lake in Azure.

Gateway setup

Typically, an On-premises data gateway is required for connecting to data sources that reside within a private organizational network or a virtual network.

A data gateway is required when:

- Authoring a dataflow in Power Query Online that connects to private organizational data.
- Refreshing a dataflow that connects to private organizational data.

> **Tip**
>
> Dataflows require a centralized **data gateway** in *standard mode*. A gateway in ***personal mode*** isn't supported when working with dataflows.

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. The activity log is also valuable for supporting governance efforts, security audits, and compliance requirements. In the advanced data preparation scenario, the activity log data is helpful to track the management and use of dataflows.

Related content

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Prototyping and sharing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

As described in the Fabric adoption roadmap, exploration, experimentation, and obtaining useful feedback from a small group of users is the purpose of phase 1 of solution adoption.

A prototype—or proof of concept (POC)—is a Power BI solution that's intended to address unknowns and mitigate risk. This solution can be shared with others to get feedback during development iterations. The solution might be a temporary, short-lived solution, or it might ultimately evolve into a solution that's fully validated and released. Creating a prototype is commonly done for departmental BI and enterprise BI scenarios (and can occasionally be done for team BI scenarios).

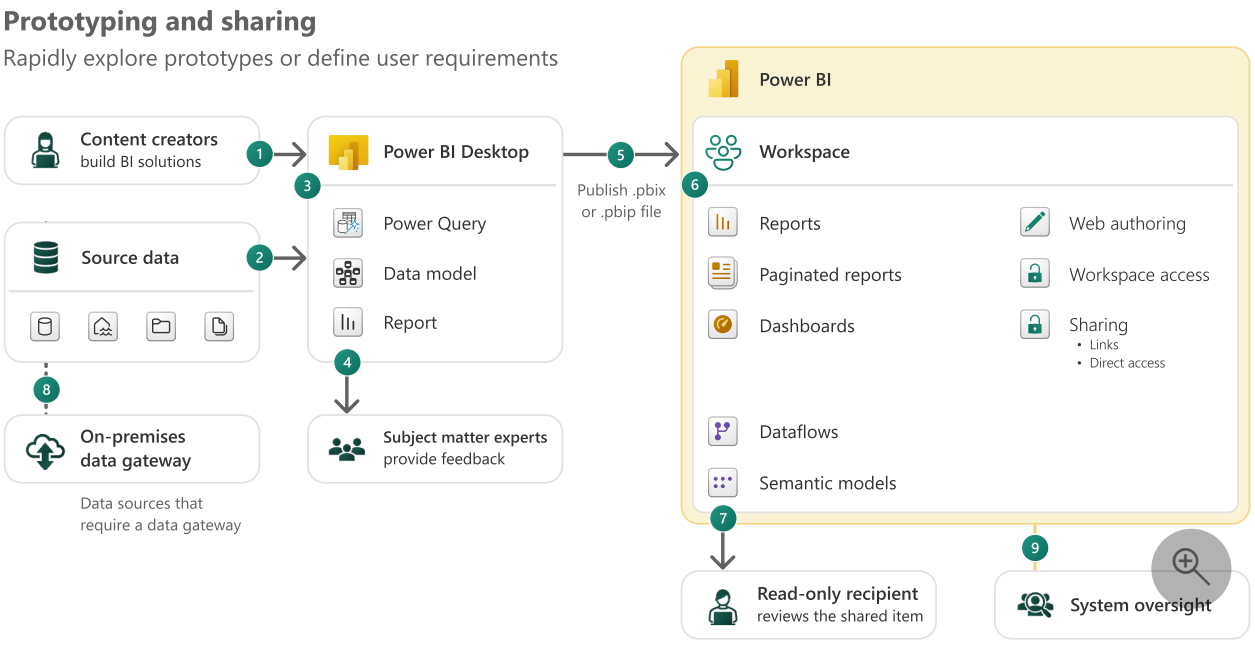
Prototyping often occurs naturally during self-service BI development efforts. Or a prototype might be a small project that has specific goals and a scope.

> **Note**
>
> The prototyping and sharing scenario is one of the self-service BI scenarios. For a complete list of the self-service scenarios, see the Power BI usage scenarios article.
>
> For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components to support prototyping activities. The focus is on using Power

BI Desktop during an interactive prototyping session. Focus can also be on sharing in the Power BI service when additional feedback is needed from subject matter experts.



> **Tip**
>
> We encourage you to [download the scenario diagram](powerbi-implementation-planning-usage-scenario-diagrams.md#prototyping-and-sharing if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Power BI content creators develop BI solutions using Power BI Desktop.
2	Power BI Desktop connects to data from one or more data sources. Queries and data mashups, which combine multiple sources, are developed in the Power Query Editor.
3	Data model development and report creation are done in Power BI Desktop. The purpose is to help team members understand the meaning and significance of data by placing it in a visual context.
4	Subject matter experts provide feedback during an interactive prototyping session. Based on feedback from the subject matter experts (and other team members), content creators make iterative improvements directly to the BI solution.

Item	Description
5	If desired, content creators publish their Power BI Desktop file (.pbix) of Power BI project file (.pbip) to the Power BI service. Publication of prototyping solutions to the Power BI service is optional.
6	The content is published to a non-production workspace. Its primary purpose is to provide a development area that enables review by team members.
7	An individual report is shared with a colleague to provide read-only permissions to the report (and its underlying data). The sharing operation can be done with a sharing link or direct access sharing. Sharing can be advantageous for a prototyping solution to provide temporary access during the feedback process.
8	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
9	Power BI administrators oversee and monitor activity in the Power BI service. A development workspace (containing non-production and prototyping solutions) is usually governed to a much lesser extent than a production workspace.

Key points

The following are some key points to emphasize about the prototyping and sharing scenario.

Interactive prototyping sessions

Interactive prototyping sessions are valuable to get immediate feedback when exploring user requirements, validating calculations, clarifying visual layout needs, validating user experience, and confirming report presentation. Use Power BI Desktop during prototyping sessions that are interactively conducted with subject matter experts.

Power BI service

Publishing prototyping solutions to the Power BI service is optional. It can be useful when there's a need to share preliminary results for feedback and decision-making purposes.

> **Tip**
>
> Prototyping solutions should be clearly separated from other production content so that consumers have proper expectations for a non-production solution. For example, consumers of a prototype report might not expect it to include all the

data or be refreshed on a schedule. A prototype report shouldn't be used for business decisions until it's fully validated, finalized, and published to a production workspace.

Workspace

A development workspace is appropriate in this scenario since it involves working with a small team BI collaboration scenario (rather than a personal workspace as described in the personal BI scenario). Once the solution is finalized and fully tested, it can be quickly promoted to a production workspace (as described in the self-service content publishing scenario).

Sharing reports and dashboards

The scenario diagram depicts sharing directly to a recipient (rather than workspace roles or using a Power BI app). Using the sharing feature is appropriate for collaboration scenarios when colleagues work closely together in an informal way. Sharing is useful in this situation because it's limited to a small number of colleagues who need to review and provide feedback on the prototyped solution.

Tip

Individual item sharing should be done infrequently. Since sharing is configured per individual items in a workspace, it's more tedious to maintain and increases the risk of error. A valid alternative to sharing (not depicted in the scenario diagram) is to use **workspace roles** (described in the **team BI** scenario). Workspace roles work best when colleagues need access to *all items* in a workspace.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

Note

For team, departmental, and enterprise BI scenarios, a centralized **data gateway** in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and detect risky activities. Auditing and governance requirements are typically less stringent for prototyping and personal BI scenarios.

Related content

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Self-service content publishing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

When analytical solutions are critical to the organization, it's important to ensure content in the Power BI service is stable and reliable for consumers. IT teams often solve this problem by working in multiple environments:

- In the **development** environment, content creators and owners make changes and improvements to the solution. When these changes are ready for broader review, the solution is deployed (sometimes known as *promoted*) to the test environment.
- In the **test** environment, reviewers validate the changes made to the solution. This review can involve validating the solution functionality and data. When the review is complete, the solution is deployed to the production environment.
- The **production** environment is where consumers view and interact with the released solution.

This structured approach ensures that content creators, owners, and reviewers can make and validate changes without negatively affecting consumers.

Using methodical and disciplined lifecycle management processes reduces errors, minimizes inconsistencies, and improves the user experience for consumers. Content creators and owners can use Power BI deployment pipelines for *self-service content publishing*. Deployment pipelines simplify the process and improve the level of control when releasing new content.

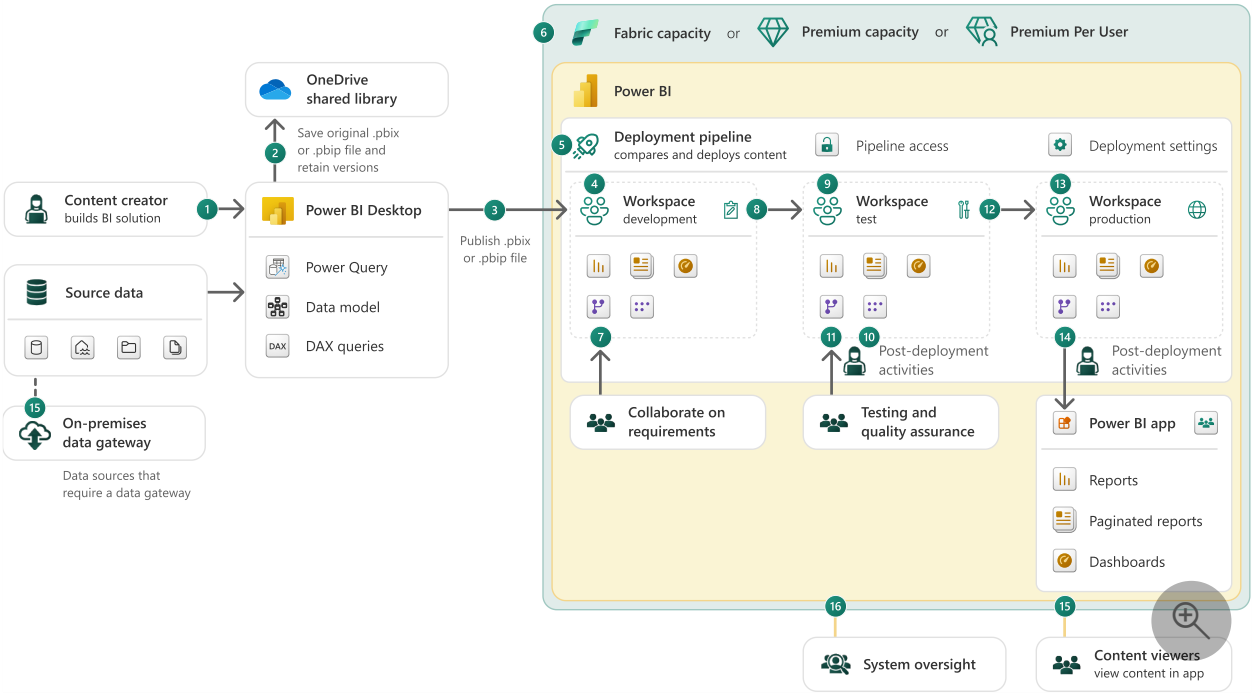
> **Note**
>
> This self-service content publishing scenario is one of the content management and deployment scenarios. For a complete list of the self-service scenarios, see the Power BI usage scenarios article.
>
> For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those

articles first.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components to support self-service content publishing. The focus is on use of a Power BI deployment pipeline for promoting content through development, test, and production workspaces.

Self-service content publishing
Publish content to development, test, and production by using deployment pipelines



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	The Power BI content creator develops a BI solution using Power BI Desktop.

Item	Description
2	The Power BI Desktop file (.pbix) of Power BI project file (.pbip) is saved to a shared library in OneDrive. The content creator retains versions of these files in OneDrive.
3	When ready, the content creator publishes the Power BI Desktop file to the Power BI service.
4	Content is published to a workspace that's dedicated to **development**.
5	A deployment pipeline administrator sets up the Power BI deployment pipeline with three stages: development, test, and production. Each stage aligns to a separate workspace in the Power BI service. Deployment settings and access are set up for the deployment pipeline.
6	The development (or test) workspace is set to **Fabric capacity**, **Premium capacity**, **Premium Per User**, or **Embedded** license mode. Power BI deployment pipelines are a feature available only in workspaces with these license modes.
7	Content creators and owners collaborate in the development workspace to ensure all requirements are met.
8	When the development content is ready, the deployment pipeline compares the content between the development and test stages.
9	Some, or all, Power BI items are deployed to a workspace that's dedicated to testing.
10	Once the deployment pipeline has completed its deployment, the content creator manually performs post-deployment activities for the test workspace. Activities can include configuring scheduled data refresh or publishing a Power BI app for the test workspace.
11	Quality assurance, data validations, and user acceptance testing occur by reviewers of the test workspace.
12	When the test content is fully validated, the deployment pipeline compares the content between the test and production stages.
13	Some, or all, Power BI items are deployed to a workspace that's dedicated to production. For a production workspace, **Fabric capacity** or **Premium capacity** license mode is often more appropriate when there's a large number of read-only consumers.
14	After the deployment pipeline completes deployment, content creators can manually perform post-deployment activities. Activities can include configuring scheduled data refresh or publishing a Power BI app for the production workspace.
15	Content viewers access the content using the production workspace or a Power BI app.
16	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.

Item	Description
17	Fabric administrators oversee and monitor activity in the Fabric portal. Content that's deemed critical enough to have separate development, test, and production workspaces could be subject to stricter governance requirements than less critical content.

> **Tip**
>
> We recommend that you reviewing the **advanced data model management** usage scenario as too. It builds upon concepts introduced in this scenario.

Key points

The following are some key points to emphasize about the self-service content publishing scenario.

Deployment pipeline

A deployment pipeline consists of three stages: development, test, and production. A single workspace is assigned to each stage in the deployment pipeline. Power BI items that are supported by deployment pipelines are published (or cloned) from one workspace to another when a deployment occurs. Once testing and validations are complete, the deployment pipeline can be reused many times to promote content quickly. The deployment pipeline interface is easy to implement for content creators who don't have the skills or desire to use code-based deployments (use of the Power BI REST APIs are described in the enterprise content publishing scenario).

> **Note**
>
> Publishing content using a deployment pipeline is known as a *metadata-only deployment*. In this case, data isn't overwritten or copied to the target workspace. A data refresh is usually required once the deployment completes—see the **post-deployment activities** topic below.

Deployment process

It's a best practice to consider the entire workspace content as an *analytical package* that can be deployed together as a unit. Therefore, it's important to have clarity on the purpose and expectations of each workspace. Although a selective deployment of

specific Power BI items is possible, it's more efficient and less risky when a deployment represents a logical unit of content.

> **Tip**
>
> Plan for how urgent issues will be handled, in addition to planned deployments. If an immediate fix is required, still follow the **standard practice** of propagating all changes from development through to test and production using the deployment pipeline.

Permissions model

Spend time planning the permissions model. Full flexibility for applying different workspace roles (between development, test, and production) is supported. As depicted in the scenario diagram, it's common to assign the following workspace permissions:

- **Development workspace**: Limit access to a team of content creators and owners who collaborate together.
- **Test workspace**: Limit access to reviewers involved with quality assurance, data validations, and user acceptance testing activities.
- **Production workspace**: Grant viewer access to content consumers of the Power BI app (and the workspace, when appropriate). Limit access to those who need to manage and publish production content, involving the fewest number of users possible.

> **Note**
>
> Most content consumers are unaware of the development and test workspaces.

Access for deployment pipeline

Pipeline user permissions (for who can deploy content with a deployment pipeline) are managed separately from the workspace roles. Access to both the workspace and the deployment pipeline are required for the users conducting a deployment. Relevant Premium permissions are also required.

When possible, it's recommended that the existing content creator or owner conduct the deployments. In some situations, permissions are more restricted for the production workspace. In that case, it might be appropriate to coordinate the production deployment with someone else who has permission to deploy to production.

Pipeline users who are assigned to the workspace member (or admin) role are allowed to compare stages and deploy content. Assigning pipeline users to this role minimizes permissions issues and allows for a smoother deployment process.

> **Tip**
>
> Keep in mind that workspace roles are set separately for development, test, and production. However, pipeline access is set once for the entire pipeline.

Power BI Premium licensing

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Power BI deployment pipelines are a Premium feature. There are various ways to obtain licensing, depending on whether the content is used for development, test, or production purposes. The scenario diagram depicts use of a Premium P SKUs such as P1, P2, P3, P4, or P5 for the production workspace, and a Power BI Premium Per User (PPU) user-based Premium license for the development and test workspaces. Using PPU licensing for workspaces with very few users (as depicted in the scenario diagram) is a cost-effective way to use Premium features, while keeping them separate from the Premium capacity that's assigned for production workloads.

Deployment settings

Data source rules and parameter rules are available for dynamically managing values that differ between development, test, and production. Use of deployment settings are an effective way to reduce effort and the risk of errors.

Post-deployment activities

Purposefully, certain properties aren't copied to the target workspace during a deployment. Several key post-deployment activities include:

- **Data refresh**: Data isn't copied from the source workspace to the target workspace. Publishing from a deployment pipeline is always a metadata-only deployment. Therefore, a data refresh is usually required after deploying to a target workspace. For first-time deployments, the data source credentials or gateway connectivity (as appropriate) must be configured as well.
- **Apps**: Power BI apps aren't published automatically by deployment pipelines.
- **Access roles, sharing permissions, and app permissions**: Permissions aren't overwritten during a deployment.
- **Workspace properties**: Properties, such as contacts and the workspace description, aren't overwritten during a deployment.
- **Power BI item properties**: Certain Power BI item properties, such as sensitivity labels, might be overwritten during a deployment in certain circumstances.
- **Unsupported Power BI items**: Additional manual steps might need to be taken for Power BI items that aren't supported by the deployment pipeline.

> ⊗ **Caution**
>
> There isn't a rollback process once a deployment has occurred with a deployment pipeline. Consider carefully what change management processes and approvals are required in order to deploy to the production workspace.

OneDrive storage

The scenario diagram depicts using OneDrive for storing the source Power BI Desktop files. The goal is to store the source files in a location that is:

- Appropriately secured to ensure only publishers can access the source files. A shared library (rather than a personal library) is a good choice.
- Backed up frequently so the files are safe from loss.
- Versioned when changes occur, to allow for a rollback to an earlier version.

> 💡 **Tip**
>
> If a OneDrive location is **synchronized to a workspace**, configure it only for the development workspace.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a Power BI Desktop file is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

When working with multiple environments, it's common to configure development, test, and production connections to use different source systems. In this case, use data source rules and parameter rules to manage values that differ between environments.

> **Note**
>
> A centralized data gateway in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand deployment activities that occur.

Related content

In the next article in the series, learn about the advanced data modeling usage scenario.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Enterprise content publishing

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

When content creators collaborate to deliver analytical solutions that are important to the organization, they must ensure timely and reliable delivery of content to consumers. Technical teams address this challenge by using a process called *DevOps*. DevOps allows teams to automate and scale processes by adopting practices that improve and accelerate delivery.

> **Note**
>
> Data teams that address the same challenges might also practice *DataOps*. DataOps builds upon DevOps principles, but DataOps includes additional practices specific to data management, such as data quality assurance and governance. We refer to DevOps in this article, but be aware that the underlying principles can also apply to DataOps.

Content creators and consumers benefit from several advantages when adopting DevOps practices to publish Power BI content. The following points are a high-level overview of how this process works.

1. **Develop content and commit work to a remote repository**: Content creators develop their solution on their own machine. They commit and save their work to a *remote repository* at regular intervals during development. A remote repository contains the latest version of the solution, and it's accessible to the entire development team.
2. **Collaborate and manage content changes by using version control**: Other content creators can make revisions to the solution by creating a *branch*. A branch is copy of a remote repository. When these revisions are ready and approved, the branch is merged with the latest version of the solution. All revisions to the solution are tracked. This process is known as *version control* (or *source control*).

3. **Deploy and promote content by using pipelines**: In the self-service content publishing usage scenario, content is promoted (or *deployed*) through development, test, and production workspaces by using Power BI deployment pipelines. Power BI deployment pipelines can promote content to Power BI Premium workspaces manually using the user interface, or programmatically using the REST APIs. In contrast, enterprise content publishing (the focus of this usage scenario) promotes content by using *Azure Pipelines*. Azure Pipelines are an Azure DevOps service that automates testing, management, and deployment of content by using a series of customized, programmatic steps. In the enterprise content publishing usage scenario, these pipelines can also be referred to as *continuous integration and deployment* (or CI/CD) pipelines. These pipelines frequently and automatically integrate changes and streamline content publishing.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

DevOps supports a mature, systematic approach to content management and publication. It enables content creators to collaborate on solutions, and it ensures fast and reliable delivery of content to consumers. When you adhere to DevOps practices, you benefit from streamlined workflows, fewer errors, and improved experiences for content creators and content consumers.

You set up and manage DevOps practices for your Power BI solution by using Azure DevOps. In enterprise scenarios, you can automate content publication with Azure DevOps and the Power BI REST APIs in three different ways.

- **Power BI REST APIs with Power BI deployment pipelines**: You can import content to development workspaces and use deployment pipelines to deploy content through test and production workspaces. You still control deployment from Azure DevOps, and use the Power BI REST APIs to manage deployment pipelines instead of individual content items. Additionally, you use the XMLA endpoint to deploy data model metadata instead of a Power BI Desktop file (.pbix) with Azure DevOps. This metadata allows you to track object-level changes by using version control. While more robust and easier to maintain, this approach does require Premium

licensing and moderate scripting effort to set up content import and deployment with the Power BI REST APIs. Use this approach when you want to simplify content lifecycle management with deployment pipelines, and you have a Premium license. The XMLA endpoint and Power BI deployment pipelines are Premium features.

- **Power BI REST APIs**: You can also import content to development, test and production workspaces by using Azure DevOps and only the Power BI REST APIs. This approach doesn't require Premium licensing, but it does involve high scripting effort and setup, because deployment is managed outside of Power BI. Use this approach when you want to deploy Power BI content centrally from Azure DevOps, or when you don't have a Premium license. For a visual comparison between the first two approaches, see the release pipeline approaches flow diagram.
- **Power BI automation tools with Power BI deployment pipelines**: You can use the Power BI automation tools Azure DevOps extension to manage deployment pipelines instead of the Power BI REST APIs. This approach is an alternative to the first option, which uses the Power BI REST APIs with Power BI deployment pipelines. The Power BI automation tools extension is an open source tool. It helps you manage and publish content from Azure DevOps without the need to write PowerShell scripts. Use this approach when you want to manage deployment pipelines from Azure DevOps with minimal scripting effort, and you have a Premium capacity.

This article focuses on the first option, which uses the Power BI REST APIs with Power BI deployment pipelines. It describes how you can use Azure DevOps to set up DevOps practices. It also describes how you can use Azure Repos for your remote repositories and automate content testing, integration, and delivery with Azure Pipelines. Azure DevOps isn't the only way to set up enterprise content publishing, but simple integration with Power BI makes it a good choice.

> **Note**
>
> This usage scenario is one of the content management and deployment scenarios. For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

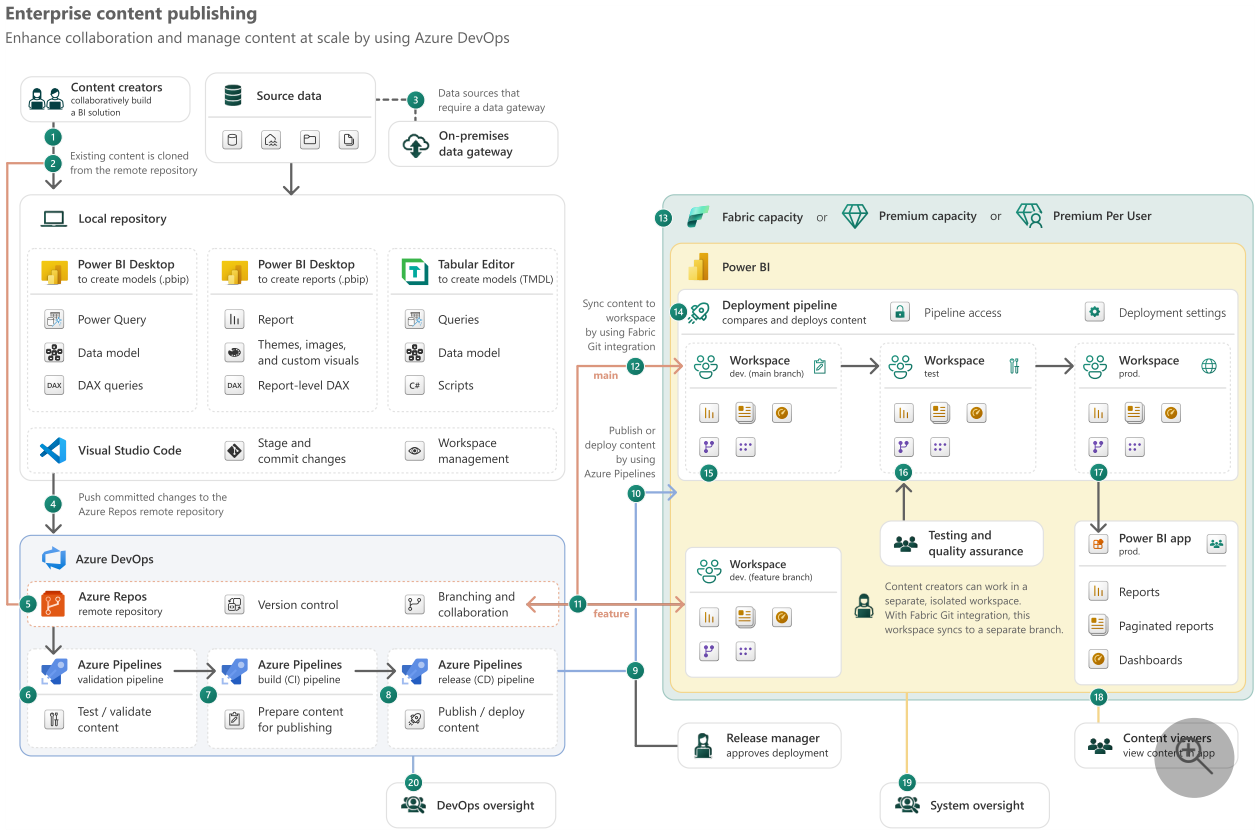
> **Tip**
>
> Microsoft Fabric provides other options for enterprise content publishing by using Fabric Git integration. Git integration lets you link a Fabric workspace to a branch in your Azure Repos remote repository. Content saved to that branch will

synchronize to the workspace automatically, as if that content was published to the workspace. Conversely, content creators can commit and push changes from the Fabric workspace to the remote repository.

Git integration can streamline collaboration and content publishing, but it requires more planning for how you'll use Fabric workspaces and what your branching strategy is. For more information about how you can set up and use Fabric Git integration, see **Get started with Git integration** or **Tutorial: End to end lifecycle management**.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support enterprise content publishing. The focus is on the use of Azure DevOps to manage and publish content programmatically at scale through development, test, and production workspaces in the Power BI service.



Tip

We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster.

Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, processes, and features.

Expand table

Item	Description
1	Content creators develop data models by using Power BI Desktop or Tabular Editor, and they develop reports by using Power BI Desktop. Content creators save their work to a local repository during development.
2	Content creators can clone a remote repository to get a local copy of that content.
3	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
4	Content creators regularly commit and push their changes to a remote repository during development by using a Git client such as Visual Studio Code. In the diagram, the remote repository is Azure Repos.
5	Other content creators use Azure Repos to track changes with version control. They collaborate by committing changes into separate branches.
6	Changes to content in the remote repository trigger Azure Pipelines. A **validation pipeline** is the first pipeline that is triggered. The validation pipeline performs automated tests to validate content before it's published.
7	Content that passes the validation pipeline triggers a subsequent **build pipeline**. The build pipeline prepares content for publishing to the Power BI service. The process up to this point is typically referred to as *continuous integration* (CI).
8	Content is published and deployed by using **release pipelines**. The release pipelines use either the Power BI REST APIs or the workspace XMLA endpoint to programmatically import content to the Power BI service. Deployment by using release pipelines is typically referred to as *continuous deployment* (CD).
9	A release manager controls deployment to test and production workspaces by using an Azure Pipelines release approval. In enterprise content publishing, a release manager typically plans and coordinates the content release to test and production environments. They coordinate and communicate with content creators, stakeholders, and users.
10	A release pipeline publishes content to the development workspace, or promotes content from development to test workspaces, or test to production workspaces.
11	Content creators working in a workspace that has **Fabric capacity** license mode can use Git integration. With Git integration, content creators can work in a private workspace during development. The content creator syncs a remote branch (typically a specific feature

Item	Description
	branch or a bug branch) from Azure Repos to their private workspace. Content changes are synchronized between the remote branch in Azure Repos and the workspace. In this scenario, content creators don't need to use Azure Pipelines to publish content. Content creators can also regularly commit and push changes from the workspace after publishing. When ready, content creators can make a pull request (PR) to merge their changes to the main branch.
12	When using Git integration, the development workspace syncs with the main branch to get the latest versions of content. This content includes all changes from pull requests that a release manager reviews, approves, and merges.
13	Workspaces are set to **Fabric capacity**, **Premium capacity**, **Premium per user**, or **Embedded** license mode, to allow using Power BI deployment pipelines and the XMLA read/write endpoint.
14	A deployment pipeline administrator sets up the Power BI deployment pipeline with three stages: development, test, and production. Each stage aligns to a separate workspace in the Power BI service. Deployment settings and access are set for the deployment pipeline.
15	The development workspace contains the latest versions of content including all approved and merged changes. Once approved, a release pipeline deploys content from the development to the test workspace.
16	Reviewers within the test workspace perform testing and quality assurance on content. Once approved, a release pipeline deploys content from the test to the production workspace. When using Git integration with deployment pipelines, the test workspace doesn't sync with any branch.
17	After the deployment pipeline completes deployment, content creators manually perform post-deployment activities. Activities can include setting up scheduled data refresh or updating a Power BI app for the production workspace. When using Git integration with deployment pipelines, the production workspace doesn't sync with any branch.
18	Content viewers access the content by using the production workspace or a Power BI app.

> **Tip**
>
> We recommend that you also review the **self-service content publishing** and **advanced data model management** usage scenarios. The enterprise content publishing usage scenario builds upon concepts that these scenarios introduce.

Key points

The following are some key points to emphasize about the enterprise content publishing scenario.

Version control

Tracking changes during the content lifecycle is important to ensure stable and consistent delivery of content to consumers. In this usage scenario, content creators and owners manage content changes in a remote repository by using *version control*. Version control is the practice of managing changes to files or code in a central repository. This practice allows for better collaboration and effective management of version history. Version control has advantages for content creators, including the ability to roll back or merge changes.

Content creators typically develop data models in Tabular Editor to support better version control. Unlike a data model that you develop in Power BI Desktop, a data model developed in Tabular Editor is saved in human-readable metadata format. This format enables data model object-level version control. You should use object-level version control when collaborating with multiple people on the same data model. For more information, see the advanced data model management usage scenario. It isn't possible to see changes that you made in a Power BI Desktop file (.pbix), such as the report definition or data model. For example, you can't track changes to a report page, such as the visuals used, their positions, and their field mappings or formatting.

Content creators store data model metadata files and .pbix files in a central remote repository, like Azure Repos. These files are curated by a *technical owner*. While a content creator develops a solution, a technical owner is responsible for managing the solution and reviewing the changes, and merging them into a single solution. Azure Repos provides sophisticated options for tracking and managing changes. This approach differs from the approach described in the self-service content publishing usage scenario, where the creator uses OneDrive storage with version tracking. Maintaining a well-curated, documented repository is essential because it's the foundation of all content and collaboration.

Here are some key considerations to help you set up a remote repository for version control.

- **Scope**: Clearly define the scope of the repository. Ideally, the scope of the repository is identical to the scope of the downstream workspaces and apps that you use to deliver content to consumers.
- **Access**: You should set up access to the repository by using a similar permissions model as you have set up for your deployment pipeline permissions and workspace roles. Content creators need access to the repository.
- **Documentation**: Add text files to the repository to document its purpose, ownership, access, and defined processes. For example, the documentation can describe how changes should be staged and committed.

- **Tools**: To commit and push changes to a remote repository, content creators need a *Git* client like Visual Studio or Visual Studio Code. Git is a distributed version control system that tracks changes in your files. To learn Git basics, see What is Git?.

> **Note**
>
> Consider using Git Large File Storage (LFS) if you plan to commit Power BI Desktop files (.pbix). Git LFS provides advanced options for managing files where changes aren't visible (*undiffable* files), like a .pbix file. For example, you can use file locking to prevent concurrent changes to a Power BI report during development. However, Git LFS has its own client and configuration.

Collaboration with Azure DevOps

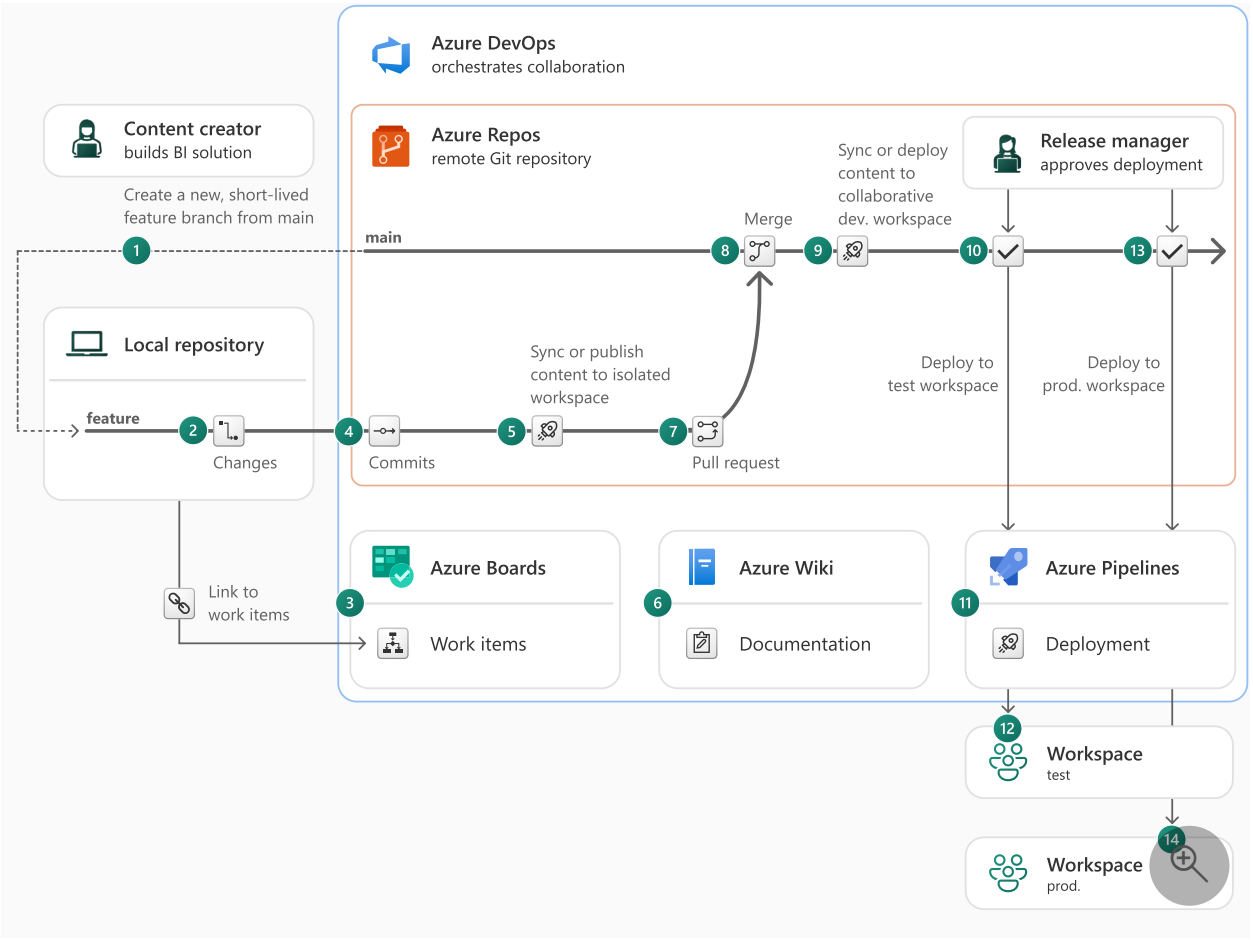
As a solution increases in scope and complexity, it might become necessary for multiple content creators and owners to work in collaboration. Content creators and owners communicate and collaborate in a central, organized hub by using Azure DevOps.

To collaborate and communicate in Azure DevOps, you use supporting services.

- **Azure Boards**: Content owners use boards to track *work items*. Work items are each assigned to a single developer on the team, and they describe issues, bugs, or features in the solution, and the corresponding stakeholders.
- **Azure Wiki**: Content creators share information with their team to understand and contribute to the solution.
- **Azure Repos**: Content creators track changes in the remote repository and merge them into a single solution.
- **Azure Pipelines**: Pipeline owners set up programmatic logic to deploy the solution, either automatically or on-demand.

Collaboration flow diagram

The following diagram depicts a high-level overview of one example of how Azure DevOps enables collaboration in the enterprise content publishing usage scenario. The focus of this diagram is on the use of Azure DevOps to create a structured and documented content publishing process.



The diagram depicts the following user actions, processes, and features.

Expand table

Item	Description
1	A content creator makes a new, short-lived branch by cloning the main branch, which contains the latest version of the content. The new branch is often referred to as the *feature* branch, as it's used to develop a specific feature or fix a specific issue.
2	The content creator commits their changes to a local repository during development.
3	The content creator links their changes to work items that are managed in Azure Boards. Works items describe specific developments, improvements, or bug fixes scoped to their branch.
4	The content creator regularly commits their changes. When ready, the content creator publishes their branch to the remote repository.
5	To test their changes, the content creator deploys their solution to an isolated workspace for their development (not shown in this diagram). The content creator can also sync the feature branch to the workspace by using Fabric Git integration.
6	Content creators and content owners document the solution and its processes in an Azure Wiki, which is available to the entire development team.

Item	Description
7	When ready, the content creator opens a pull request to merge the feature branch into the main branch.
8	A technical owner is responsible for reviewing the pull request and merging changes. When they approve the pull request, they merge the feature branch into the main branch.
9	A successful merge triggers deployment of the solution to a development workspace by using an Azure Pipeline (not shown in this diagram). When using Fabric Git integration, the main branch syncs to the development workspace.
10	The release manager performs a final review and approval of the solution. This release approval prevents the solution from being published before it's ready. In enterprise content publishing, a release manager typically plans and coordinates the content release to test and production workspaces. They coordinate and communicate with content creators, stakeholders and users.
11	When the release manager approves the release, Azure Pipelines automatically prepare the solution for deployment. Alternatively, an Azure Pipeline can also trigger a deployment pipeline to promote content between workspaces.
12	Users test and validate content in the test workspace. When using Git integration with Azure Pipelines for deployment, the test workspace doesn't sync with any branch.
13	After users accept and validate changes, the release manager performs a final review and approval of the solution to deploy to the production workspace.
14	Users view content that's published to the production workspace. When using Git integration with Azure Pipelines for deployment, the production workspace doesn't sync with any branch.

To elaborate, content creators achieve collaboration by using a *branching strategy*. A branching strategy is how content creators create, use, and merge branches to effectively make and manage content changes. Individual content creators work in isolation in their local repository. When ready, they combine their changes as a single solution in the remote repository. Content creators should scope their work to branches by linking them to work items for specific developments, improvements, or bug fixes. Each content creator creates their own branch of the remote repository for their scope of work. Work done on their local solution is committed and pushed to a version of the branch in the remote repository with a *commit message*. A commit message describes the changes made in that commit.

To merge the changes, a content creator opens a *pull request*. A pull request is a submission for peer review that can lead to the merge of the work done into a single solution. Merging can result in conflicts, which must be resolved before the branch can

be merged. Pull request reviews are important to ensure creators adhere to organizational standards and practices for development, quality, and compliance.

Collaboration recommendations

We recommend that you define a structured process for how content creators should collaborate. Ensure that you determine:

- How work is scoped and how branches are created, named, and used.
- How authors group changes and describe them with commit messages.
- Who's responsible for reviewing and approving pull requests.
- How merge conflicts are resolved.
- How changes made in different branches are merged together into a single branch.
- How content is tested and who performs testing before content is deployed.
- How and when changes are deployed to development, test, and production workspaces.
- How and when deployed changes or versions of the solution should be rolled back.

> **Important**
>
> The value provided by DevOps is directly proportional to the adherence to the processes that define its use.
>
> A successful collaboration depends on a well-defined process. It's important to clearly describe and document the end-to-end development workflow. Ensure that the selected strategies and processes align with existing practices in your team, and if not, how you'll manage change. Further, ensure that the processes are clear and communicated to all team members and stakeholders. Make sure that team members and stakeholders who are new to the processes are trained in how to adopt them, and that they appreciate the value of successful DevOps adoption.

Power BI REST APIs

You develop programmatic logic to import and deploy content from Azure DevOps by using the Power BI REST APIs. You import Power BI files (.pbix) to a workspace by using an import operation. You use a pipeline operation to deploy some content or all content to test or production workspaces by using Power BI deployment pipelines. The programmatic logic is defined in the Azure Pipelines.

We recommend that you use a service principal to call Power BI REST APIs in your pipelines. A service principal is intended for unattended, automated activities, and it doesn't rely on user credentials. However, some items and activities aren't supported by the Power BI REST APIs or when using a service principal, like dataflows.

When you use a service principal, be sure to carefully manage permissions. Your goal should be to follow the principle of least privilege. You should set sufficient permissions for the service principal without over-provisioning permissions. Use Azure Key Vault or another service that securely stores the service principal secrets and credentials.

> **⊗ Caution**
>
> If you have a data model that's saved as a human-readable metadata format, it can't be published by using the Power BI REST APIs. Instead, you must publish it by using the **XMLA endpoint**. You can publish metadata files by using third-party tools like the **Tabular Editor command-line interface** (CLI). You can also publish metadata files programmatically by using your own custom .NET development. Developing a custom solution requires more effort, since you must use the Microsoft Tabular Object Model (TOM) extension of the Analysis Management Object (AMO) client libraries.

Azure Pipelines

Azure Pipelines programmatically automate testing, management, and deployment of content. When a pipeline is run, steps in the pipeline execute automatically. Pipeline owners can customize its triggers, steps, and functionality to meet deployment needs. As such, the number and types of pipelines vary depending on the solution requirements. For example, an Azure Pipeline could run automated tests or modify data model parameters before a deployment.

There are three types of Azure Pipelines that you can set up to test, manage, and deploy your Power BI solution:

- Validation pipelines.
- Build pipelines.
- Release pipelines.

> **ⓘ Note**
>
> It's not necessary to have all three of these pipelines in your publishing solution. Depending on your workflow and needs, you might set up one or more of the

variants of the pipelines described in this article to automate content publication. This ability to customize the pipelines is an advantage of Azure Pipelines over the built-in Power BI deployment pipelines. For example, you don't have to have a validation pipeline; you can use only use build and release pipelines.

Validation pipelines

Validation pipelines perform basic quality checks of data models before they're published to a development workspace. Typically, changes in a branch of the remote repository trigger the pipeline to validate those changes with automated testing.

Examples of automated testing include scanning the data model for best practice rule violations by using Best Practice Analyzer (BPA), or by running DAX queries against a published semantic model. The results of these tests are then stored in the remote repository for documentation and auditing purposes. Data models that fail validation shouldn't be published. Instead, the pipeline should notify content creators of the issues.

Build pipelines

Build pipelines prepare data models for publication to the Power BI service. These pipelines combine serialized model metadata into a single file that's later published by a release pipeline (described in the release pipelines diagram). A build pipeline can also make other changes to the metadata, like modifying parameter values. The build pipelines produce deployment artifacts that consist of data model metadata (for data models) and Power BI Desktop files (.pbix) that are ready for publication to the Power BI service.

Release pipelines

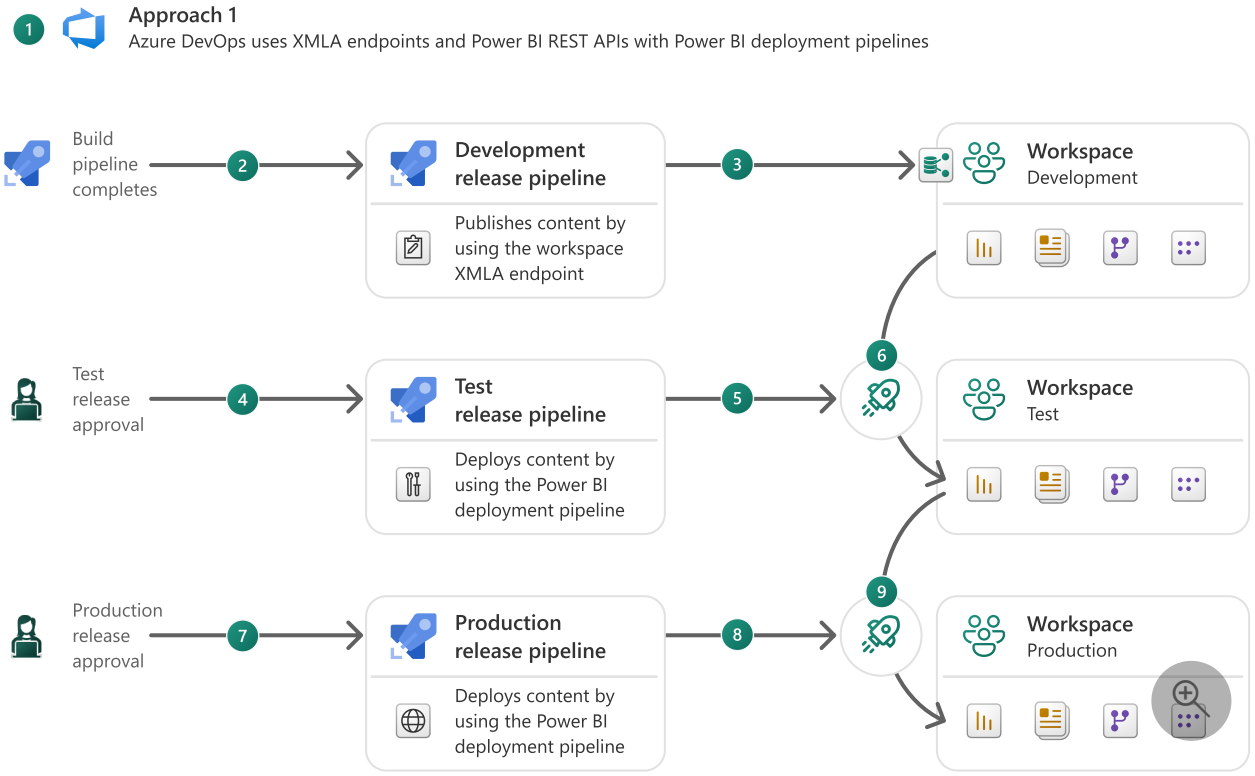
Release pipelines publish or deploy content. A publishing solution typically includes several release pipelines, depending on the target environment.

- **Development release pipeline**: This first pipeline is triggered automatically. It publishes content to a development workspace after the build and validation pipelines succeed.
- **Test and production release pipelines**: These pipelines aren't triggered automatically. Instead, they're triggered on-demand or when approved. Test and production release pipelines deploy content to a test or production workspace, respectively, after *release approval*. Release approvals ensure that content isn't automatically deployed to a test or production stage before it's ready. These

approvals are provided by release managers, who are responsible for planning and coordinating content release to test and production environments.

There are two different approaches to publish content with test and release pipelines. Either they promote content by using a Power BI deployment pipeline, or they publish content to the Power BI service from Azure DevOps.

The following diagram depicts the first approach. In this approach, release pipelines orchestrate content deployment to test and production workspaces by using Power BI deployment pipelines. Content is promoted through development, test, and production workspaces in Power BI. While this approach is more robust and simpler to maintain, it requires Premium licensing.



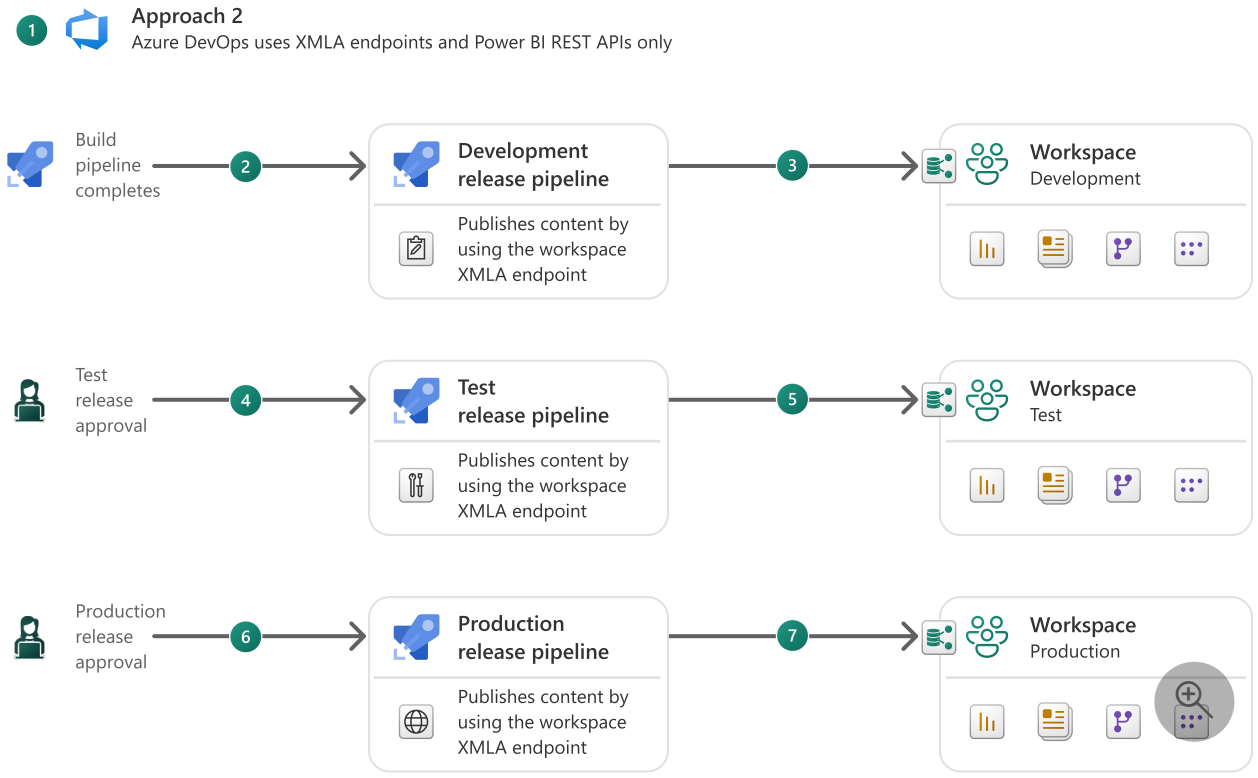
The diagram depicts the following user actions, processes, and features of the first approach.

Expand table

Item	Description
1	In the first approach, the release pipelines publish content by using the XMLA endpoint and the Power BI REST APIs with Power BI deployment pipelines. Content is published and then promoted through development, test, and production workspaces. Power BI deployment pipelines and the XMLA read/write endpoint are Premium features.
2	Either a successful branch merge or completion of an upstream pipeline triggers the build pipeline. The build pipeline then prepares content for publishing, and triggers the development release pipeline.

Item	Description
3	The development release pipeline publishes content to the development workspace by using the XMLA endpoint (for data model metadata) or the Power BI REST APIs (for Power BI Desktop files, which can contain data models and reports). The development pipeline uses the Tabular Editor command-line interface (CLI) to deploy data model metadata by using the XMLA endpoint.
4	A release approval or on-demand trigger activates the test release pipeline.
5	The test release pipeline deploys content by using the Power BI REST API deploy operations, which run the Power BI deployment pipeline.
6	The Power BI deployment pipeline promotes content from the development workspace to the test workspace. After deployment, the release pipeline performs post-deployment activities by using the Power BI REST APIs (not shown in the diagram).
7	A release approval or on-demand trigger activates the production release pipeline.
8	The production release pipeline deploys content by using the Power BI REST API deploy operations, which run the Power BI deployment pipeline.
9	The Power BI deployment pipeline promotes content from the test workspace to the production workspace. After deployment, the release pipeline performs post-deployment activities by using the Power BI REST APIs (not shown in the diagram).

The following diagram depicts the second approach. This approach doesn't use deployment pipelines. Instead, it uses release pipelines to publish content to test and production workspaces from Azure DevOps. Notably, this second approach doesn't require Premium licensing when you publish only Power BI Desktop files with the Power BI REST APIs. It does involve more setup effort and complexity, because you must manage deployment outside of Power BI. Development teams that already use DevOps for data solutions outside of Power BI might be more familiar with this approach. Development teams that use this approach can consolidate deployment of data solutions in Azure DevOps.



The diagram depicts the following user actions, processes, and features in the second approach.

Expand table

Item	Description
1	In the second approach, the release pipelines publish content by using the XMLA endpoint and the Power BI REST APIs only. Content is published to development, test, and production workspaces.
2	Either a successful branch merge or completion of an upstream pipeline triggers the build pipeline. The build pipeline then prepares content for publishing, and triggers the development release pipeline.
3	The development release pipeline publishes content to the development workspace by using the XMLA endpoint (for data model metadata) or the Power BI REST APIs (for Power BI Desktop files, which can contain data models and reports). The development pipeline uses the Tabular Editor command-line interface (CLI) to deploy data model metadata by using the XMLA endpoint.
4	A release approval or on-demand trigger activates the test release pipeline.
5	The development release pipeline publishes content to the test workspace by using the XMLA endpoint (for data model metadata) or Power BI REST APIs (for Power BI Desktop files, which can contain data models and reports). The development pipeline uses the Tabular Editor command-line interface (CLI) to deploy data model metadata by using the XMLA endpoint. After deployment, the release pipeline performs post-deployment activities by using the Power BI REST APIs (not shown in the diagram).

Item	Description
6	A release approval or on-demand trigger activates the production release pipeline.
7	The development release pipeline publishes content to the production workspace by using the XMLA endpoint (for data model metadata) or Power BI REST APIs (for Power BI Desktop files, which can contain data models and reports). The development pipeline uses the Tabular Editor command-line interface (CLI) to deploy data model metadata by using the XMLA endpoint. After deployment, the release pipeline performs post-deployment activities by using the Power BI REST APIs (not shown in the diagram).

Release pipelines should manage post-deployment activities. These activities could include setting semantic model credentials or updating the Power BI app for test and production workspaces. We recommend that you set up notifications to inform relevant people about deployment activities.

> **Tip**
>
> Using a repository for version control allows content creators to create a rollback process. The rollback process can reverse the last deployment by restoring the previous version. Consider creating a separate set of Azure Pipelines that you can trigger to roll back production changes. Think carefully about what processes and approvals are required to initiate a rollback. Ensure that these processes are documented.

Power BI deployment pipelines

A Power BI deployment pipeline consists of three stages: development, test, and production. You assign a single Power BI workspace to each stage in the deployment pipeline. When a deployment occurs, the deployment pipeline promotes Power BI items from one workspace to another.

An Azure Pipelines release pipeline uses the Power BI REST APIs to deploy content by using a Power BI deployment pipeline. Access to both the workspace and the deployment pipeline is required for the users conducting a deployment. We recommend that you plan deployment pipeline access so that pipeline users can view deployment history and compare content.

> **Tip**
>
> When you **separate data workspaces from reporting workspaces**, consider using Azure Pipelines to orchestrate content publishing with multiple Power BI

deployment pipelines. Semantic model are deployed first, and then they're refreshed. Lastly, reports are deployed. This approach helps you simplify deployment.

Premium licensing

Power BI deployment pipelines and the XMLA read/write endpoint are Premium features. These features are available with Power BI Premium capacity and Power BI Premium Per User (PPU).

PPU is a cost-effective way to manage enterprise content publishing for development and test workspaces, which typically have few users. This approach has the added advantage of isolating development and test workloads from production workloads.

> **Note**
>
> You can still set up enterprise content publishing without a Premium license, as described by the second approach in the release pipeline section. In the second approach, you use Azure Pipelines to manage deployment of Power BI Desktop files to development, test, and production workspaces. However, you can't deploy model metadata by using the XMLA endpoint because it's not possible to publish a metadata format semantic model with the Power BI REST APIs. Also, it's not possible to promote content through environments with deployment pipelines without a Premium license.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The two purposes of a gateway are to refresh imported data, and view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

When working with multiple environments, it's common to set up development, test, and production connections to different source systems. In this case, use data source rules and parameter rules to manage values that differ between environments. You can use Azure Pipelines to manage gateways by using the gateway operations of the Power BI REST APIs.

> **Note**

A centralized data gateway in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records events that occur in the Power BI service. Power BI administrators can use the activity log to audit deployment activities.

You can use the Power BI metadata scanning APIs to create a tenant inventory. The API results are helpful to verify which items have been deployed to each workspace, to check lineage, and to validate security settings.

There's also an audit log within Azure DevOps, which is outside of the Power BI service. Azure DevOps administrators can use the audit log to review activities in remote repositories and pipelines.

Related content

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Advanced data model management

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This usage scenario focuses on *advanced data model management*, which is when a Power BI content creator relies on a third-party tool to develop, manage, or optimize data models. Some third-party tools are *external tools*, which Power BI Desktop supports directly. You can also manage a published a data model (semantic model) by communicating directly with the XMLA endpoint in the Power BI service.

Data models are hosted in either the Power BI service, Azure Analysis Services (AAS), or SQL Server Analysis Services (SSAS). This usage scenario focuses on using the XMLA endpoint in the Power BI service.

> **Tip**
>
> Many people refer to third-party tools as *external tools*. However, there are distinctions in how different tools can be used. Connecting to a local data model in Power BI Desktop is the most literal interpretation of the term *external tool*. This advanced data model management usage scenario focuses on connecting to a remote data model (a semantic model hosted in the Power BI service) by using the XMLA endpoint. More details on the different ways to use third-party tools are described later in this article.

You can achieve connectivity to a data model by using the *XML for Analysis (XMLA)* protocol. The XMLA protocol is an industry standard protocol that's supported by more than 25 vendors, including Microsoft. All tools, including third-party tools, that are compliant with the XMLA protocol use Microsoft client libraries to read and/or write data to a data model. Connectivity is achieved with an XMLA endpoint, which is an API exposed by a data model that broadens the development and management capabilities available to semantic model creators.

> **Note**
>
> This advanced data model management usage scenario is one of the content management and deployment scenarios. For a complete list of the self-service usage scenarios, see Power BI usage scenarios.
>
> For brevity, some aspects described in the content collaboration and delivery scenarios topic aren't covered in this article. For complete coverage, read those articles first.

Scenario diagram

The focus of this advanced data model management usage scenario is on using Tabular Editor to manage the data model. You can publish a data model to the Power BI service by using the XMLA endpoint, which is available with Power BI Premium.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see Important update coming to Power BI Premium licensing and Power BI Premium FAQ.

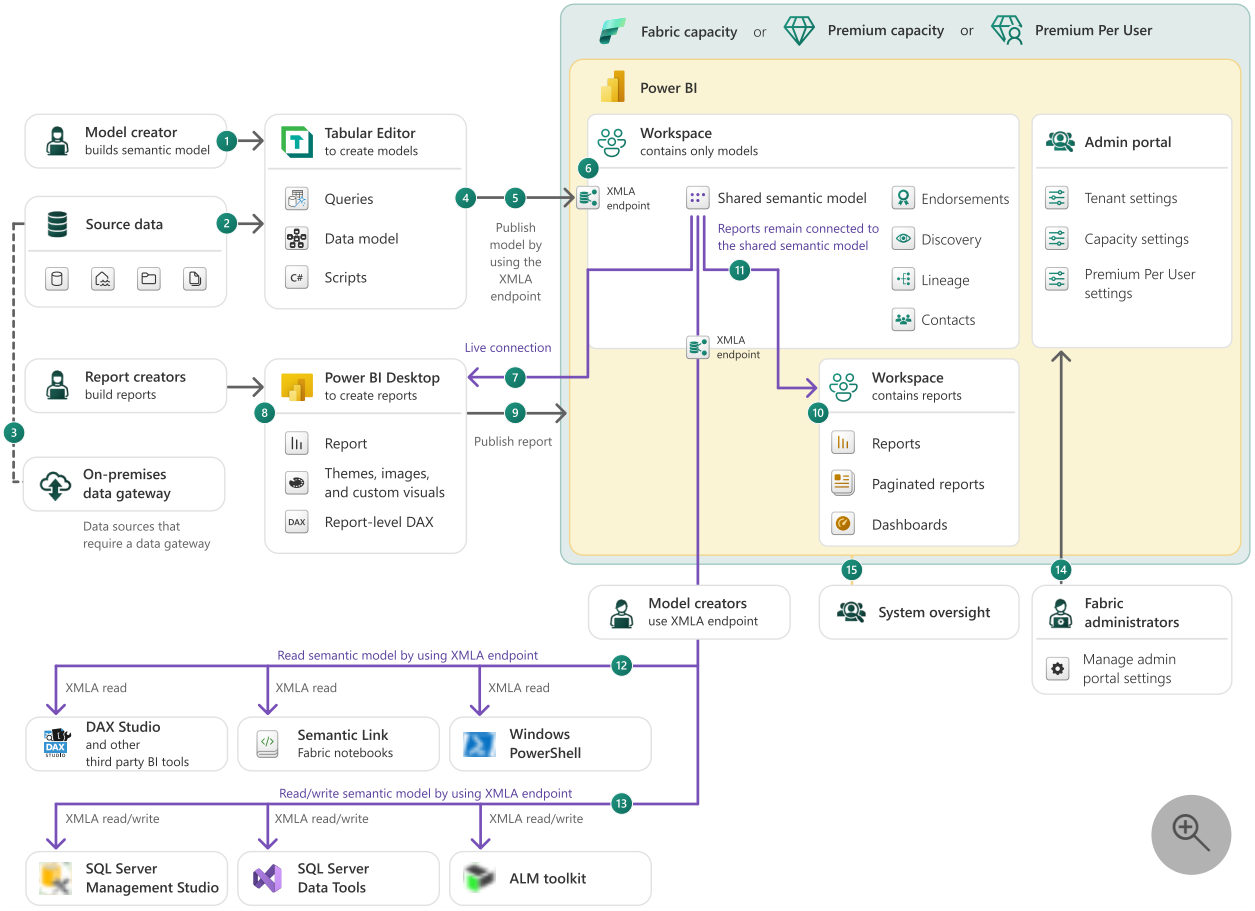
> **Tip**
>
> We recommend that you review the self-service content publishing usage scenario if you're not familiar with it. The advanced data model management scenario builds upon that scenario.

> **Note**
>
> Sometimes the terms *semantic model* and *data model* are used interchangeably. Generally, from a Power BI service perspective, it's referred to as *semantic model*. From a development perspective, it's referred to as a *data model* (or *model* for

short). In this article, both terms have the same meaning. Similarly, a semantic model creator and a data modeler have the same meaning.

The following diagram depicts a high-level overview of the most common user actions and tools that can help you develop, manage, or optimize data models.

Advanced data model management
Empower creators with advanced modeling and publishing capabilities



Tip

We encourage you to download the scenario diagram if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Model creators develop data models by using Tabular Editor. It's common that the initial design work (like Power Query work) is done in Power BI Desktop before switching to

Item	Description
	Tabular Editor (not depicted in the scenario diagram).
2	The data model connects to data from one or more data sources.
3	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
4	Data model development is done in Tabular Editor. Editing of Power Query (M) scripts is supported. Model creators may use C# scripts to accelerate development.
5	When ready, semantic model creators publish the data model from Tabular Editor to the Power BI service by using the XMLA endpoint of the target workspace.
6	The data model is published to a workspace dedicated to storing and securing shared semantic models. Access to the workspace by using the XMLA endpoint is only possible when the workspace license mode is set to **Fabric capacity**, **Premium capacity**, **Premium per user**, or **Embedded**.
7	Report creators create reports by using a live connection to the shared semantic model.
8	Report creators develop reports in Power BI Desktop. Other than purposefully separating reports from semantic models, content creators follow the typical report creation process.
9	When ready, report creators publish their Power BI Desktop file (.pbix) or Power BI project file (.pbip) to the Power BI service.
10	Reports are published to a workspace dedicated to storing and securing reports and dashboards.
11	Published reports remain connected to the shared semantic model that's stored in a different workspace. Any changes made to the shared semantic model affect all dependent reports.
12	Third-party tools can use the XMLA endpoint to query the shared semantic model. Other XMLA-compliant tools—such as DAX Studio, Semantic Link from Fabric notebooks, or Windows PowerShell—can be used to query or update the shared semantic model. Power BI Desktop, Excel, and Report Builder can also connect by using the XMLA endpoint (not depicted in the scenario diagram).
13	Other Microsoft and third-party tools can use the XMLA endpoint to manage the semantic model and provide application lifecycle management. To learn more, see XMLA endpoint-based client tools.
14	Fabric administrators manage the tenant setting to enable the use of the XMLA endpoint. The administrator must enable the XMLA endpoint for Fabric capacities, Premium capacities, and Premium Per User settings.
15	Fabric administrators oversee and monitor activity in the Fabric portal.

Key points

The following are some key points to emphasize about the advanced data model management scenario.

Third-party applications and tools

Enterprise BI teams commonly use client tools, such as Tabular Editor (depicted in the scenario diagram and described in the next topic), to help them manage centralized semantic models. However, any semantic model creator that wants to work with advanced modeling capabilities can take advantage of the methods described in this usage scenario.

There are several ways to use third-party applications:

- **Connect to a remote data model by using the XMLA endpoint**: Some third-party tools can connect directly to a remote data model in the Power BI service (or Analysis Services). Once connected to the XMLA endpoint, all Tabular Object Model (TOM) operations are supported. This approach is the primary focus of this usage scenario.
- **Connect to a local data model in Power BI Desktop**: Some third-party tools can connect to a local data model that's open in Power BI Desktop (not depicted in the scenario diagram). However, there are some limitations, and not all external tool functionality is officially supported.
- **Connect to a template file in Power BI Desktop**: Some third-party tools distribute their functionality in a lightweight way by using a Power BI Desktop template file (.pbit) (not depicted in the scenario diagram).

Tabular Editor

Tabular Editor is depicted in the scenario diagram. It's a third-party tool that's achieved widespread adoption by the Power BI community. Some advantages of managing tabular data models with Tabular Editor include:

- **Setting up data model capabilities not supported in Power BI Desktop**: Tabular Editor provides an interface to set up object-level security (OLS), calculation groups, perspectives, translations, and partitions.
- **Support for concurrent model development**: Microsoft data model development tools, such as Visual Studio with Analysis Services projects, store the entire data model definition in a *Model.bim* file. This single file can make it difficult for a team of developers to work together on a single data model. Tabular Editor has a feature

called *Folder serialization*. Folder serialization deconstructs the Model.bim file into separate object-specific files within an organized folder structure. Different data modelers can then work on different files with less risk of overwriting each other's efforts.

- **Integration with source control**: Folder serialization allows source control system to easily detect data model changes, making source merges and conflict resolution easier to do.
- **Improved data model quality and design**: Tabular Editor integrates with Best Practices Analyzer (BPA). BPA helps data modelers with a set of customizable rules that can improve the quality, consistency, and performance of data models. You can download a set of best practice rules (provided by Microsoft) from GitHub.
- **Increased productivity when developing data models**: The Tabular Editor interface makes it well-suited for performing batch edits, debugging, and viewing data model dependencies. Tabular Editor is different from Power BI Desktop in that it works in *disconnected mode*. You can make data model changes in disconnected mode and commit them as a batch of edits. Working this way allows for faster development and validation, especially for experienced data modelers. It's also possible to create C# scripts and save them as macros. These scripts can help you to improve the efficiency of managing and synchronizing multiple data models.

XMLA endpoint

The XMLA endpoint uses the XMLA protocol to expose all features of a tabular data model, including some data modeling operations that aren't supported by Power BI Desktop. You can use the TOM API to make programmatic changes to a data model.

The XMLA endpoint also provides connectivity. You can only connect to a semantic model when the workspace that has its license mode set to **Premium per user**, **Premium per capacity**, or **Embedded**. Once a connection is made, an XMLA-compliant tool can operate on the data model in two ways:

- **Write data and metadata**: Read/write use of the XMLA endpoint allows for:
  - Data modeling capabilities that aren't supported by Power BI Desktop, like object-level security (OLS), calculation groups, perspectives, translations, and partition management.
  - More complex deployments. For example, a partial deployment or a metadata-only deployment that publishes only a single new measure.
  - Asynchronous semantic model refresh. For example, refreshing a single table or partition.
- **Read data and metadata**: Read-only use of the XMLA endpoint allows for:

- Monitoring, debugging, and tracing semantic models and queries.
- Allowing third-party data reporting tools to visualize data sourced from a shared semantic model. This technique is a great way to extend the benefits and investments in managed self-service BI.

> ⚠ **Warning**
>
> Once you modify or publish a semantic model by using the XMLA endpoint, you can no longer download it from the Power BI service as a Power BI Desktop file.

XMLA settings per capacity

Each Power BI Premium capacity and Power BI Embedded capacity has a setting to control whether the XMLA endpoint is read-only, read/write, or off. This setting is also available for all Premium Per User workspaces in the Power BI tenant. Read/write XMLA access must be enabled for each capacity that contains semantic models that you want to manage with a tool other than Power BI Desktop.

> **Tip**
>
> The XMLA endpoint setting (read/write, read-only, or off) applies to all workspaces and semantic models assigned to a particular capacity. You can set up multiple capacities to decentralize and/or customize how content is managed for each capacity.

XMLA tenant setting

In addition to the XMLA endpoint settings, a Power BI administrator must use the tenant settings to allow XMLA endpoints and Analyze in Excel with on-premises semantic models. When enabled, you can allow all users, or specific security groups, to use XMLA endpoint functionality.

> **Note**
>
> All standard security and data protection features still apply to specify which users can view and/or edit content.

Third-party tools

Power BI Desktop can handle the end-to-end needs for most self-service content creators. However, third-party tools offer other enterprise features and functionality. For this reason, third-party tools, such as Tabular Editor, have become prevalent in the Power BI community, especially for advanced content creators, developers, and IT professionals.

> **Tip**
>
> This **blog post** describes how third-party tools allow the Power BI product team to reevaluate their development priorities, increase the reach of the Power BI platform, and satisfy more advanced and diverse requests from the user community.

> **Note**
>
> Some third-party tools require a paid license, such as Tabular Editor 3. Other community tools are free and open source (such as Tabular Editor 2, DAX Studio, and ALM Toolkit). We recommend that you carefully evaluate the features of each tool, cost, and support model so you can adequately support your community of content creators.

Data model management

The primary focus of this usage scenario is on the content creator who uses Tabular Editor to manage a data model. For infrequent advanced data model management requirements, like occasional partition management, you might choose to use a tool such as SQL Server Management Studio (SSMS). It's also possible for a .NET developer to create and manage Power BI semantic models by using the TOM API.

> **Tip**
>
> When using the XMLA endpoint for data model management, we recommend that you enable the **large semantic model storage format** setting. When enabled, the large semantic model storage format can improve XMLA write operation performance.

Separation of data model and reports

For this usage scenario to be successful, you should separate reports from the data model. This approach results in managing separate Power BI Desktop files as described in the managed self-service BI usage scenario. Even if the same person is responsible for all development, the separation of semantic models and reports is important because Tabular Editor doesn't have an awareness of report content.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The On-premises data gateway becomes relevant once a data model is published to the Power BI service. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model (not depicted in the scenario diagram).

> **Note**
>
> A centralized data gateway in *standard mode* is strongly recommended over gateways in ***personal mode***. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).
>
> For more information, see On-premises data gateway (standard mode).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand activities that connect through XMLA endpoints.

Related content

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Power BI usage scenarios: Self-service real-time analytics

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This usage scenario focuses on how a business analyst can produce *real-time* Power BI reports. What's meant by real-time is that the data is always current, and report consumers aren't required to interact with visualizations. Data visualizations should refresh automatically to always show current data.

Real-time reports allow organizations to monitor and make confident decisions based on up-to-date data.

> **Note**
>
> In this article, the term *real-time* actually means *near real-time*. Near real-time means that there's always a degree of delay (known as latency), due to data processing and network transmission time.

> **Important**
>
> Microsoft has announced a retirement date for real-time streaming in Power BI. for more information, see Announcing the retirement of real-time streaming in Power BI.

To develop self-service real-time analytics, the business analyst will first need to create (or connect to) a DirectQuery semantic model. They can then build a report and set up its automatic page refresh settings. Once set up, Power BI automatically refreshes report pages to show current data.

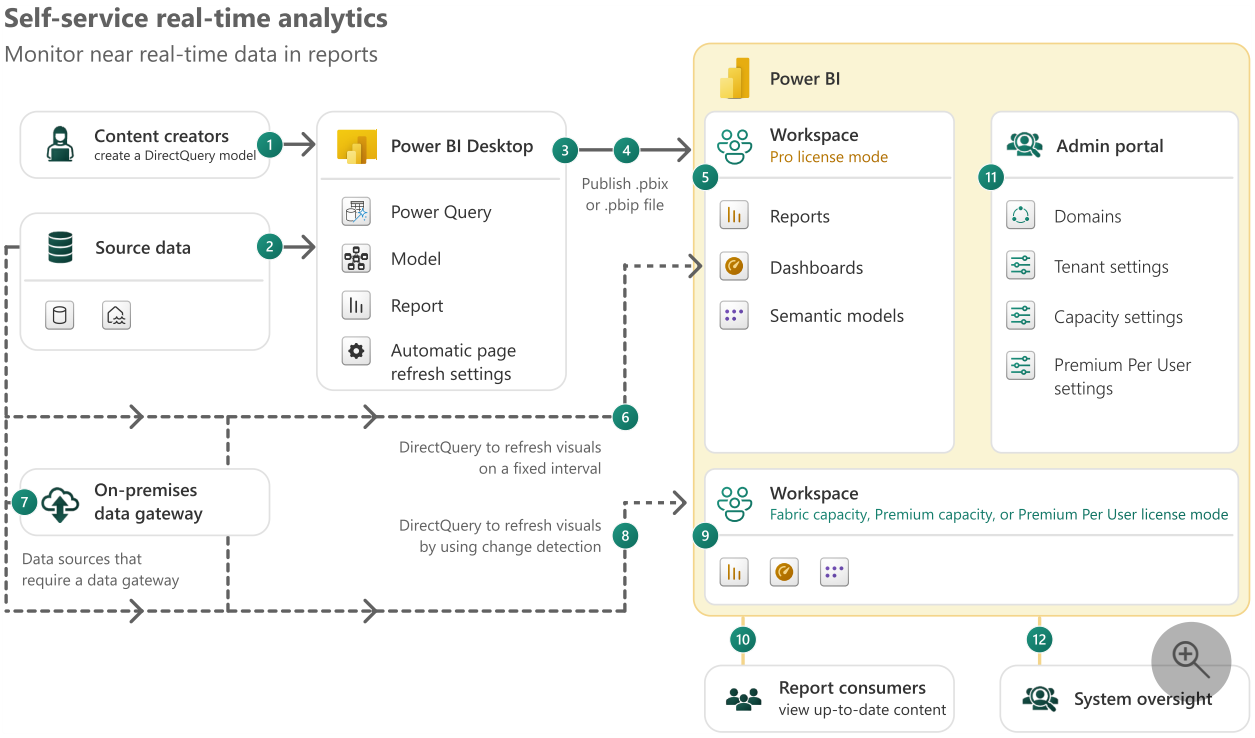
> **Tip**

You can also achieve real-time analytics in Power BI by using *push datasets*. However, this topic is out of scope for this self-service real-time usage scenario because it targets developers. Push datasets usually involve developing a programmatic solution.

For a complete understanding of Power BI real-time analytics, work through the Monitor data in real-time with Power BI learning path.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support self-service real-time analytics. The primary objective is on creating a DirectQuery model and building Power BI reports that use automatic page refresh.



Tip

We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The above diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	Content creators use Power BI Desktop to create a DirectQuery model.
2	Power BI Desktop sends native queries to the underlying data source in order to retrieve current data.
3	Content creators build a report that will display near real-time updates by enabling and setting up automatic page refresh.
4	When ready, the content creators publish their Power BI Desktop file (.pbix) or Power BI project file (.pbip) to a workspace in the Power BI service or Fabric portal.
5	Once published, the workspace contains a new report and DirectQuery semantic model. When the workspace is a personal or Pro workspace, the minimum automatic page refresh interval is 30 minutes (even when the report creator sets a lower interval).
6	When report consumers open a report page that has automatic page refresh enabled, data visualizations refresh automatically to show current data.
7	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
8	Each visual on an automatic page refresh page queries the semantic model to retrieve current data from the underlying data source.
9	When an automatic page refresh report is stored in a workspace that uses **Fabric capacity**, **Premium capacity**, or **Premium Per User** license mode, Power BI can automatically refresh on intervals of one minute or more. It's also possible to use the change detection refresh type so Power BI can avoid unnecessary refreshes. When the change detection refresh type is set, at each refresh interval, Power BI sends change detection queries to determine whether data has changed since the last automatic refresh. When Power BI detects change, it refreshes all visuals on the page.
10	Report consumers view up-to-date content from a workspace or Power BI app.
11	Capacity administrators can enable or disable the automatic page refresh feature. When the feature is disabled, automatic page refresh won't work for any reports stored in workspaces assigned to the capacity. Capacity administrators can also set a minimum refresh interval and a minimum execution interval. These minimum intervals will override any report page setting that uses a lower interval.
12	Fabric administrators oversee and monitor activity in the Fabric portal.

Important

At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.

For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Key points

The following are some key points to emphasize about the self-service real-time analytics scenario.

Supported data sources

The automatic page refresh feature doesn't work for reports connected to import models, where all tables use import storage mode. The feature only works when the Power BI report connects to a semantic model that:

- Includes DirectQuery storage mode tables.
- Uses incremental refresh to get the latest data in real-time with DirectQuery. This capability is described later in this topic.
- Is a live connection to a tabular model in Azure Analysis Services (AAS) or SQL Server Analysis Services (SSAS).
- Is a push dataset.

A DirectQuery model is an alternative to an import model. Models developed in DirectQuery mode don't import data. Instead, they consist only of metadata defining the model structure. When the model is queried, native queries are used to retrieve data from the underlying data source.

From a self-service perspective, the business analyst can add DirectQuery storage tables to their model in Power BI Desktop, providing the data source supports this storage mode. Typically, relational databases are supported by DirectQuery. For a full listing of data sources that support DirectQuery, see Data sources supported by DirectQuery.

A business analyst can also enhance an import model by setting up incremental refresh. By enabling the *Get the latest data in real-time with DirectQuery* option (only supported by Premium workspaces), Power BI Desktop adds a DirectQuery partition to ensure the latest data is retrieved. For more information, see Incremental refresh and real-time data for semantic models.

The business analyst can also create a live connection to an existing tabular model that includes DirectQuery storage mode tables.

Involve data source owners

Before publishing an automatic page refresh report, it's a good idea to first discuss the real-time requirements with the data source owners. That's because automatic page refresh can place a significant workload on the data source.

Consider a single report page that's set to refresh every five minutes and that includes two visuals. When the report page is open, Power BI will send at least 24 queries per hour (12 refreshes multiplied by two visuals) to the underlying data source. Now consider that 10 report consumers open the same report page at the same time. In this case, Power BI will send 240 queries per hour.

It's important to discuss the real-time requirements, including the number of visuals on the report page and the desired refresh interval. When the use case is justified, the data source owner can take proactive steps by scaling up the data source resources. They can also optimize the data source by adding useful indexes and materialized views. For more information, see DirectQuery model guidance in Power BI Desktop.

Refresh type

The automatic page refresh feature supports two refresh types.

- **Fixed interval**: Updates all page visuals based on a fixed interval, which can be from every one second to multiple days.
- **Change detection**: Updates all page visuals providing that source data has changed since the last automatic refresh. It avoids unnecessary refreshes, which can help to reduce resource consumption for the Power BI service and the underlying data source. Power BI only supports this refresh type for Premium workspaces and for data models hosted by Power BI. Remote data models, which are hosted in AAS or SSAS, aren't supported.

To set up change detection, you must create a special type of measure called a *change detection measure*. For example, a change detection measure might query for the maximum sales order number. Power BI uses the change detection measure to query the data source. Each time, Power BI stores the query result so it can compare it with the next result (according to the refresh interval you set). When the results differ, Power BI refreshes the page.

A model can only have one change detection measure, and there can only be a maximum of 10 change detection measures per tenant.

For more information, see Refresh types.

Capacity administration

When a workspace is attached to a Premium capacity, capacity administrators can enable or disable the automatic page refresh feature for the capacity. When the feature is disabled, automatic page refresh won't work for any report stored in any of the attached workspaces.

Capacity administrators can also set a minimum refresh interval (default five minutes) and a minimum execution interval (default five minutes). The execution interval determines the frequency of change detection queries. When a report page interval is less that the capacity minimum interval, Power BI will use the capacity minimum interval.

> **Note**
>
> Minimum intervals don't apply to reports open in Power BI Desktop.

When there are performance issues related to automatic page refresh, a capacity administrator can:

- Scale up the capacity to a larger Premium SKU.
- Raise the minimum intervals.

For more information, see Page refresh intervals.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The gateway supports the DirectQuery operations (visual queries and change detection queries).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption.

By using the Fabric Capacity Metrics app that's available to administrators, you can visualize how much of the capacity is being used by low-priority queries. Low-priority queries consist of automatic page refresh queries and model refresh queries. Change detection queries aren't low priority.

Related content

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Embed for your organization

Article • 12/30/2024

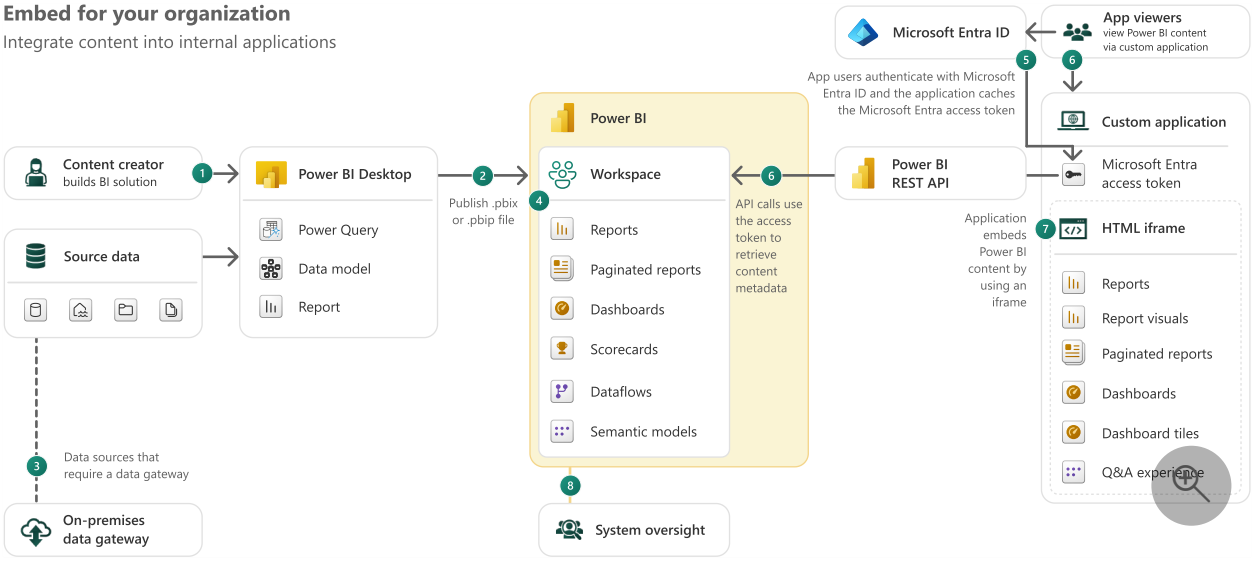
> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This usage scenario focuses on how a developer can programmatically embed Power BI content in a custom application *for your organization*. (The developer isn't necessarily responsible for creating the Power BI content.) The **Embed for your organization** scenario applies when the application audience comprises users who have permission and appropriate licenses to access Power BI content in the organization. These users must have organizational accounts (including guest accounts), which authenticate with Microsoft Entra ID.

> **Note**
>
> In this scenario, Power BI is software-as-a-service (SaaS). The embedding scenario is sometimes referred to as *User owns data*.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support embedding for your organization.



Tip

We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The above diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	The Power BI content creator develops a BI solution by using Power BI Desktop.
2	When ready, the content creator publishes the Power BI Desktop file (.pbix) or Power BI project file (.pbip) to the Power BI service.
3	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
4	A Power BI workspace contains Power BI items ready for embedding. For non-personal workspaces, users of the custom application have permission to view (or create or modify) Power BI content because they belong to a workspace role or they have direction permissions.
5	The custom application prompts the app user to authenticate with Microsoft Entra ID. When authentication succeeds, the custom application caches a Microsoft Entra access token.
6	The custom application uses the Microsoft Entra access token to make Power BI REST API calls on behalf of the app user. Specifically, the application uses the access token to

Item	Description
	retrieve metadata about workspace items. Metadata includes properties required to embed content in the custom application.
7	The custom application embeds a specific Power BI item in an `iframe` HTML element. The application can support the creation and editing of Power BI reports, providing the user has permission to do so.
8	Power BI administrators oversee and monitor activity in the Power BI service.

Key points

The following are some key points to emphasize about programmatically embed Power BI content in a custom application for your organization.

Use cases

There are several reasons why you might embed Power BI content for your organization.

- **Internal business intelligence portal**: You might want to create an internal business intelligence (BI) portal as a replacement for the Power BI service. That way, you can create a custom application that integrates content from Power BI and other BI tools.
- **Internal app**: You might want to develop an intranet app that shows data visualizations. For example, an intranet site for a manufacturing department could show real-time visuals that provide up-to-date information about the production line.
- **Customized logging**: You might want to log custom events to record Power BI content access and use, beyond what the activity log records.

> **Tip**
>
> If you're looking to create a BI portal styled for your organization, you might be able to achieve that by simply adding **custom branding to the Power BI service**.

No-code embedding

Developing a programmatic solution requires skill, time, and effort. Consider that there are embedding techniques known as *no-code embedding* that non-developers can use to embed content in a simple internal portal or website.

- Use the Power BI report web part to embed Power BI reports in SharePoint Online.
- Use a secure embed code (or HTML) that's generated by Power BI to embed Power BI reports in internal web portals.
- Embed Power BI reports or dashboards in Power Pages.
- Embed reports in a Microsoft Teams channel or chat.

These techniques require that report consumers belong to the organization, be authenticated, and have permission to access the reports. Power BI ensures that all permissions and data security are enforced when consumers view the reports. Sometimes, users might be challenged to authenticate by signing in to Power BI.

Embeddable content

When embedding for your organization, you can embed the following Power BI content types:

- Power BI reports
- Specific Power BI report visuals
- Paginated reports
- Q&A experience
- Dashboards
- Specific dashboard tiles

There's no limitation on where the content resides. The content could reside in a personal workspace or a regular workspace. What matters is that the app user has permission to view (or create or edit) the content. For example, it's possible to embed content from the app user's personal workspace.

Any content the user can see in the Power BI service can be embedded in a custom application. If the user has permission to create or edit content, it's possible for a custom app to support that functionality (for Power BI reports only).

Authentication

The authentication flow is *interactive authentication* with Microsoft Entra ID. Interactive authentication means that the app user will be challenged to authenticate. When authenticated, Microsoft Entra ID returns an access token. It's the responsibility of the custom application to cache the access token so that it can be used to make Power BI REST API calls and to embed content inside an `iframe` HTML element. Those calls can retrieve metadata about Power BI content on behalf of the app user, including the properties required to embed it in the custom application.

Licensing

There's no specific licensing requirement to embed for your organization. What matters is that the app user has permission and an appropriate Power BI license to view (or create or edit) the content. It's even possible to embed content from a personal workspace when the app user only has a Fabric (free) license.

Power BI client APIs

The Power BI client APIs allow a developer to achieve tight integration between the custom application and the Power BI content. They develop the application by writing custom logic with JavaScript or TypeScript that runs in the browser.

The application can set up and automate operations, and it can respond to user-initiated actions. Additionally, you can integrate Power BI capabilities, including navigation, filters and slicers, menu operations, layout, and bookmarks.

> **Tip**
>
> The *Power BI Embedded Analytics Playground* is a website that helps you learn, explore, and experiment with Power BI embedded analytics. It includes a developer sandbox for hands-on experiences that use the client APIs with sample Power BI content or your own content. Code snippets and showcases are available for you to explore, too.
>
> For more information, see **What is the Power BI embedded analytics playground?**

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model.

> **Note**
>
> A centralized **data gateway** in *standard mode* is strongly recommended over gateways in **personal mode**. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption. Logged events will describe the consumption method as *Embedding for your organization*. There's presently no way to determine whether content was viewed in a no-code embedding experience in a custom application.

Related content

To learn more about Power BI embedded analytics, work through the Embed Power BI analytics learning path.

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: Embed for your customers

Article • 12/30/2024

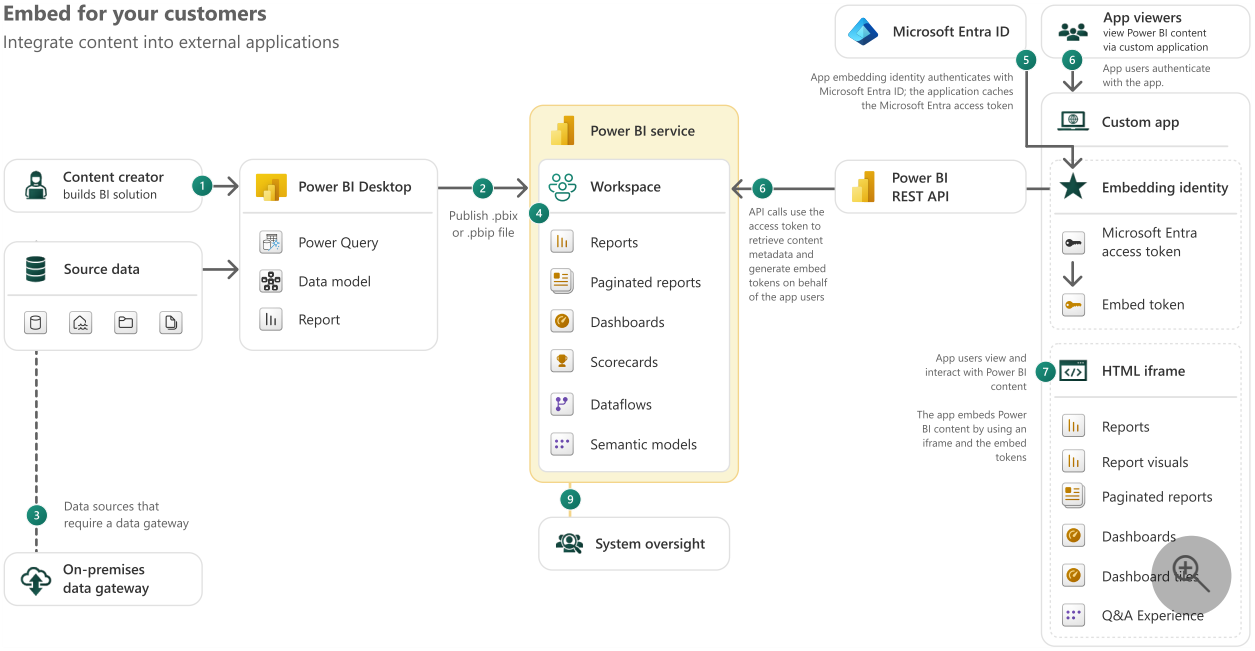
> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

This usage scenario focuses on how a developer can programmatically embed Power BI content in a custom application *for your customers*. (The developer isn't necessarily responsible for creating the Power BI content.) The **Embed for your customers** scenario applies when the application audience comprises users who don't have permission or appropriate licenses to access Power BI content in your organization. The custom application requires an embedding identity that has permission and an appropriate license to access Power BI content. The custom application could be a multitenancy application.

> **Note**
>
> In this scenario, Power BI is platform-as-a-service (PaaS). The embedding scenario is sometimes referred to as *App owns data*.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components that support embedding for your customers.



> **Tip**
>
> We encourage you to download the scenario diagram if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The above diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	The Power BI content creator develops a BI solution by using Power BI Desktop.
2	When ready, the content creator publishes the Power BI Desktop file (.pbix) or Power BI project file (.pbip) to the Power BI service.
3	Some data sources may require an On-premises data gateway or VNet gateway for data refresh, like those that reside within a private organizational network.
4	A Power BI workspace contains Power BI items ready for embedding. An embedding identity, either a service principal or master user account, must belong to either the workspace Admin or Member role. In a multi-tenancy solution, the separation of tenants is achieved by creating one workspace for each tenant. This design pattern is known as *workspace separation*.
5	The custom application prompts the app user to authenticate by using any authentication method (not necessarily Microsoft Entra ID.

Item	Description
6	When authentication succeeds, the custom application uses the embedding identity to acquire and cache a Microsoft Entra access token.
7	The custom application uses the Microsoft Entra access token to make Power BI REST API calls on behalf of the embedding identity. Specifically, the application uses the access token to retrieve metadata about workspace items. Metadata includes properties required to embed content in the custom application. It also uses the access token to generate and cache embed tokens, which represent facts about Power BI content and how the application can access it.
8	The custom application embeds a specific Power BI item in an `iframe` HTML element. The application can support the creation and editing of Power BI reports, providing the embedding identity has permission to do so.
9	Power BI administrators oversee and monitor activity in the Power BI service.

Key points

The following are some key points to emphasize about programmatically embed Power BI content in a custom application for your customers.

Use case

Often, embedding for your customers is done by Independent Software Vendors (ISVs). ISVs recognize a need to embed analytics in their apps. It allows users to have direct access to in-context insights, helping them make decisions based on facts instead of opinions. Instead of developing visualizations, it's usually faster and less expensive to embed Power BI content.

ISVs can develop a *multitenancy application*, where each of their customers is a tenant. A multitenancy application that embeds Power BI analytics will use the *Embed for your customers* scenario because the application users include external users. Multitenancy applications are described in more detail later in this article.

Embeddable content

When embedding for your customers, you can embed the following Power BI content types:

- Power BI reports
- Specific Power BI report visuals

- Paginated reports
- Q&A experience
- Dashboards
- Specific dashboard tiles

There's no limitation on where the content resides, except the content can't reside in a personal workspace. What matters is that the embedding identity has permission to view (or create or edit) the content.

Authentication

The authentication flow is *non-interactive authentication* with Microsoft Entra ID (also known as *silent authentication*). Non-interactive authentication means that the app user isn't required to have a Power BI account, and even when they do, it isn't used. So a dedicated Microsoft Entra identity, known as the embedding identity, authenticates with Microsoft Entra ID. An embedding identity can be a service principal or a master user account (described later).

The authentication flow attempts to acquire a Microsoft Entra token in a way in which the authentication service can't prompt the user for additional information. Once the app user authenticates with the app (the app can use any authentication method), the app uses the embedding identity to acquire a Microsoft Entra token by using a non-interactive authentication flow.

Once the app acquires a Microsoft Entra token, it caches it and then uses it to generate an *embed token*. An embed token represents facts about Power BI content and how to access them. The app uses the embed token to embed content inside an `iframe` HTML element.

Service principal

An app can use a service principal to acquire a Microsoft Entra token. A Microsoft Entra service principal is a security identity used by apps. It defines the access policy and permissions for the app in the Microsoft Entra tenant, enabling core features such as authentication of the app during sign in, and authorization during resource access. A service principal can authenticate by using an app secret or certificate. A service principal can only use Power BI REST APIs, when the *Allow service principals to use Power BI APIs* tenant setting is enabled, and the service principal belongs to an allowed group.

Tip

We recommend using a service principal for production apps. It provides the highest security and for this reason it's the approach recommended by Microsoft Entra ID. Also, it supports better automation and scale and there's less management overhead. However, it requires Power BI admin rights to set up and manage.

Master user account

An app can use a *master user account* to acquire an AD token. A master user account is a regular Microsoft Entra user. In Power BI, the account must belong to the workspace Admin or Member role to embed workspace content. It must also have either a Power BI Pro or Power BI Premium Per User (PPU) license.

> **Note**
>
> It's not possible to use a master user account to embed paginated reports.

For more information about embedding identities, see Set up permissions to embed Power BI content.

Licensing

When embedding Power BI content for your customers, you need to ensure that content resides in a workspace that has one of the following license modes:

- **Premium capacity**: This license mode is available with Power BI Premium.
- **Embedded**: This license mode is available with Power BI Embedded.
- **Fabric capacity**: This license mode is available with Microsoft Fabric.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

Each license mode option requires the purchase of a billable product that is a capacity-based license. A capacity-based license allows you to create *reserved capacities*.

Capacities represent the computational resources that are required to process workloads, such as report rendering and data refresh. Reserved capacities are isolated from other customers' workloads, so they offer scale that can deliver dependable and consistent performance.

> **Note**
>
> It's not possible to use the *Embed for your customers* scenario in production environments with the Fabric (free), Power BI Pro, or Power BI PPU licenses.

For more information about products and licensing, see Select the appropriate Power BI embedded analytics product.

Power BI client APIs

The Power BI client APIs allow a developer to achieve tight integration between the custom application and the Power BI content. They develop the application by writing custom logic with JavaScript or TypeScript that runs in the browser.

The application can set up and automate operations, and it can respond to user-initiated actions. Additionally, you can integrate Power BI capabilities, including navigation, filters and slicers, menu operations, layout, and bookmarks.

> **Tip**
>
> The *Power BI Embedded Analytics Playground* is a website that helps you learn, explore, and experiment with Power BI embedded analytics. It includes a developer sandbox for hands-on experiences that use the client APIs with sample Power BI content or your own content. Code snippets and showcases are available for you to explore, too.
>
> For more information, see **What is the Power BI embedded analytics playground?**

Enforce data permissions

When the app users should only have access to view a subset of data, you need to develop a solution that restricts access to Power BI semantic model data. The reason might be because some users aren't permitted to view specific data, such as sales results

of other sales regions. Achieving this requirement commonly involves setting up row-level security (RLS), which involves defining roles and rules that filter model data.

When you use the *For your customers scenario*, the app must set the effective identity of the embed token to restrict access to data. This effective identity determines how Power BI will connect to the model and how it will enforce RLS roles. How you set up the effective identity depends on the type of Power BI semantic model.

For more information about RLS roles for embedded content, see Enforce data permissions for Power BI embedded analytics.

Multitenancy applications

Multiple organizations can use a multitenancy app, where each organization is a tenant. A multitenancy app that embeds Power BI analytics can use the *Embed for your customers* scenario because the app users include external users. When designing a multitenancy app, you can choose from two different tenancy models.

The recommended approach is to use the *workspace separation* model. You can achieve this approach by creating one Power BI workspace for each tenant. Each workspace contains Power BI artifacts that are specific to that tenant, and the semantic models connect to a separate database for each tenant.

> **Tip**
>
> For more information about the workspace separation model, see **Automate workspace separation**. For more information about scalable multitenancy apps, see **Service principal profiles for multitenancy apps in Power BI Embedded**.

Alternatively, the single multi-customer database model is available. When you use this model, your solution will achieve separation with a single workspace that includes a set of Power BI items that are shared across all tenants. RLS roles, which are defined in the semantic models, will help filter the data more securely to ensure that organizations only view their own data.

No-code embedding

Developing a programmatic solution requires skill, time, and effort. Consider that there's one embedding technique known as *no-code embedding* that non-developers can use to embed Power BI reports or dashboards in Power Pages.

Gateway setup

Typically, a data gateway is required when accessing data sources that reside within the private organizational network or a virtual network. The two purposes of a gateway are to refresh imported data, or view a report that queries a live connection or DirectQuery semantic model.

> **Note**
>
> A centralized data gateway in *standard mode* is strongly recommended over gateways in personal mode. In standard mode, the data gateway supports live connection and DirectQuery operations (in addition to scheduled data refresh operations).

System oversight

The activity log records user activities that occur in the Power BI service. Power BI administrators can use the activity log data that's collected to perform auditing to help them understand usage patterns and adoption.

Related content

To learn more about Power BI embedded analytics, work through the Embed Power BI analytics learning path.

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenarios: On-premises reporting

Article • 12/30/2024

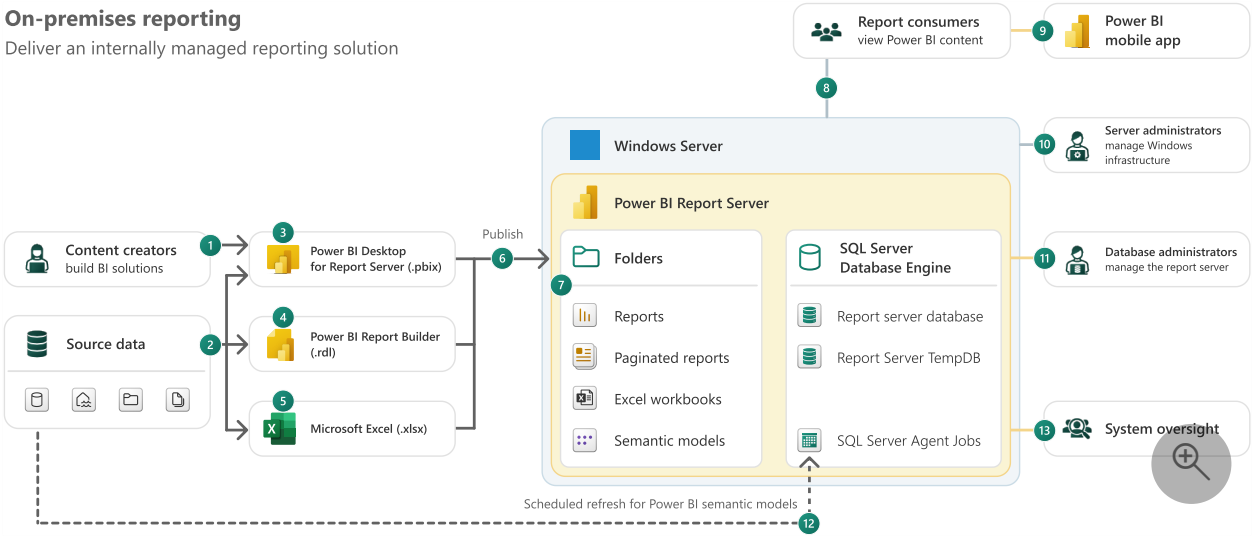
> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

The *on-premises reporting* scenario is one of several *hybrid and custom scenarios* for deploying Power BI solutions without using the Power BI service.

This scenario involves using Power BI Report Server, which is an on-premises portal for publishing, sharing, and consuming business intelligence content within the organizational network. It's useful when the organization needs an alternative to the cloud-based Power BI service for deploying some (or all) BI content. For example, a fully customer-managed platform could be necessary for regulatory, legal, or intellectual property reasons.

Scenario diagram

The following diagram depicts a high-level overview of the most common user actions and Power BI components to support on-premises reporting. The focus is on using Power BI Report Server, which runs on a Windows server within the organizational network.



> **Tip**
>
> We encourage you to **download the scenario diagram** if you'd like to embed it in your presentation, documentation, or blog post—or print it out as a wall poster. Because it's a Scalable Vector Graphics (SVG) image, you can scale it up or down without any loss of quality.

The scenario diagram depicts the following user actions, tools, and features:

Expand table

Item	Description
1	A Power BI content creator builds a BI solution.
2	Power BI Desktop for Report Server connects to data from one or more data sources. Queries and data mashups, which combine multiple sources, are developed in the Power Query Editor.
3	Data model development and report creation are done in Power BI Desktop for Report Server. It generates a specific type of Power BI Desktop file (.pbix) that can be published to Power BI Report Server.
4	The report creator can also build paginated reports using Power BI Report Builder. This tool generates a Report Definition Language file (.rdl) that can be published to Power BI Report Server.
5	The report creator can also develop reports using Excel. The Excel workbook file (.xlsx) can be published to Power BI Report Server.
6	When ready, the content creator publishes their file to Power BI Report Server.
7	Content is published to a folder in Power BI Report Server.
8	Report consumers view reports published to Power BI Report Server.
9	Report consumers can also view reports using Power BI mobile apps.
10	Server administrators manage the Windows server infrastructure.
11	Database administrators manage Power BI Report Server, including the report server databases, and SQL Server Agent.
12	SQL Server Agent jobs periodically refresh import semantic models.

Item	Description
13	Administrators oversee and monitor activity in Power BI Report Server.

Key points

The following are some key points to emphasize about the on-premises reporting scenario.

Report creator experience

Content creators use a specific tool named Power BI Desktop for Report Server. This version of Power BI Desktop is updated three times per year and is compatible with the Power BI Report Server release cycle.

> **Note**
>
> For report creators who create content for both the Power BI service and Power BI Report Server, the two versions of Power BI Desktop can be installed side by side.

Report consumer experience

The report consumer experience for Power BI Report Server is very different from the Power BI service. The Power BI Report Server is a web portal for viewing, storing, and managing content. Content files (.pbix, .rdl, or .xlsx) are published to a folder hierarchy. For more information, see Manage content in the web portal.

Power BI Report Server

Power BI Report Server is a distinct product from SQL Server Reporting Services (SSRS). It's licensed and installed separately. Power BI Report Server is considered a superset of SSRS because it comprises additional capabilities beyond SSRS.

> **Important**
>
> Although Power BI Report Server and the Power BI service are supported by the same engineering team at Microsoft, there are substantial functionality differences between the two products. Power BI Report Server is a basic reporting portal for on-premises reporting. For this reason there are many **feature differences** between

it and the Power BI service. The feature set of Power BI Report Server is intentionally simple, and parity should not be expected. Before installing Power BI Report Server, verify that critical features you intend to use are supported.

Report server databases

SQL Server hosts the Report Server databases. Most commonly, a SQL Server Database Engine instance is installed on a Windows server in an on-premises data center. It can also be installed on a virtual machine in Azure (hosted cloud) or hosted by Azure SQL Managed Instance (not depicted in the scenario diagram). The database infrastructure is managed by a database administrator.

Mobile access

Additional configurations must be done to enable remote mobile access to Power BI Report Server. For more information, see Configure Power BI mobile app access to Report Server remotely.

Licensing Power BI Report Server

There are two ways to license Power BI Report Server: Power BI Premium and SQL Server Enterprise Edition with Software Assurance.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

With the purchase of Power BI Premium capacity, Power BI Report Server can be installed on an on-premises server, provided it has the same number of cores as the capacity node's v-cores. This way, it's possible to adopt a hybrid approach supporting publication of content to the Power BI service (cloud) and to Power BI Report Server (on-premises or hosted cloud in Azure).

> **Note**
>
> When licensing Power BI Report Server as part of the Premium capacity feature set, it's only available with the P SKUs. The other capacity-based SKUs (EM and A SKUs) do not offer this benefit, nor does Power BI Premium Per User (PPU).

Related content

For other useful scenarios to help you with Power BI implementation decisions, see the Power BI usage scenarios article.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI usage scenario diagrams

Article • 12/30/2024

> **Note**
>
> This article forms part of the *Power BI implementation planning* series of articles. This series focuses primarily on the Power BI experience within Microsoft Fabric. For an introduction to the series, see Power BI implementation planning.

We encourage you to download the Power BI usage scenario diagrams if you'd like to embed them in your presentations, documentation, or blog posts—or print them out as wall posters. Because they're Scalable Vector Graphics (SVG) images, you can scale them up or down without any loss of quality.

Advanced data model management

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Advanced data preparation

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Customizable managed self-service BI

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Departmental BI

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Embed for your customers

- Usage scenario diagram

- Usage scenario diagram (without legend numbers)

Embed for your organization

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Enterprise BI

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Enterprise content publishing

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Managed self-service BI

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

On-premises reporting

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Personal BI

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Prototyping and sharing

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Self-service content publishing

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Self-service data preparation

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Self-service real-time analytics

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Team BI

- Usage scenario diagram
- Usage scenario diagram (without legend numbers)

Related content

For more information, see Power BI usage scenarios.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Microsoft's BI transformation

Article • 12/30/2024

> **Tip**
>
> This article focuses on Microsoft's experience establishing a **Center of Excellence**. When setting up your own Center of Excellence, we recommend that you also review the information covered in the **Fabric adoption roadmap**.

This article targets IT professionals and IT managers. You'll learn about our BI strategy and vision, which enables us to continuously leverage our data as an asset. You'll also learn how we successfully drive a data culture of business decision making with Power BI.

Some background first: Today, the explosion of data is impacting consumers and businesses at breakneck speeds. Succeeding in this data-intensive environment requires analysts and executives who can distill enormous amount of data into succinct insights. The revolutions in Microsoft's BI tools have changed the way that Microsoft itself explores their data and gets to the right insights needed to drive impact in the company.

So, how can your organization, too, revolutionize the way it works with data? Let's help you understand by sharing the story of our BI transformation journey.

Microsoft journey

Several years ago at Microsoft, our organizational culture encouraged individuals to pursue full ownership of data and insights. It also experienced strong cultural resistance to doing things in a standardized way. So, the organizational culture led to reporting and analytic challenges. Specifically, it led to:

- Inconsistent data definitions, hierarchies, metrics, and Key Performance Indicators (KPIs). For example, each country or region had their own way of reporting on new revenue. There was no consistency, yet much confusion.
- Analysts spending 75% of time collecting and compiling data.
- 78% of reports being created in "offline environment".
- Over 350 centralized finance tools and systems.
- Approximately $30 million annual spend on "shadow applications".

These challenges prompted us to think about how we could do things better. Finance and other internal teams received executive support to transform the business review process, which led to building a unified BI platform as our single source of truth. (We'll discuss more about our BI platform later in this article.) Ultimately, these innovations led to business reviews being transformed from dense tabular views into simpler, more insightful visuals focused on key business themes.

How did we achieve this successful outcome? Delivering centralized BI managed by IT and extending it with self-service BI (SSBI) led to success. We describe it in two creative ways: *discipline at the core* and *flexibility at the edge*.

Discipline at the core

Discipline at the core means that IT retains control by curating a single master data source. Delivering standardized corporate BI and defining consistent taxonomies and hierarchies of KPIs is part of that discipline. Importantly, data permissions are enforced centrally to ensure our people can only read the data they need.

First, we understood that our BI transformation wasn't a technology problem. To achieve success we learned to first define success, and then translate it into key metrics. It cannot be understated how important it was for us to achieve consistency of definition across our data.

Our transformation didn't happen all at once. We prioritized the delivery of the subsidiary scorecard consisting of about 30 KPIs. Then, over several years, we gradually expanded the number and depth of subject areas, and built out more complex KPI hierarchies. Today, it allows us to roll up lower-level KPIs at customer level to higher ones at company level. Our total KPI count now exceeds 2000, and each is a key measure of success and is aligned to corporate objectives. Now across the entire company, corporate reports and SSBI solutions present KPIs that are well-defined, consistent, and secure.

Flexibility at the edge

At the edge of the core, our analysts in the Finance, Sales, and Marketing teams became more flexible and agile. They now benefit from the ability to analyze data more quickly. More formally, this scenario is described as *managed self-service BI (SSBI)*. We now understand that managed SSBI is about *mutual benefit* for IT and analysts. Importantly, we experienced optimizations by driving standardization, knowledge, and the reuse of our data and BI solutions. And, as a company, we derived more value synergistically as we found the right balance between centralized BI and managed SSBI.

Our solution

Starlight is the name we give to our internal data unification and analytics platform, which supports finance, sales, marketing, and engineering. Its mission is to deliver a robust, shared, and scalable data platform. The platform was built entirely by Finance, and continues in operation today using the latest Microsoft products.

The **KPI Lake** isn't an Azure Data Lake. Rather, it's a Starlight-powered tabular BI semantic model hosted in Azure IaaS using Microsoft SQL Server Analysis Services. The BI semantic model delivers data sourced from over 100 internal sources, and defines numerous hierarchies and KPIs. Its mission is to enable business performance reporting and analysis teams across Finance, Marketing, and Sales. It does so to obtain timely, accurate, and well performing insights through unified BI semantic models from relevant sources.

When first deployed, it was an exciting time because the tabular BI semantic model resulted in immediate and measurable benefits. The first version centralized C+E Finance and Marketing BI platforms. Then, over the past six years, it's been expanded to consolidate additional business insight solutions. Today, it continues to evolve, powering our global and commercial business reviews as well as standard reporting and SSBI. Its adoption has spiked 5X since its release—well beyond our initial expectations.

Here's a summary of key benefits:

- It powers our subsidiary scorecard, worldwide business reviews, and finance, marketing, sales reports and analytics.
- It supports self-service analytics, enabling analysts to discover insights hidden in data.
- It drives reporting and analytics for incentive compensation, marketing and operations analysis, sales performance metrics, senior leadership reviews, and the annual planning process.
- It delivers automated and dynamic reporting and analytics from a *single source of truth*.

The **KPI Lake** is a great success story. It's often presented to our customers to showcase an example of how to effectively use our latest technologies. Not surprisingly, it's highly resonant with many of them.

How it works

The Starlight platform manages the flow of data from acquisition, to processing, and then all the way to publication:

1. Robust and agile data integration takes place on a scheduled basis, consolidating data from over 100 disparate raw sources. Source data systems include relational databases, Azure Data Lake Storage, and Azure Synapse databases. Subject areas include finance, marketing, sales, and engineering.
2. Once staged, the data is conformed and enriched using master data and business logic. It's then loaded to data warehouse tables. The tabular BI semantic model is then refreshed.
3. Analysts across the company use Excel and Power BI to deliver insights and analytics from the tabular BI semantic model. And, it enables business owners to champion metric definitions for their own business. When necessary, scaling is achieved using Azure IaaS with load balancing.

Deliver success

Humorously, everybody wants one version of the truth... as long as it's theirs. But for some organizations it's their reality. They have multiple versions of the truth as a result of individuals pursuing full ownership of data and insights. For these organizations, this unmanaged approach isn't likely to be a pathway to business success.

It's why we believe you need a *Center of Excellence (COE)*. A COE is a central team that's responsible for defining company-wide metrics and definitions, and much more. It's also a business function that organizes people, processes, and technology components into a comprehensive set of business competencies and capabilities.

We see much evidence to support that a comprehensive and robust COE is critical to delivering value and maximizing business success. It can include change initiatives, standard processes, roles, guidelines, best practices, support, training, and much more.

We invite you to read the articles in this COE series to learn more. Let's help you discover how your organization can embrace change to deliver success.

Related content

For more information about this article, check out the following resources:

- Establish a Center of Excellence
- Fabric adoption roadmap: Center of Excellence
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

In the next article in this series, learn how a COE helped us at Microsoft to create a standardized analytics and data platform to unlock insights from our data.

Professional services

Certified Power BI partners are available to help your organization succeed when setting up a COE. They can provide you with cost-effective training or an audit of your data. To find a Power BI partner, visit the Microsoft Power BI partners portal.

You can also engage with experienced consulting partners. They can help you assess, evaluate, or implement Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

How Microsoft established a Center of Excellence

Article • 12/30/2024

> **Tip**
>
> This article focuses on Microsoft's experience establishing a **Center of Excellence**. When setting up your own Center of Excellence, we recommend that you also review the information covered in the **Fabric adoption roadmap**.

This article targets IT professionals and IT managers. You'll learn how to set up a BI and analytics Center of Excellence (COE) in your organization, and how Microsoft has set up theirs.

For some, there's a misconception that a COE is just a help desk—this thinking, however, is far from reality.

Generally, a BI and analytics COE is a team of professionals that's responsible for establishing and maintaining a BI platform. It's also responsible for creating a single source of truth, and defining a set of consistent company-wide metrics to unlock and accelerate insights. Yet, a COE is a broad term. As such, it can be implemented and managed in different ways, and its structure and scope can vary from organization to organization. At its core, it's always about a robust platform delivering the right data and insight capabilities to the right people at the right time. Ideally, it also promotes evangelizing, training, and support. At Microsoft, it's described as *discipline at the core*, and it's delivered as our BI platform and single source of truth.

In larger organizations, you could find multiple COEs with the core COE *extended* by satellite COEs—often at department level. This way, a satellite COE is a group of experts familiar with taxonomies and definitions, who know how to transform core data into what makes sense *for their department*. Departmental analysts are granted permissions to core data, and they trust it for use in their own reports. They build solutions that rely upon carefully prepared core dimensions, facts, and business logic. At times, they might also extend it with smaller, department-specific semantic models and business logic. Importantly, satellite COEs aren't ever disconnected nor do they act in isolation. At Microsoft, satellite COEs promote *flexibility at the edge*.

For this extended scenario to succeed, departments must *pay to play*. In other words, departments must financially invest in the core COE. This way, there isn't concern that they're "not getting their fair share" or that their requirements are ever de-prioritized.

To support this scenario, the core COE must scale to meet funded departmental needs. Once several semantic models have been onboarded economies of scale set in. At Microsoft, it quickly became evident that working centrally is more economic and brings about faster results. When each new subject area was onboarded, we experienced even greater economies of scale that allowed for leveraging and contributing across the entire platform, reinforcing our underlying data culture.

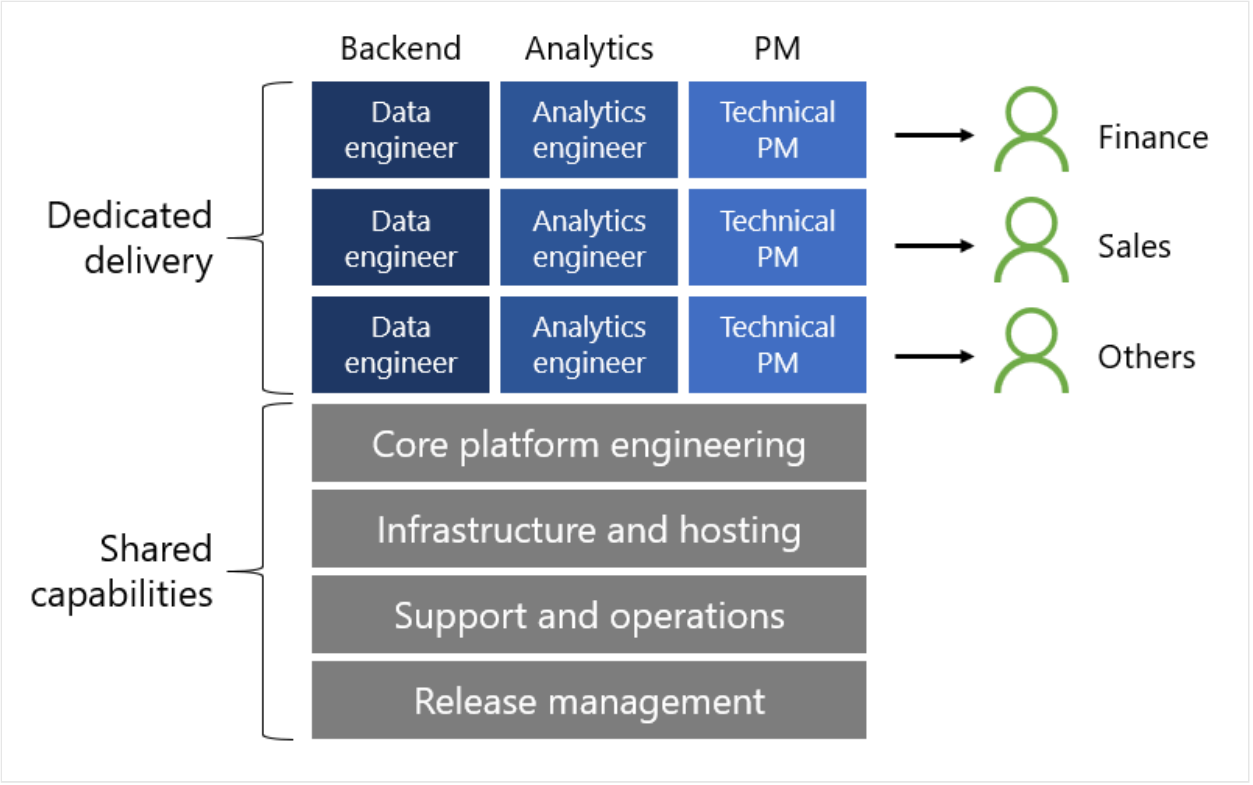
Consider an example: Our BI platform delivers core dimensions, facts, and business logic for Finance, Sales, and Marketing. It also defines hundreds of Key Performance Indicators (KPIs). Now, an analyst in the Power Platform business needs to prepare a leadership dashboard. Some of the KPIs, like revenue and pipelines, come directly from the BI platform. Others, however, are based on more granular needs of the business. One such need is for a KPI on user adoption of Power BI-specific feature: dataflows. So, the analyst produces a Power BI composite model to integrate core BI platform data with departmental data. They then add business logic to define their departmental KPIs. Finally, they author their leadership dashboard based on the new model, which leverages the company-wide COE resources amplified with local knowledge and data.

Importantly, a division of responsibility between the core and satellite COEs allows departmental analysts to focus on breaking new ground, rather than managing a data platform. At times, there can even be a mutually beneficial relationship between the satellite COEs and the core COE. For example, a satellite COE may define new metrics that—having proved beneficial to their department—end up as core metrics beneficial to the entire company, available from—and supported by—the core COE.

BI platform

In your organization, the COE might be recognized by a different name, like the BI team or group. The name matters less than what it actually does. If you don't have a formalized team, we recommend you cultivate a team that brings together your core BI experts to establish your BI platform.

At Microsoft, the COE is known as the BI Platform. It has many stakeholder groups representing different divisions within the company like Finance, Sales, and Marketing. It's organized to run shared capabilities and dedicated deliveries.



Shared capabilities

Shared capabilities are required to establish and operate the BI platform. They support all stakeholder groups that fund the platform. They comprise the following teams:

- **Core platform engineering**: We designed the BI platform with an engineering mindset. It's really a set of frameworks that support data ingestion, processing to enrich the data, and delivery of that data in BI semantic models for analyst consumption. Engineers are responsible for the technical design and implementation of the core BI platform capabilities. For example, they design and implement the data pipelines.
- **Infrastructure and hosting**: IT engineers are responsible for provisioning and managing all Azure services.
- **Support and operations**: This team keeps the platform running. Support looks after user needs like data permissions. Operations keep the platform running, ensuring that Service Level Agreements (SLAs) are met, and communicating delays or failures.
- **Release management**: Technical program managers (PMs) release changes. Changes can range from platform framework updates to change requests made to BI semantic models. They're the last line of defense to ensure changes don't break anything.

Dedicated deliveries

There's a dedicated delivery team for each stakeholder group. It typically consists of a data engineer, an analytics engineer, and a technical PM—all funded by their stakeholder group.

BI team roles

At Microsoft, our BI platform is operated by scalable teams of professionals. Teams are aligned to dedicated and shared resources. Today, we have the following roles:

- **Program managers**: PMs are a dedicated resource. They act as the primary contact between the BI team and stakeholders. It's their job to translate stakeholder business requirements to a technical specification. And, they manage the prioritization of stakeholder deliverables.
- **Database leads**: They're a dedicated resource responsible for onboarding new semantic models into the centralized data warehouse. Onboarding a semantic model can involve setting up conformed dimensions, adding business logic and custom attributes, and standard names and formatting.
- **Analytics leads**: They're a dedicated resource responsible for the design and development of BI semantic models. They strive to apply a consistent architecture using standard naming and formatting. Performance optimization is an important part of their role.
- **Operations and infrastructure**: They're a shared resource responsible for managing jobs and data pipelines. They're also responsible for managing Azure subscriptions, Power BI capacities, virtual machines, and data gateways.
- **Support**: They're a shared resource responsible for writing documentation, organizing training, communicating BI semantic model changes, and answering user questions.

Governance and compliance

For each stakeholder group, PM leads provide cross-program governance and oversight. Its overriding goal is to ensure investments in IT generate business value and mitigate risk. Steering committee meetings are held on a regular basis to review progress and approve major initiatives.

Grow your own community

Establish and grow a community within your organization by:

- Holding regular Office hours events that sets aside time with the BI team to allow people to ask questions, make suggestions, share ideas, and even lodge complaints.
- Creating a Teams channel to provide support and encourage anyone to ask and respond to posted questions.
- Run and promote informal user groups and encourage employees to present or attend.
- Run more formal training events on specific products and the BI platform itself. Consider delivering Dashboard in a Day, which is available as a free course kit and is a great way to introduce employees to Power BI for the first time.

Related content

For more information about this article, check out the following resources:

- BI solution architecture in the COE
- Fabric adoption roadmap: Center of Excellence
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

In the next article in this series, learn about BI solution architecture in the COE and the different technologies employed.

Professional services

Certified Power BI partners are available to help your organization succeed when setting up a COE. They can provide you with cost-effective training or an audit of your data. To find a Power BI partner, visit the Microsoft Power BI partners portal.

You can also engage with experienced consulting partners. They can help you assess, evaluate, or implement Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

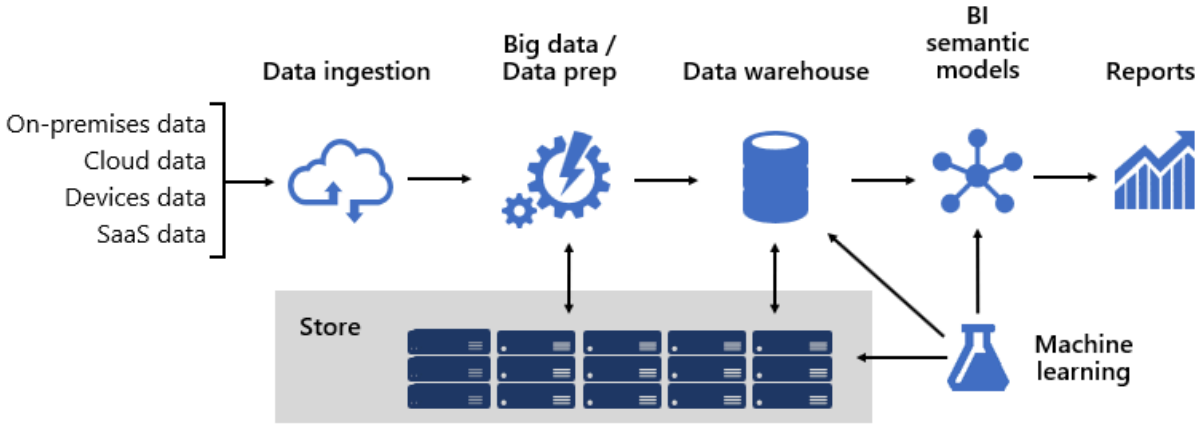
BI solution architecture in the Center of Excellence

Article • 12/30/2024

This article targets IT professionals and IT managers. You'll learn about BI solution architecture in the COE and the different technologies employed. Technologies include Azure, Power BI, and Excel. Together, they can be leveraged to deliver a scalable and data-driven cloud BI platform.

Designing a robust BI platform is somewhat like building a bridge; a bridge that connects transformed and enriched source data to data consumers. The design of such a complex structure requires an engineering mindset, though it can be one of the most creative and rewarding IT architectures you could design. In a large organization, a BI solution architecture can consist of:

- Data sources
- Data ingestion
- Big data / data preparation
- Data warehouse
- BI semantic models
- Reports



The platform must support specific demands. Specifically, it must scale and perform to meet the expectations of business services and data consumers. At the same time, it must be secure from the ground up. And, it must be sufficiently resilient to adapt to change—because it's a certainty that in time new data and subject areas must be brought online.

Frameworks

At Microsoft, from the outset we adopted a systems-like approach by investing in framework development. Technical and business process frameworks increase the reuse of design and logic and provide a consistent outcome. They also offer flexibility in architecture leveraging many technologies, and they streamline and reduce engineering overhead via repeatable processes.

We learned that well-designed frameworks increase visibility into data lineage, impact analysis, business logic maintenance, managing taxonomy, and streamlining governance. Also, development became faster and collaboration across large teams became more responsive and effective.

We'll describe several of our frameworks in this article.

Data models

Data models provide you with control over how data is structured and accessed. To business services and data consumers, data models are their interface with the BI platform.

A BI platform can deliver three different types of models:

- Enterprise models
- BI semantic models
- Machine Learning (ML) models

Enterprise models

Enterprise models are built and maintained by IT architects. They're sometimes referred to as dimensional models or data marts. Typically, data is stored in relational format as dimension and fact tables. These tables store cleansed and enriched data consolidated from many systems and they represent an authoritative source for reporting and analytics.

Enterprise models deliver a consistent and single source of data for reporting and BI. They're built once and shared as a corporate standard. Governance policies ensure data is secure, so access to sensitive data sets—such as customer information or financials—is restricted on a needs-basis. They adopt naming conventions ensuring consistency, thereby further establishing credibility of data and quality.

In a cloud BI platform, enterprise models can be deployed to a Synapse SQL pool in Azure Synapse. The Synapse SQL pool then becomes the single version of truth the organization can count on for fast and robust insights.

BI semantic models

BI semantic models represent a semantic layer over enterprise models. They're built and maintained by BI developers and business users. BI developers create core BI semantic models that source data from enterprise models. Business users can create smaller-scale, independent models—or, they can extend core BI semantic models with departmental or external sources. BI semantic models commonly focus on a single subject area, and are often widely shared.

Business capabilities are enabled not by data alone, but by BI semantic models that describe concepts, relationships, rules, and standards. This way, they represent intuitive and easy-to-understand structures that define data relationships and encapsulate business rules as calculations. They can also enforce fine-grained data permissions, ensuring the right people have access to the right data. Importantly, they accelerate query performance, providing extremely responsive interactive analytics—even over terabytes of data. Like enterprise models, BI semantic models adopt naming conventions ensuring consistency.

In a cloud BI platform, BI developers can deploy BI semantic models to Azure Analysis Services, Power BI Premium capacities of Microsoft Fabric capacities.

> **Important**
>
> At times this article refers to Power BI Premium or its capacity subscriptions (P SKUs). Be aware that Microsoft is currently consolidating purchase options and retiring the Power BI Premium per capacity SKUs. New and existing customers should consider purchasing Fabric capacity subscriptions (F SKUs) instead.
>
> For more information, see **Important update coming to Power BI Premium licensing** and **Power BI Premium FAQ**.

We recommend deploying to Power BI when it's used as your reporting and analytics layer. These products support different storage modes, allowing data model tables to cache their data or to use DirectQuery, which is a technology that passes queries through to the underlying data source. DirectQuery is an ideal storage mode when model tables represent large data volumes or there's a need to deliver near-real time results. The two storage modes can be combined: Composite models combine tables that use different storage modes in a single model.

For heavily queried models, Azure Load Balancer can be used to evenly distribute the query load across model replicas. It also allows you to scale your applications and create highly available BI semantic models.

Machine Learning models

Machine Learning (ML) models are built and maintained by data scientists. They're mostly developed from raw sources in the data lake.

Trained ML models can reveal patterns within your data. In many circumstances, those patterns can be used to make predictions that can be used to enrich data. For example, purchasing behavior can be used to predict customer churn or segment customers. Prediction results can be added to enterprise models to allow analysis by customer segment.

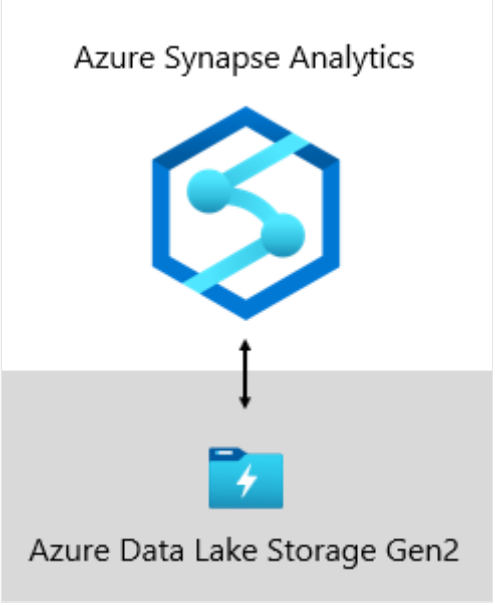
In a cloud BI platform, you can use Azure Machine Learning to train, deploy, automate, manage, and track ML models.

Data warehouse

Sitting at the heart of a BI platform is the data warehouse, which hosts your enterprise models. It's a source of sanctioned data—as a system of record and as a hub—serving enterprise models for reporting, BI, and data science.

Many business services, including line-of-business (LOB) applications, can rely upon the data warehouse as an authoritative and governed source of enterprise knowledge.

At Microsoft, our data warehouse is hosted on Azure Data Lake Storage Gen2 (ADLS Gen2) and Azure Synapse Analytics.



- **ADLS Gen2** makes Azure Storage the foundation for building enterprise data lakes on Azure. It's designed to service multiple petabytes of information while sustaining hundreds of gigabits of throughput. And, it offers low-cost storage

capacity and transactions. What's more, it supports Hadoop compatible access, which allows you to manage and access data just as you would with a Hadoop Distributed File System (HDFS). In fact, Azure HDInsight, Azure Databricks, and Azure Synapse Analytics can all access data stored in ADLS Gen2. So, in a BI platform, it's a good choice to store raw source data, semi-processed or staged data, and production-ready data. We use it to store all our business data.

- **Azure Synapse Analytics** is an analytics service that brings together enterprise data warehousing and Big Data analytics. It gives you the freedom to query data on your terms, using either serverless on-demand or provisioned resources—at scale. Synapse SQL, a component of Azure Synapse Analytics, supports complete T-SQL-based analytics, so it's ideal to host enterprise models comprising your dimension and fact tables. Tables can be efficiently loaded from ADLS Gen2 using simple Polybase T-SQL queries. You then have the power of MPP to run high-performance analytics.

Business Rules Engine framework

We developed a **Business Rules Engine** (BRE) framework to catalog any business logic that can be implemented in the data warehouse layer. A BRE can mean many things, but in the context of a data warehouse it's useful for creating calculated columns in relational tables. These calculated columns are usually represented as mathematical calculations or expressions using conditional statements.

The intention is to split business logic from core BI code. Traditionally, business rules are hard-coded into SQL stored procedures, so it often results in much effort to maintain them when business needs change. In a BRE, business rules are defined once and used multiple times when applied to different data warehouse entities. If calculation logic needs to change, it only needs to be updated in one place and not in numerous stored procedures. There's a side benefit, too: a BRE framework drives transparency and visibility into implemented business logic, which can be exposed via a set of reports that create self-updating documentation.

Data sources

A data warehouse can consolidate data from practically any data source. It's mostly built over LOB data sources, which are commonly relational databases storing subject-specific data for sales, marketing, finance, etc. These databases can be cloud-hosted or they can reside on-premises. Other data sources can be file-based, especially web logs or IOT data sourced from devices. What's more, data can be sourced from Software-as-a-Service (SaaS) vendors.

At Microsoft, some of our internal systems output operational data direct to ADLS Gen2 using raw file formats. In addition to our data lake, other source systems comprise relational LOB applications, Excel workbooks, other file-based sources, and Master Data Management (MDM) and custom data repositories. MDM repositories allow us to manage our master data to ensure authoritative, standardized, and validated versions of data.

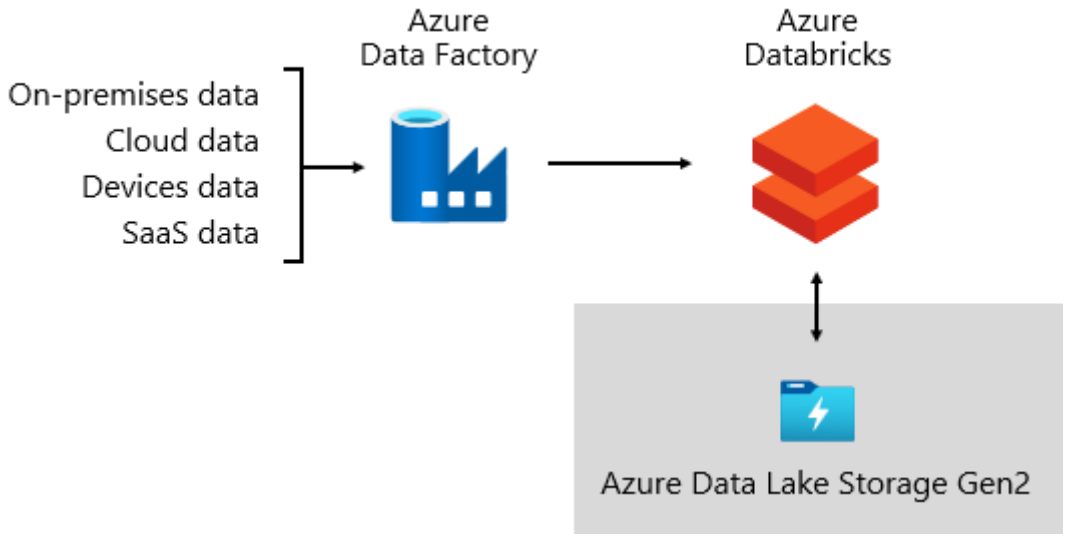
Data ingestion

On a periodic basis, and according to the rhythms of the business, data is ingested from source systems and loaded into the data warehouse. It could be once a day or at more frequent intervals. Data ingestion is concerned with extracting, transforming, and loading data. Or, perhaps the other way round: extracting, loading, and then transforming data. The difference comes down to where the transformation takes place. Transformations are applied to cleanse, conform, integrate, and standardize data. For more information, see Extract, transform, and load (ETL).

Ultimately, the goal is to load the right data into your enterprise model as quickly and efficiently as possible.

At Microsoft, we use Azure Data Factory (ADF). The services is used to schedule and orchestrate data validations, transformations, and bulk loads from external source systems into our data lake. It's managed by custom frameworks to process data in parallel and at scale. In addition, comprehensive logging is undertaken to support troubleshooting, performance monitoring, and to trigger alert notifications when specific conditions are met.

Meanwhile, Azure Databricks—an Apache Spark-based analytics platforms optimized for the Azure cloud services platform—performs transformations specifically for data science. It also builds and executes ML models using Python notebooks. Scores from these ML models are loaded into the data warehouse to integrate predictions with enterprise applications and reports. Because Azure Databricks accesses the data lake files directly, it eliminates or minimizes the need to copy or acquire data.



Ingestion framework

We developed an **ingestion framework** as a set of configuration tables and procedures. It supports a data-driven approach to acquiring large volumes of data at high speed and with minimal code. In short, this framework simplifies the process of data acquisition to load the data warehouse.

The framework depends on configuration tables that store data source and data destination-related information such as source type, server, database, schema, and table-related details. This design approach means we don't need to develop specific ADF pipelines or SQL Server Integration Services (SSIS) packages. Instead, procedures are written in the language of our choice to create ADF pipelines that are dynamically generated and executed at run time. So, data acquisition becomes a configuration exercise that's easily operationalized. Traditionally, it would require extensive development resources to create hard-coded ADF or SSIS packages.

The ingestion framework was designed to simplify the process of handling upstream source schema changes, too. It's easy to update configuration data—manually or automatically, when schema changes are detected to acquire newly added attributes in the source system.

Orchestration framework

We developed an *orchestration framework* to operationalize and orchestrate our data pipelines. The orchestration framework uses a data-driven design that depends on a set of configuration tables. These tables store metadata describing pipeline dependencies and how to map source data to target data structures. The investment in developing this adaptive framework has since paid for itself; there's no longer a requirement to hard-code each data movement.

Data storage

A data lake can store large volumes of raw data for later use along with staging data transformations.

At Microsoft, we use ADLS Gen2 as our single source of truth. It stores raw data alongside staged data and production-ready data. It provides a highly scalable and cost-effective data lake solution for big data analytics. Combining the power of a high-performance file system with massive scale, it's optimized for data analytic workloads, accelerating time to insight.

ADLS Gen2 provides the best of two worlds: it's BLOB storage and a high-performance file system namespace, which we configure with fine-grained access permissions.

Refined data is then stored in a relational database to deliver a high-performance, highly scalable data store for enterprise models, with security, governance, and manageability. Subject-specific data marts are stored in Azure Synapse Analytics, which are loaded by Azure Databricks or Polybase T-SQL queries.

Data consumption

At the reporting layer, business services consume enterprise data sourced from the data warehouse. They also access data directly in the data lake for ad hoc analysis or data science tasks.

Fine-grained permissions are enforced at all layers: in the data lake, enterprise models, and BI semantic models. The permissions ensure data consumers can only see the data they have rights to access.

At Microsoft, we use Power BI reports and dashboards, and Power BI paginated reports. Some reporting and ad hoc analysis is done in Excel—particularly for financial reporting.

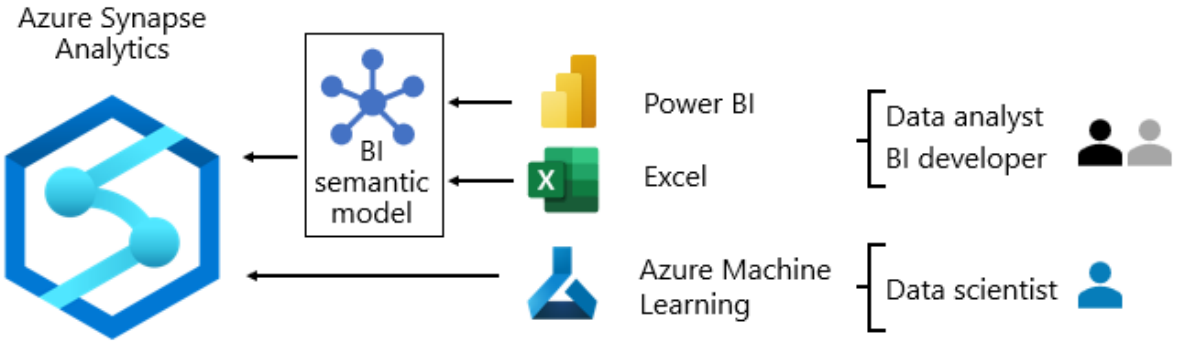
We publish data dictionaries, which provide reference information about our data models. They're made available to our users so they can discover information about our BI platform. Dictionaries document model designs, providing descriptions about entities, formats, structure, data lineage, relationships, and calculations. We use Azure Data Catalog to make our data sources easily discoverable and understandable.

Typically, data consumption patterns differ based on role:

- **Data analysts** connect directly to core BI semantic models. When core BI semantic models contain all data and logic they need, they use live connections to create Power BI reports and dashboards. When they need to extend the models with

departmental data, they create Power BI composite models. If there's a need for spreadsheet-style reports, they use Excel to produce reports based on core BI semantic models or departmental BI semantic models.

- **BI developers** and operational report authors connect directly to enterprise models. They use Power BI Desktop to create live connection analytic reports. They can also author operational-type BI reports as Power BI paginated reports, writing native SQL queries to access data from the Azure Synapse Analytics enterprise models by using T-SQL, or Power BI semantic models by using DAX or MDX.
- **Data scientists** connect directly to data in the data lake. They use Azure Databricks and Python notebooks to develop ML models, which are often experimental and require specialty skills for production use.



Related content

For more information about this article, check out the following resources:

- Fabric adoption roadmap: Center of Excellence
- Enterprise BI in Azure with Azure Synapse Analytics
- Questions? Try asking the Fabric Community
- Suggestions? Contribute ideas to improve Fabric

Professional services

Certified Power BI partners are available to help your organization succeed when setting up a COE. They can provide you with cost-effective training or an audit of your data. To find a Power BI partner, visit the Microsoft Power BI partners portal.

You can also engage with experienced consulting partners. They can help you assess, evaluate, or implement Power BI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

White papers for Power BI

Article • 10/28/2024

White papers allow you to explore Power BI topics at a deeper level. Here you can find a list of available white papers for Power BI.

Expand table

White paper	Description	Date
Power BI and Dataflows	This white paper describes dataflows in technical detail, and describes the capabilities and initiatives behind dataflow features and functionality.	November 2018
Power BI Premium Planning and Deployment	The content of this white paper has been incorporated into general guidance. See the link for guidance and best practices for planning and deploying Premium capacity for well-defined workloads.	March 2019
Capacity planning guidance for Power BI Report Server	This paper aims to offer guidance on capacity planning for Power BI Report Server by sharing results of numerous load test executions of various workloads against a report server.	March 2018
Security	Provides a detailed explanation of security within Power BI.	March 2019
Advanced Analytics with Power BI	Describes the advanced analytics capabilities of Power BI, including predictive analytics, custom visualizations, R integration, and data analysis expressions.	February 2017
Bidirectional filtering	Explains bidirectional cross-filtering in Power BI Desktop (the white paper also covers SQL Server Analysis Services 2016, both have the same behavior).	July 2018
DirectQuery in SQL Server 2016 Analysis Services	For SQL Server 2016, DirectQuery was redesigned for dramatically improved speed and performance, however, it is also now more complex to understand and implement.	January 2017
Power BI and SAP BW	This document describes how SAP customers can benefit from connecting Power BI to their existing SAP Business Warehouse (BW) systems. Updated in November 2019.	November 2019
Securing the Tabular BI Semantic Model	This paper introduces the security model for tabular BI semantic and Power BI. You will learn how to create roles, implement dynamic security, configure impersonation settings, manage roles, and choose a method for connecting to models that works in your network security context.	April 2016
Power BI and GDPR	This link takes you to the list of white papers on the Service Trust Portal, including the Microsoft Power BI GDPR white	April 2018

White paper	Description	Date
	paper.	
Power BI migration	This link takes you to an article that describes how to migrate from other business intelligence tools to Power BI.	September 2020

> **Note**
>
> If you're interested in viewing or deleting personal data, please review Microsoft's guidance in the Windows Data Subject Requests for the GDPR site. If you're looking for general information about GDPR, see the GDPR section of the Service Trust portal.

More questions? Try asking the Power BI Community

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community

Power BI security white paper

06/11/2025

Summary: Power BI is an online software service (*SaaS*, or Software as a Service) offering from Microsoft that lets you easily and quickly create self-service business intelligence dashboards, reports, semantic models, and visualizations. With Power BI, you can connect to many different data sources, combine and shape data from those connections, then create reports and dashboards that can be shared with others.

Writers: Yitzhak Kesselman, Paddy Osborne, Matt Neely, Tony Bencic, Srinivasan Turuvekere, Cristian Petculescu, Adi Regev, Naveen Sivaraj, Ben Glastein, Evgeny Tshiorny, Arthi Ramasubramanian Iyer, Sid Jayadevan, Ronald Chang, Ori Eduar, Anton Fritz, Idan Sheinberg, Ron Gilad, Sagiv Hadaya, Paul Inbar, Igor Uzhviev, Michael Roth, Jaime Tarquino, Gennady Pats, Orion Lee, Yury Berezansky, Maya Shenhav, Romit Chattopadhyay, Yariv Maimon, Bogdan Crivat

Technical Reviewers: Cristian Petculescu, Amir Netz, Sergei Gundorov

Applies to: Power BI SaaS, Power BI Desktop, Power BI Premium, Power BI Embedded Analytics, Power BI Mobile.

> ⓘ **Note**
>
> You can save or print this white paper by selecting **Print** from your browser, then selecting **Save as PDF**.

Introduction

Power BI is an online software service (*SaaS*, or Software as a Service) offering from Microsoft that lets you easily and quickly create self-service business intelligence dashboards, reports, semantic models, and visualizations. With Power BI, you can connect to many different data sources, combine and shape data from those connections, then create reports and dashboards that can be shared with others.

The world is rapidly changing; organizations are going through an accelerated digital transformation, and we're seeing a massive increase in remote working, increased customer demand for online services, and increased use of advanced technologies in operations and business decision-making. And all of this is powered by the cloud.

As the transition to the cloud has changed from a trickle to a flood, and with the new, exposed surface area that comes with it, more companies are asking *How secure is my data in the cloud?*

and *What end-to-end protection is available to prevent my sensitive data from leaking?* And for the BI platforms that often handle some of the most strategic information in the enterprise, these questions are doubly important.

The decades-old foundations of the BI security model—object-level and row-level security—while still important, clearly no longer suffice for providing the kind of security needed in the cloud era. Instead, organizations must look for a cloud-native, multi-tiered, defense-in-depth security solution for their business intelligence data.

Power BI was built to provide industry-leading complete and hermetic protection for data. The product has earned the highest security classifications available in the industry, and today many national security agencies, financial institutions, and health care providers entrust it with their most sensitive information.

It all starts with the foundation. After a rough period in the early 2000s, Microsoft made massive investments to address its security vulnerabilities, and in the following decades built a strong security stack that goes as deep as the machine on-chip bios kernel and extends all the way up to end-user experiences. These deep investments continue, and today over 3,500 Microsoft engineers are engaged in building and enhancing Microsoft's security stack and proactively addressing the ever-shifting threat landscape. With billions of computers, trillions of logins, and countless zettabytes of information entrusted to Microsoft's protection, the company now possesses the most advanced security stack in the tech industry and is broadly viewed as the global leader in the fight against malicious actors.

Power BI builds on this strong foundation. It uses the same security stack that earned Azure the right to serve and protect the world's most sensitive data, and it integrates with the most advanced information protection and compliance tools of Microsoft 365. On top of these, it delivers security through multi-layered security measures, resulting in end-to-end protection designed to deal with the unique challenges of the cloud era.

To provide an end-to-end solution for protecting sensitive assets, the product team needed to address challenging customer concerns on multiple simultaneous fronts:

- *How do we control who can connect, where they connect from, and how they connect? How can we control the connections?*
- *How is the data stored? How is it encrypted? What controls do I have on my data?*
- *How do I control and protect my sensitive data? How do I ensure this data can't leak outside the organization?*
- *How do I audit who conducts what operations? How do I react quickly if there's suspicious activity on the service?*

This article provides a comprehensive answer to all these questions. It starts with an overview of the service architecture and explains how the main flows in the system work. It then moves

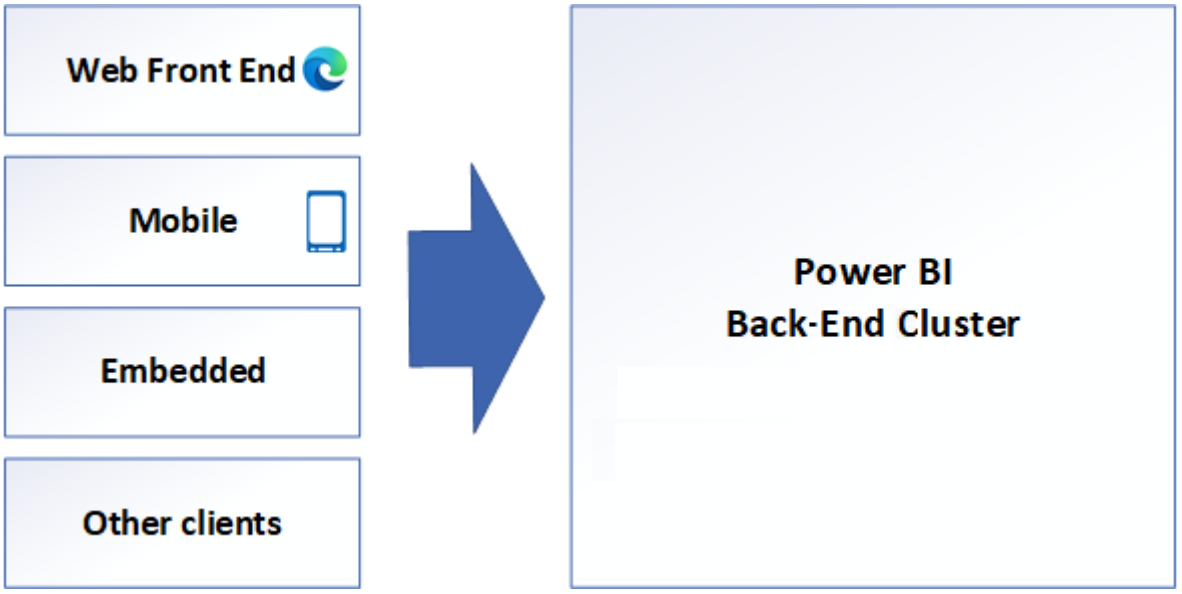
on to describe how users authenticate to Power BI, how data connections are established, and how Power BI stores and moves data through the service. The last section discusses the security features that allow you, as the service admin, to protect your most valuable assets.

The Power BI service is governed by the Microsoft Online Services Terms, and the Microsoft Enterprise Privacy Statement. For the location of data processing, refer to the Location of Data Processing terms in the Microsoft Online Services Terms and to the Data Protection Addendum. For compliance information, the Microsoft Trust Center is the primary resource for Power BI. The Power BI team is working hard to bring its customers the latest innovations and productivity. Learn more about compliance in the Microsoft compliance offerings.

The Power BI service follows the Security Development Lifecycle (SDL), strict security practices that support security assurance and compliance requirements. The SDL helps developers build more secure software by reducing the number and severity of vulnerabilities in software, while reducing development cost. Learn more at Microsoft Security Development Lifecycle Practices.

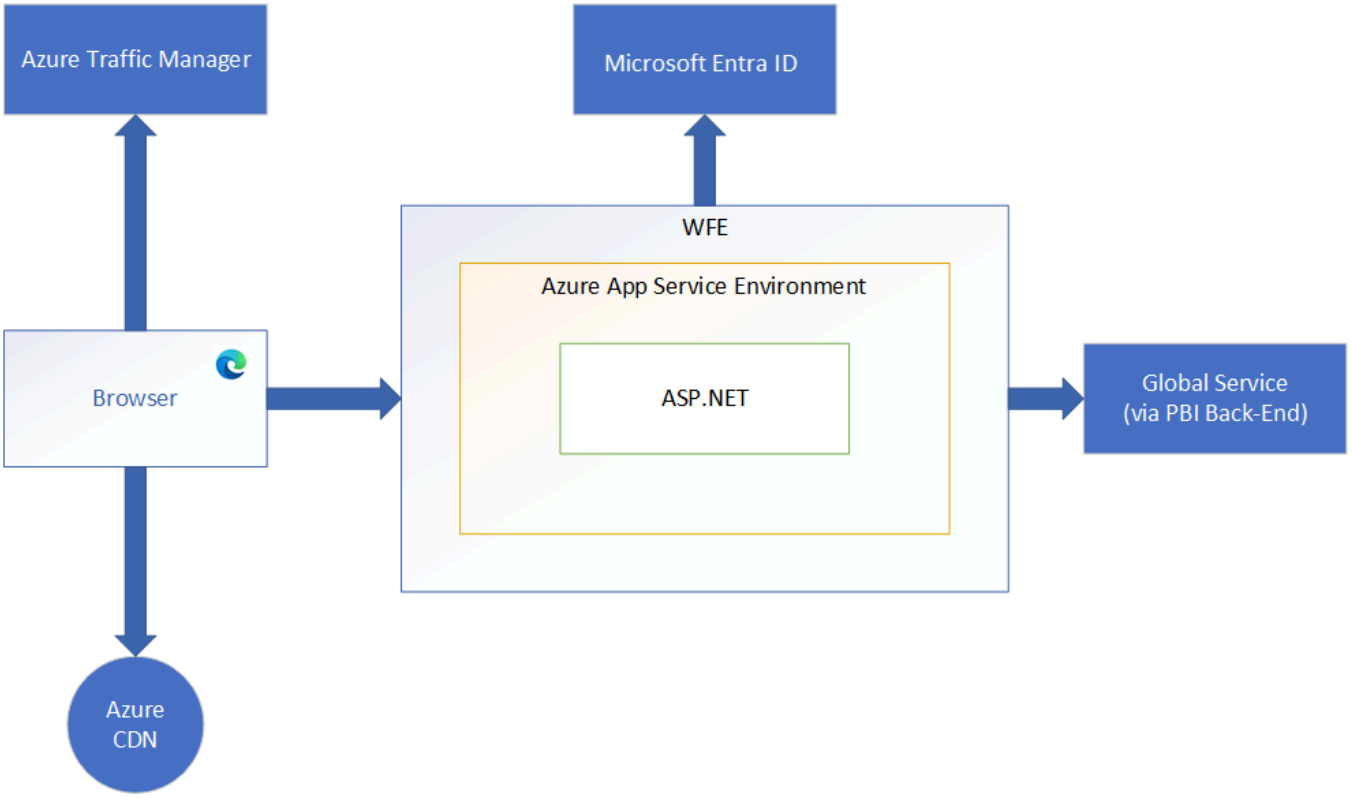
Power BI architecture

The Power BI service is built on Azure, Microsoft's cloud computing platform. Power BI is currently deployed in many datacenters around the world; there are many active deployments made available to customers in the regions served by those datacenters, and an equal number of passive deployments that serve as backups for each active deployment.



Web front-end cluster (WFE)

The web front-end cluster (WFE) cluster provides the user's browser with the initial HTML page contents on site load, and pointers to Azure Content Delivery Network (CDN) content used to render the site in the browser.



A WFE cluster consists of an ASP.NET website running in the Azure App Service Environment. When users attempt to connect to the Power BI service, the client's DNS service might communicate with the Azure Traffic Manager to find the most appropriate (usually nearest) datacenter with a Power BI deployment. For more information about this process, see Performance traffic-routing method for Azure Traffic Manager.

Static resources such as *.*js*, *.*css*, and image files are mostly stored on an Azure CDN and retrieved directly by the browser. Note that Sovereign Government cluster deployments are an exception to this rule, and for compliance reasons will omit the CDN and instead use a WFE cluster from a compliant region for hosting static content.

Power BI back-end cluster

The back-end cluster is the backbone of all the functionality available in Power BI. It consists of several service endpoints consumed by WFE and API clients as well as background working services, databases, caches, and various other components.

The back end is available in most Azure regions, and is being deployed in new regions as they become available. A single Azure region hosts one or more back-end clusters that allow unlimited horizontal scaling of the Power BI service once the vertical and horizontal scaling limits of a single cluster are exhausted.

Each back-end cluster is stateful and hosts all the data of all the tenants assigned to that cluster. A cluster that contains the data of a specific tenant is referred to as the tenant's home

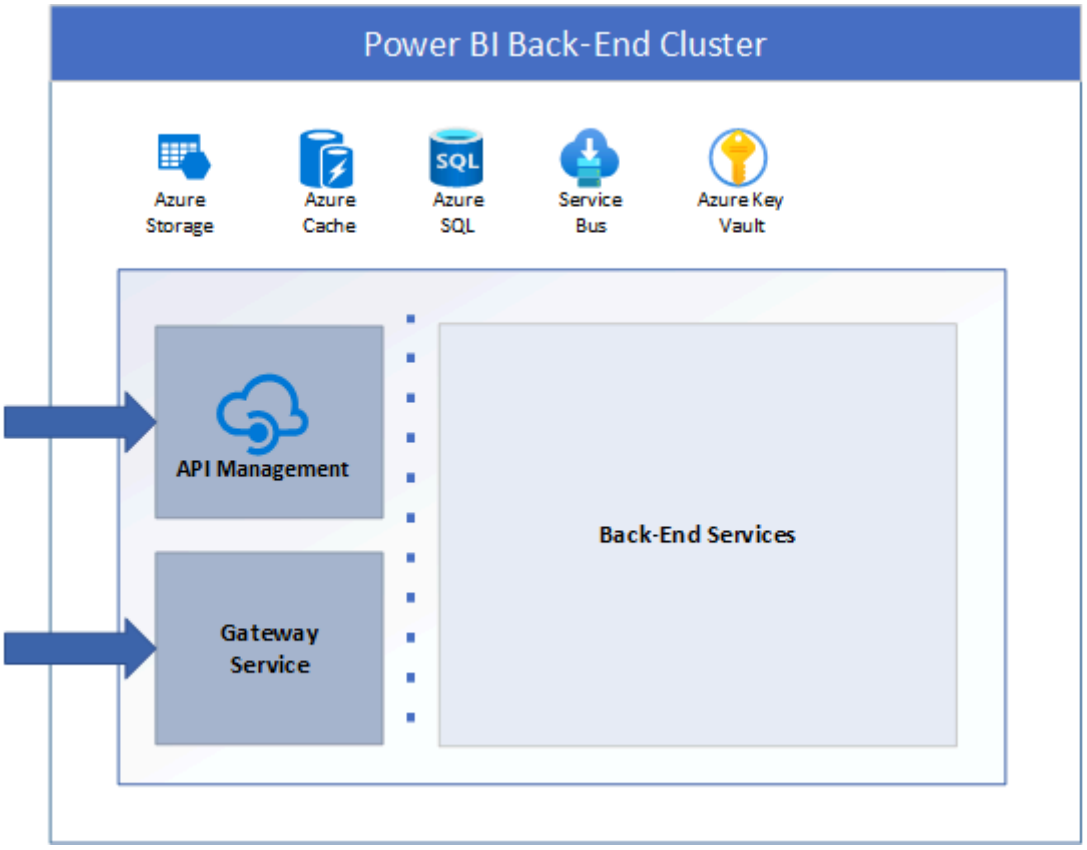
cluster. An authenticated user's home cluster information is provided by Global Service and used by the WFE to route requests to the tenant's home cluster.

Each back-end cluster consists of multiple virtual machines combined into multiple resizable-scale sets tuned for performing specific tasks, stateful resources such as SQL databases, storage accounts, service buses, caches, and other necessary cloud components.

Tenant metadata and data are stored within cluster limits except for data replication to a secondary back-end cluster in a paired Azure region in the same Azure geography. The secondary back-end cluster serves as a failover cluster in case of regional outage, and is passive at any other time.

Back-end functionality is served by micro-services running on different machines within the cluster's virtual network that aren't accessible from the outside, except for two components that can be accessed from the public internet:

- Gateway Service
- Azure API Management

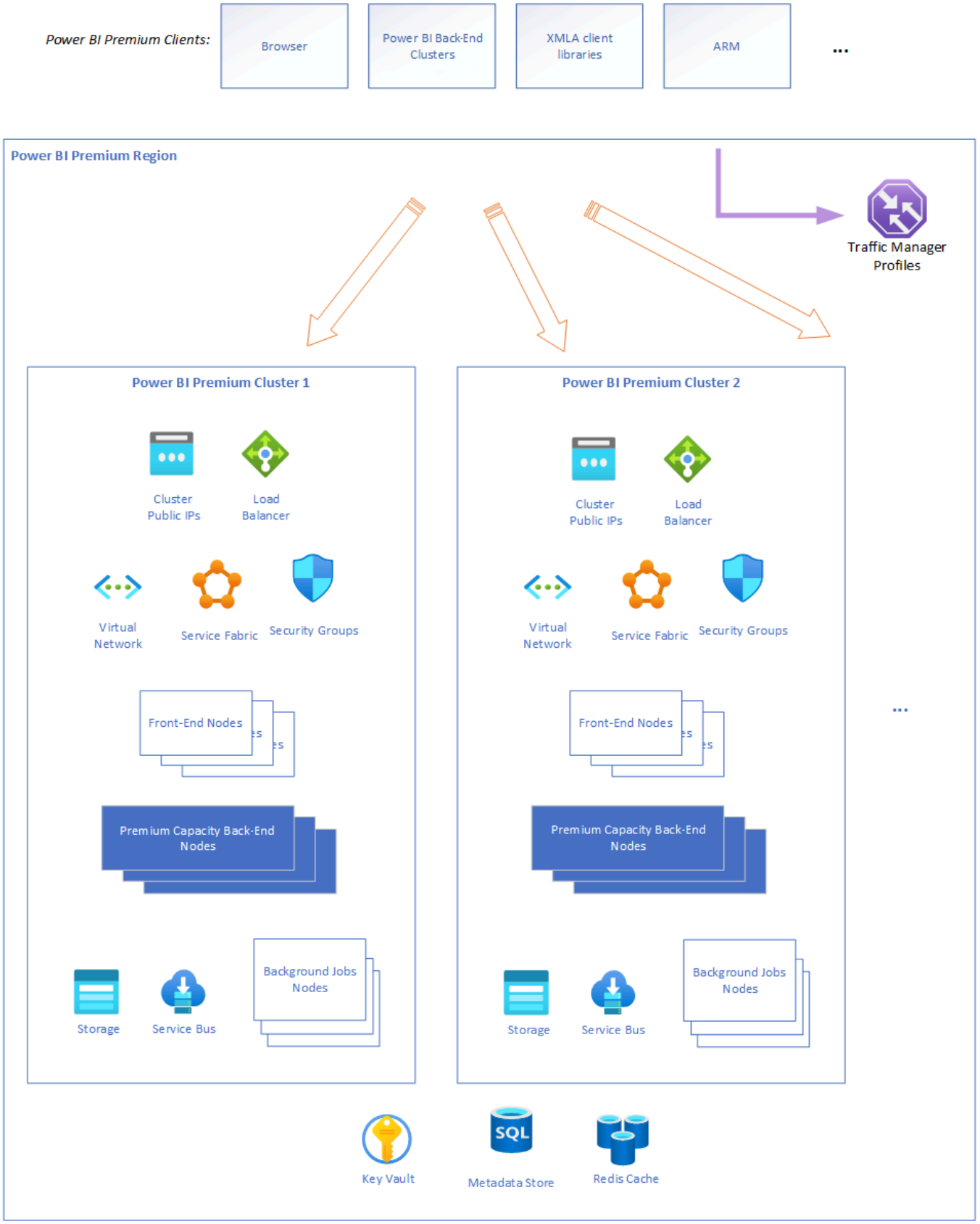


Power BI Premium infrastructure

Power BI Premium offers a service for subscribers who require premium Power BI features, such as advanced AI, distribution to unlicensed users, etc. When a customer signs up for a Power BI Premium subscription, the Premium capacity is created through the Azure Resource Manager.

Power BI Premium capacities are hosted in back-end clusters that are independent of the regular Power BI back end, as described above. This provides better isolation, resource allocation, supportability, security isolation, and scalability of the Premium offering.

The following diagram illustrates the architecture of the Power BI Premium infrastructure:



The connection to the Power BI Premium infrastructure can be done in many ways, depending on the user scenario. Power BI Premium clients can be a user's browser, a regular Power BI back

end, direct connections via XMLA clients, ARM APIs, etc.

The Power BI Premium infrastructure in an Azure region consists of multiple Power BI Premium clusters (the minimum is one). Most Premium resources are encapsulated inside a cluster (for instance, compute), and there are some common regional resources (for example, metadata storage). Premium infrastructure allows two ways of achieving horizontal scalability in a region: increasing resources inside clusters and/or adding more clusters on demand as needed (if cluster resources are approaching their limits).

The backbone of each cluster is compute resources managed by Virtual Machine Scale Sets and Azure Service Fabric. Virtual Machine Scale Sets and Service Fabric allow fast and painless increase of compute nodes as usage grows and orchestrates the deployment, management, and monitoring of Power BI Premium services and applications.

There are many surrounding resources that ensure a secure and reliable infrastructure: load balancers, virtual networks, network security groups, service bus, storage, etc. Any secrets, keys, and certificates required for Power BI Premium are managed by Azure Key Vault exclusively. Any authentication is done via integration with Microsoft Entra ID exclusively.

Any request that comes to Power BI Premium infrastructure goes to front-end nodes first - they're the only nodes available for external connections. The rest of the resources are hidden behind virtual networks. The front-end nodes authenticate the request, handle it, or forward it to the appropriate resources (for example, back-end nodes).

Back-end nodes provide most of the Power BI Premium capabilities and features.

Power BI Mobile

Power BI Mobile is a collection of apps designed for the three primary mobile platforms: Android, iOS, and Windows (UWP). Security considerations for the Power BI Mobile apps fall into two categories:

- Device communication
- The application and data on the device

For device communication, all Power BI Mobile applications communicate with the Power BI service, and use the same connection and authentication sequences used by browsers, which are described in detail earlier in this white paper. The Power BI mobile applications for iOS and Android bring up a browser session within the application itself, while the Windows mobile app brings up a broker to establish the communication channel with Power BI (for the sign-in process).

The following table shows certificate-based authentication (CBA) support for Power BI Mobile, based on the mobile device platform:

Expand table

CBA support	iOS	Android	Windows
Power BI (sign in to service)	Supported	Supported	Not supported
SSRS ADFS on-premises (connect to SSRS server)	Not supported	Supported	Not supported
SSRS App Proxy	Supported	Supported	Not supported

Power BI Mobile apps actively communicate with the Power BI service. Telemetry is used to gather mobile app usage statistics and similar data, which is transmitted to services that are used to monitor usage and activity; no customer data is sent with telemetry.

The Power BI application stores data on the device that facilitates use of the app:

- Microsoft Entra ID and refresh tokens are stored in a secure mechanism on the device, using industry-standard security measures.
- Data and settings (key-value pairs for user configuration) is cached in storage on the device, and can be encrypted by the OS. In iOS this is automatically done when the user sets a passcode. In Android this can be configured in the settings. In Windows it's accomplished by using BitLocker.
- For the Android and iOS apps, the data and settings (key-value pairs for user configuration) are cached in storage on the device in a sandbox and internal storage that is accessible only to the app.
- Geolocation is enabled or disabled explicitly by the user. If enabled, geolocation data isn't saved on the device and isn't shared with Microsoft.
- Notifications are enabled or disabled explicitly by the user. If enabled, Android and iOS don't support geographic data residency requirements for notifications.

Data encryption can be enhanced by applying file-level encryption via Microsoft Intune, a software service that provides mobile device and application management. All three platforms for which Power BI Mobile is available support Intune. With Intune enabled and configured, data on the mobile device is encrypted, and the Power BI application itself can't be installed on an SD card. Learn more about Microsoft Intune.

The Windows app also supports Windows Information Protection (WIP).

In order to implement SSO, some secured storage values related to the token-based authentication are available for other Microsoft first-party apps (such as Microsoft Authenticator) and are managed by the Microsoft Authentication Library (MSAL).

Power BI Mobile cached data is deleted when the app is removed, when the user signs out of Power BI Mobile, or when the user fails to sign in (such as after a token expiration event or password change). The data cache includes dashboards and reports previously accessed from the Power BI Mobile app.

Power BI Mobile doesn't access other application folders or files on the device.

The Power BI apps for iOS and Android let you protect your data by configuring additional identification, such as providing Face ID, Touch ID, or a passcode for iOS, and biometric data (Fingerprint ID) for Android. Learn more about additional identification.

Authentication to the Power BI service

User authentication to the Power BI service consists of a series of requests, responses, and redirects between the user's browser and the Power BI service or the Azure services used by Power BI. That sequence describes the process of user authentication in Power BI, which follows the Microsoft Entra auth code grant flow. For more information about options for an organization's user authentication models (sign-in models), see Choosing a sign-in model for Microsoft 365.

Authentication sequence

The user authentication sequence for the Power BI service occurs as described in the following steps, which are illustrated in the image that follows them.

1. A user initiates a connection to the Power BI service from a browser, either by typing in the Power BI address in the address bar or by selecting **Sign in** from the Power BI marketing page (https://powerbi.microsoft.com). The connection is established using TLS and HTTPS, and all subsequent communication between the browser and the Power BI service uses HTTPS.

2. The Azure Traffic Manager checks the user's DNS record to determine the most appropriate (usually nearest) datacenter where Power BI is deployed, and responds to the DNS with the IP address of the WFE cluster to which the user should be sent.

3. WFE then returns an HTML page to the browser client, which contains a MSAL.js library reference necessary to initiate the sign-in flow.

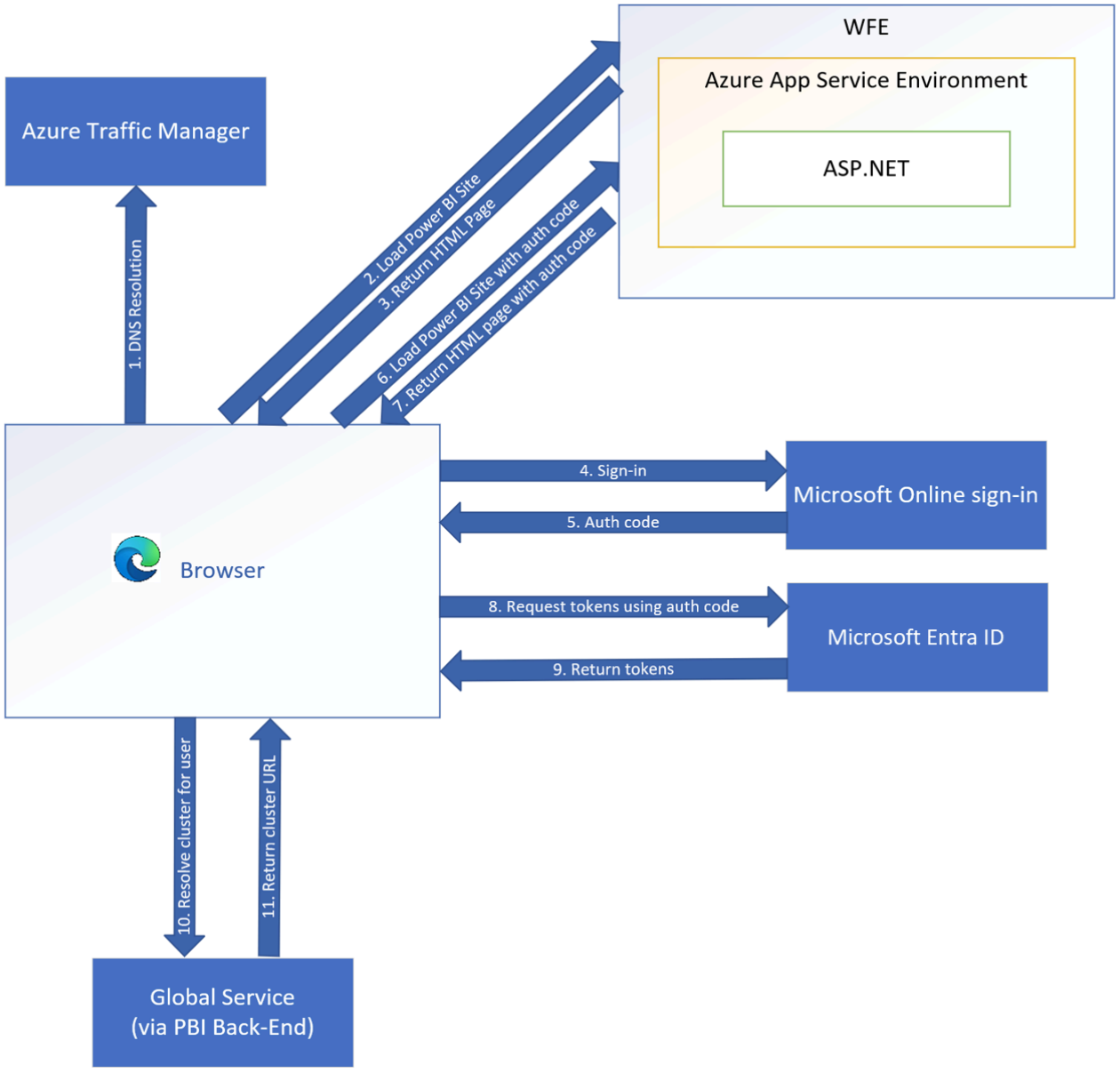
4. The browser client loads the HTML page received from the WFE, and redirects the user to the Microsoft Online Services sign-in page.

5. After the user has been authenticated, the sign-in page redirects the user back to the Power BI WFE page with an auth code.

6. The browser client loads the HTML page, and uses the auth code to request tokens (access, ID, refresh) from Microsoft Entra ID.

7. The user's tenant ID is used by the browser client to query the Power BI Global Service, which maintains a list of tenants and their Power BI back-end cluster locations. The Power BI Global Service determines which Power BI back-end service cluster contains the user's tenant, and returns the Power BI back-end cluster URL back down to the client.

8. The client is now able to communicate with the Power BI back-end cluster URL API, using the access token in the Authorization header for the HTTP requests. The Microsoft Entra access token will have an expiry date set according to Microsoft Entra policies, and to maintain the current session the Power BI Client in the user's browser will make periodic requests to renew the access token before it expires.



In the rare cases where client-side authentication fails due to an unexpected error, the code attempts to fall back to using server-side authentication in the WFE. Refer to the questions and

answers section at the end of this document for details about the server-side authentication flow.

Data residency

Unless otherwise indicated in the documentation, Power BI stores customer data in an Azure geography that is assigned when a Microsoft Entra tenant signs up for Power BI services for the first time. A Microsoft Entra tenant houses the user and application identities, groups, and other relevant information that pertain to an organization and its security.

The assignment of an Azure geography for tenant data storage is done by mapping the country or region selected as part of the Microsoft Entra tenant setup to the most suitable Azure geography where a Power BI deployment exists. Once this determination is made, all Power BI customer data will be stored in this selected Azure geography (also known as the *home geo*), except in cases where organizations utilize multi-geo deployments.

Multiple geographies (multi-geo)

Some organizations have a global presence and might require Power BI services in multiple Azure geographies. For example, a business might have their headquarters in the United States but might also do business in other geographical areas, such as Australia. In such cases, the business might require that certain Power BI data remain stored at rest in the remote region to comply with local regulations. This feature of the Power BI service is referred to as *multi-geo*.

The query execution layer, query caches, and artifact data assigned to a multi-geo workspace are hosted and remain in the remote capacity Azure geography. However, some artifact metadata, such as report structure, might remain stored at rest in the tenant's home geo. Additionally, some data transit and processing might still happen in the tenant's home geo, even for workspaces that are hosted in a multi-geo Premium capacity.

For more information about creating and managing deployments that span multiple Azure geographies, see Configure Multi-Geo support for Fabric.

Regions and datacenters

Power BI services are available in specific Azure geographies as described in the Microsoft Trust Center. For more information about where your data is stored and how it's used, refer to the Microsoft Trust Center. Commitments concerning the location of customer data at rest are specified in the Data Processing Terms of the Microsoft Online Services Terms.

Microsoft also provides datacenters for sovereign entities. For more information about Power BI service availability for national/regional clouds, see Power BI national/regional clouds.

Data handling

This section outlines Power BI data handling practices when it comes to storing, processing, and transferring customer data.

Data at rest

Power BI uses two primary data storage resource types:

- Azure Storage
- Azure SQL databases

In most scenarios, Azure Storage is utilized to persist the data of Power BI artifacts, while Azure SQL databases are used to persist artifact metadata.

All data persisted by Power BI is encrypted by default using Microsoft-managed keys. Customer data stored in Azure SQL databases is fully encrypted using Azure SQL's transparent data encryption (TDE) technology. Customer data stored in Azure Blob Storage is encrypted using Azure Storage encryption.

Optionally, organizations can utilize Power BI Premium to use their own keys to encrypt data at rest that is imported into a semantic model. This approach is often described as bring your own key (BYOK). Utilizing BYOK helps ensure that even in case of a service operator error, customer data won't be exposed – something that can't easily be achieved using transparent service-side encryption. For more information, see Bring your own encryption keys for Power BI.

Power BI semantic models allow for various data source connection modes that determine whether the data source data is persisted in the service or not.

Expand table

Semantic Model Mode (Kind)	Data Persisted in Power BI
Import	Yes
DirectQuery	No
Live Connect	No
Composite	If contains an Import data source
Streaming	If configured to persist

Regardless of the semantic model mode utilized, Power BI might temporarily cache any retrieved data to optimize query and report load performance.

Data in processing

Data is in processing when it's either actively being used by one or more users as part of an interactive scenario, or when a background process, such as refresh, touches this data. Power BI loads actively processed data into the memory space of one or more service workloads. To facilitate the functionality required by the workload, the processed data in memory isn't encrypted.

Data in transit

Power BI requires all incoming HTTP traffic to be encrypted using TLS 1.2 or above. Any requests attempting to use the service with TLS 1.1 or lower will be rejected.

Authentication to data sources

When connecting to a data source, a user can choose to import a copy of the data into Power BI or to connect directly to the data source.

In the case of import, a user establishes a connection based on the user's sign-in and accesses the data with the credential. After the semantic model is published to the Power BI service, Power BI always uses this user's credential to import data. Once data is imported, viewing the data in reports and dashboards doesn't access the underlying data source. Power BI supports single sign-on authentication for selected data sources. If the connection is configured to use single sign-on, the semantic model owner's credentials are used to connect to the data source.

If a data source is connected directly using preconfigured credentials, the preconfigured credentials are used to connect to the data source when any user views the data. If a data source is connected directly using single sign-on, the current user's credentials are used to connect to the data source when a user views the data. When used with single sign-on, row-level security (RLS) and/or object-level security (OLS) can be implemented on the data source. This allows users to view only data they have privileges to access. When the connection is to data sources in the cloud, Microsoft Entra authentication is used for single sign-on; for on-premises data sources, Kerberos, Security Assertion Markup Language (SAML), and Microsoft Entra ID are supported.

If the data source is Azure Analysis Services or on-premises Analysis Services, and RLS and/or OLS is configured, the Power BI service will apply that level security, and users who don't have sufficient credentials to access the underlying data (which could be a query used in a

dashboard, report, or other data artifact) won't see data they don't have sufficient privileges for.

Premium features

Dataflows architecture

Dataflows provide users the ability to configure back-end data processing operations that will extract data from polymorphous data sources, execute transformation logic against the data, and then land it in a target model for use across various reporting presentation technologies. Any user who has either a Member, Contributor, or Admin role in a workspace mighty create a dataflow. Users in the Viewer role might view data processed by the dataflow but might not make changes to its composition. Once a dataflow has been authored, any member, contributor, or admin of the workspace might schedule refreshes, as well as view and edit the dataflow by taking ownership of it.

Each configured data source is bound to a client technology for accessing that data source. The structure of credentials required to access them is formed to match required implementation details of the data source. Transformation logic is applied by Power Query services while the data is in flight. For premium dataflows, Power Query services execute in back-end nodes. Data might be pulled directly from the cloud sources or through a gateway installed on premises. When pulled directly from a cloud source to the service or to the gateway, the transport uses protection methodology specific to the client technology, if applicable. When data is transferred from the gateway to the cloud service, it's encrypted. See the Data in transit section above.

When customer-specified data sources require credentials for access, the owner/creator of the dataflow will provide them during authoring. They're stored using standard product-wide credential storage. See Authentication to data sources above. There are various approaches users might configure to optimize data persistence and access. By default, the data is placed in a Power BI-owned and protected storage account. Storage encryption is enabled on the Blob storage containers to protect the data while it is at rest. See Data at rest below. Users may, however, configure their own storage account associated with their own Azure subscription. When doing so, a Power BI service principal is granted access to that storage account so that it may write the data there during refresh. In this case, the storage resource owner is responsible for configuring encryption on the configured Azure Data Lake Storage account. Data is always transmitted to Blob storage using encryption.

Since performance when accessing storage accounts might be suboptimal for some data, users also have the option to use a Power BI-hosted compute engine to increase performance. In this case, data is redundantly stored in an SQL database that is available for DirectQuery through

access by the back-end Power BI system. Data is always encrypted on the file system. If the user provides a key for encrypting the data stored in the SQL database, that key will be used to doubly encrypt it.

When querying using DirectQuery, the encrypted transport protocol HTTPS is used to access the API. All secondary or indirect use of DirectQuery is controlled by the same access controls previously described. Since dataflows are always bound to a workspace, access to the data is always gated by the user's role in that workspace. A user must have at least read access to be able to query the data via any means.

When Power BI Desktop is used to access data in a dataflow, it must first authenticate the user using Microsoft Entra ID to determine if the user has sufficient rights to view the data. If so, a SaS key is acquired and used to access storage directly using the encrypted transport protocol HTTPS.

The processing of data throughout the pipeline emits Office 365 auditing events. Some of these events will capture operations related to security and privacy.

Paginated reports

Paginated reports are designed to be printed or shared. They're called paginated because they're formatted to fit well on a page. They display all the data in a table, even if the table spans multiple pages. You can control their report page layout exactly.

Paginated reports support rich and powerful expressions written in Microsoft Visual Basic .NET. Expressions are widely used throughout Power BI Report Builder paginated reports to retrieve, calculate, display, group, sort, filter, parameterize, and format data.

Expressions are created by the author of the report with access to the broad range of features of the .NET Framework. The processing and execution of paginated reports is performed inside a sandbox.

Paginated report definitions (.rdl) are stored in Power BI, and to publish and/or render a paginated report a user needs to authenticate and authorize in the same way as described in Authentication to the Power BI service.

The Microsoft Entra token obtained during the authentication is used to communicate directly from the browser to the Power BI Premium cluster.

In Power BI Premium, the Power BI service runtime provides an appropriately isolated execution environment for each report render. This includes cases where the reports being rendered belong to workspaces assigned to the same capacity.

A paginated report can access a wide set of data sources as part of the rendering of the report. The sandbox doesn't communicate directly with any of the data sources but instead communicates with the trusted process to request data, and then the trusted process appends the required credentials to the connection. In this way, the sandbox never has access to any credential or secret.

In order to support features such as Bing Maps, or calls to Azure Functions, the sandbox does have access to the internet.

Power BI embedded analytics

Independent software vendors (ISVs) and solution providers have two main modes of embedding Power BI artifacts in their web applications and portals: embed for your organization and embed for your customers. The artifact is embedded into an IFrame in the application or portal. An IFrame is not allowed to read or write data from the external web application or portal, and the communication with the IFrame is done by using the Power BI Client SDK using POST messages.

In an embed for your organization scenario, Microsoft Entra users access their own Power BI content through portals customized by their enterprises and ITs. All Power BI policies and capabilities described in this paper such as RLS and OLS are automatically applied to all users independently of whether they access Power BI through the Power BI portal or through customized portals.

In an embed for your customers scenario, ISVs typically own Power BI tenants and Power BI items (dashboards, reports, semantic models, and others). It's the responsibility of an ISV back-end service to authenticate its end users and decide which artifacts and which access level is appropriate for that end user. ISV policy decisions are encrypted in an embed token generated by Power BI and passed to the ISV back-end for further distribution to the end users according to the business logic of the ISV. End users using a browser or other client applications aren't able to decrypt or modify embed tokens. Client-side SDKs such as Power BI Client APIs automatically append the encrypted embed token to Power BI requests as an *Authorization: EmbedToken* header. Based on this header, Power BI will enforce all policies (such as access or RLS) precisely as was specified by the ISV during generation.

To enable embedding and automation, and to generate the embed tokens described above, Power BI exposes a rich set of REST APIs. These Power BI REST APIs support both user-delegated and service principal Microsoft Entra methods of authentication and authorization.

Power BI embedded analytics and its REST APIs support all Power BI network isolation capabilities described in this article including service tags and Private Links.

AI features

Power BI currently supports two broad categories of AI features in the product today: AI visuals and AI enrichments. The visual-level AI features include capabilities such as Key Influencers, Decomposition Tree, Smart Narrative, Anomaly Detection, R visual, Python visual, Clustering, Forecasting, Q&A, Quick Insights, etc. The AI enrichment capabilities include capabilities such as AutoML, Azure Machine Learning, CognitiveServices, R/Python transforms etc.

Most of the features mentioned above are supported in both Shared and Premium workspaces today. However, AutoML and CognitiveServices are supported only in Premium workspaces, due to IP restrictions. Today, with the AutoML integration in Power BI, a user can build and train a custom machine learning (ML) model (for example, Prediction, Classification, Regression, etc.) and apply it to get predictions while loading data into a dataflow defined in a Premium workspace. Additionally, Power BI users can apply several CognitiveServices APIs, such as TextAnalytics and ImageTagging, to transform data before loading it into a dataflow/semantic model defined in a Premium workspace.

The Premium AI enrichment features can be best viewed as a collection of stateless AI functions/transforms that can be used by Power BI users in their data integration pipelines used by a Power BI semantic model or dataflow. Note that these functions can also be accessed from current dataflow/semantic model authoring environments in the Power BI service and Power BI Desktop. These AI functions/transforms always run in a Premium workspace/capacity. These functions are surfaced in Power BI as a data source that requires a Microsoft Entra token for the Power BI user who is using the AI function. These AI data sources are special because they don't surface any of their own data and they only supply these functions/transforms. During execution, these features don't make any outbound calls to other services to transmit the customer's data. Let us look at the Premium scenarios individually to understand the communication patterns and relevant security-related details pertaining to them.

For training and applying an AutoML model, Power BI uses the Azure AutoML SDK and runs all the training in the customer's Power BI capacity. During training iterations, Power BI calls an experimentation Azure Machine Learning service to select a suitable model and hyper-parameters for the current iteration. In this outbound call, only relevant experiment metadata (for example, accuracy, ml algorithm, algorithm parameters, etc.) from the previous iteration is sent. The AutoML training produces an ONNX model and training report data that is then saved in the dataflow. Later, Power BI users can then apply the trained ML model as a transform to operationalize the ML model on a scheduled basis. For TextAnalytics and ImageTagging APIs, Power BI doesn't directly call the CognitiveServices service APIs, but rather uses an internal SDK to run the APIs in the Power BI Premium capacity. Today these APIs are supported in both Power BI dataflows and semantic models. While authoring a data model in Power BI Desktop, users can only access this functionality if they have access to a Premium

Power BI workspace. Hence customers are prompted to supply their Microsoft Entra credentials.

Network isolation

This section outlines advanced security features in Power BI. Some of the features have specific licensing requirements. See the sections below for details.

Service tags

A service tag represents a group of IP address prefixes from a given Azure service. It helps minimize the complexity of frequent updates to network security rules. Customers can use service tags to define network access controls on network security groups or Azure Firewall. Customers can use service tags in place of specific IP addresses when creating security rules. By specifying the service tag name (such as `PowerBI`) in the appropriate source or destination (for APIs) field of a rule, customers can allow or deny the traffic for the corresponding service. Microsoft manages the address prefixes encompassed by the service tag and automatically updates the service tag as addresses change.

Private Link integration

Azure networking provides the Azure Private Link feature that enables Power BI to provide secure access via Azure Networking private endpoints. With Azure Private Link and private endpoints, data traffic is sent privately using Microsoft's backbone network infrastructure, and thus the data doesn't traverse the internet.

Private Link ensures that Power BI users use the Microsoft private network backbone when going to resources in the Power BI service.

Using Private Link with Power BI provides the following benefits:

- Private Link ensures that traffic will flow over the Azure backbone to a private endpoint for Azure cloud-based resources.
- Network traffic isolation from non-Azure-based infrastructure, such as on-premises access, would require customers to have ExpressRoute or a VPN configured.

See Private links for secure access to Fabric for additional information.

VNet connectivity

While the Private Link integration feature provides secure inbound connections to Power BI, the VNet connectivity feature enables secure outbound connectivity from Power BI to data sources within a VNet.

VNet gateways (Microsoft-managed) eliminate the overhead of installing and monitoring on-premises data gateways for connecting to data sources associated with a VNet. However, they still follow the familiar process of managing security and data sources, as with an on-premises data gateway.

The following is an overview of what happens when you interact with a Power BI report that is connected to a data source within a VNet using VNet gateways:

1. The Power BI cloud service (or one of the other supported cloud services) kicks off a query and sends the query, data source details, and credentials to the Power Platform VNet (PP VNet) service.

2. The PP VNet service then securely injects a container running a VNet gateway into the subnet. This container can now connect to data services accessible from within this subnet.

3. The PP VNet service then sends the query, data source details, and credentials to the VNet gateway.

4. The VNet gateway gets the query and connects to the data sources with those credentials.

5. The query is then sent to the data source for execution.

6. After execution, the results are sent to the VNet gateway, and the PP VNet service securely pushes the data from the container to the Power BI cloud service.

Service principals

Power BI supports the use of service principals. Store any service principal credentials used for encrypting or accessing Power BI in a Key Vault, assign proper access policies to the vault, and regularly review access permissions.

See Automate Premium workspace and semantic model tasks with service principals for additional details.

Microsoft Purview for Power BI

Microsoft Purview Information Protection

Power BI is deeply integrated with Microsoft Purview Information Protection. Microsoft Purview Information Protection enables organizations to have a single, integrated solution for classification, labeling, auditing, and compliance across Azure, Power BI, and Office.

When information protection is enabled in Power BI:

- Sensitive data, both in the Power BI service and in Power BI Desktop, can be classified and labeled using the same sensitivity labels used in Office and in Azure.
- Governance policies can be enforced when Power BI content is exported to Excel, PowerPoint, PDF, Word, or *.pbix* files, to help ensure that data is protected even when it leaves Power BI.
- It's easy to classify and protect *.pbix* files in Power BI Desktop, just like it's done in Excel, Word, and PowerPoint applications. Files can be easily tagged according to their level of sensitivity. Even further, they can be encrypted if they contain business-confidential data, ensuring that only authorized users can edit these files.
- Excel workbooks automatically inherit sensitivity labels when they connect to Power BI, making it possible to maintain end-to-end classification and apply protection when Power BI semantic models are analyzed in Excel.
- Sensitivity labels applied to Power BI reports and dashboards are visible in the Power BI iOS and Android mobile apps.
- Sensitivity labels persist when a Power BI report is embedded in Teams, SharePoint, or a secure website. This helps organizations maintain classification and protection upon export when embedding Power BI content.
- Label inheritance upon the creation of new content in the Power BI service ensures that labels applied to semantic models or datamarts in the Power BI service will be applied to new content created on top of those semantic models and datamarts.
- Power BI admin scan APIs can extract a Power BI item's sensitivity label, enabling Power BI and InfoSec admins to monitor labeling in the Power BI service and produce executive reports.
- Power BI admin APIs enable central teams to programmatically apply sensitivity labels to content in the Power BI service.
- Central teams can create mandatory label policies to enforce applying labels on new or edited content in Power BI.
- Central teams can create default label policies to ensure that a sensitivity label is applied to all new or changed Power BI content.
- Automatic downstream sensitivity labeling in the Power BI service ensures that when a label on a semantic model or datamart is applied or changed, the label will automatically be applied or changed on all downstream content connected to the semantic model or datamart.

For more information, see Sensitivity labels in Power BI.

Microsoft Purview Data Loss Prevention (DLP) Policies for Power BI

Microsoft Purview's Data Loss Prevention (DLP) policies help organizations reduce the risk of sensitive business data leakage from Power BI. DLP policies help them meet compliance requirements of government or industry regulations, such as the European Union's General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA), and make sure their data in Power BI is managed.

When DLP policies for Power BI are set up:

- All semantic models within the workspaces specified in the policy are evaluated by the policy.
- You can detect when sensitive data is uploaded into your Premium capacities. DLP policies can detect:
  - Sensitivity labels.
  - Sensitive info types. Over 260 types are supported. Sensitive info type detection relies on Microsoft Purview content scanning.
- When you encounter a semantic model identified as sensitive, you can see a customized policy tip that helps you understand what you should do.
- If you're a semantic model owner, you can override a policy and prevent your semantic model from being identified as "sensitive" if you have a valid reason for doing so.
- If you're a semantic model owner, you can report an issue with a policy if you conclude that a sensitive info type has been falsely identified.
- Automatic risk mitigations, such as alerts to the security admin, can be invoked.

For more information, see Data loss prevention policies for Fabric and Power BI.

Microsoft Defender for Cloud Apps for Power BI

Microsoft Defender for Cloud Apps is one of the world's leading cloud access security brokers, named as leader in Gartner's Magic Quadrant for the cloud access security broker (CASB) market. Defender for Cloud Apps is used to secure the use of cloud apps. It enables organizations to monitor and control, in real time, risky Power BI sessions such as user access from unmanaged devices. Security administrators can define policies to control user actions, such as downloading reports with sensitive information.

With Defender for Cloud Apps, organizations can gain the following DLP capabilities:

- Set real-time controls to enforce risky user sessions in Power BI. For example, if a user connects to Power BI from outside of their country or region, the session can be monitored by the Defender for Cloud Apps real-time controls, and risky actions, such as

downloading data tagged with a "Highly Confidential" sensitivity label, can be blocked immediately.

- Investigate Power BI user activity with the Defender for Cloud Apps activity log. The Defender for Cloud Apps activity log includes Power BI activity as captured in the Office 365 audit log, which contains information about all user and admin activities, as well as sensitivity label information for relevant activities such as apply, change, and remove label. Admins can leverage the Defender for Cloud Apps advanced filters and quick actions for effective issue investigation.
- Create custom policies to alert on suspicious user activity in Power BI. The Defender for Cloud Apps activity policy feature can be leveraged to define your own custom rules, to help you detect user behavior that deviates from the norm, and even possibly act upon it automatically, if it seems too dangerous.
- Work with the Defender for Cloud Apps built-in anomaly detection. The Defender for Cloud Apps anomaly detection policies provide out-of-the-box user behavioral analytics and machine learning so that you're ready from the outset to run advanced threat detection across your cloud environment. When an anomaly detection policy identifies a suspicious behavior, it triggers a security alert.
- Power BI admin role in the Defender for Cloud Apps portal. Defender for Cloud Apps provides an app-specific admin role that can be used to grant Power BI admins only the permissions they need to access Power BI-relevant data in the portal, such as alerts, users at risk, activity logs, and other Power BI-related information.

For additional details, see Using Microsoft Defender for Cloud Apps controls in Power BI.

Power BI security questions and answers

The following questions are common security questions and answers for Power BI. These are organized based on when they were added to this white paper, to facilitate your ability to quickly find new questions and answers when this paper is updated. The newest questions are added to the end of this list.

How do users connect to, and gain access to data sources while using Power BI?

- Power BI manages credentials to data sources for each user for cloud credentials or for connectivity through a personal gateway. Data sources managed by an on-premises data gateway can be shared across the enterprise and permissions to these data sources can be managed by the gateway admin. When configuring a semantic model, the user is allowed to select a credential from their personal store or use an on-premises data gateway to use a shared credential.

  In the import case, a user establishes a connection based on the user's sign in and accesses the data with the credential. After the semantic model is published to Power BI

service, Power BI always uses this user's credential to import data. Once data is imported, viewing the data in reports and dashboards doesn't access the underlying data source. Power BI supports single sign-on authentication for selected data sources. If the connection is configured to use single sign-on, the semantic model owner's credential is used to connect with the data source.

For reports that are connected with DirectQuery, the data source is connected directly using a preconfigured credential. The preconfigured credential is used to connect to the data source when any user views the data. If a data source is connected directly using single sign-on, the current user's credential is used to connect to the data source when the user views the data. When using with single sign-on, RLS and/or OLS can be implemented on the data source, and this allows users to view data they have privileges to access. When the connection is to data sources in the cloud, Microsoft Entra authentication is used for single sign-on; for on-premises data sources, Kerberos, SAML, and Microsoft Entra ID are supported.

When connecting with Kerberos, the user's UPN is passed to the gateway, and using Kerberos constrained delegation, the user is impersonated and connected to the respective data sources. SAML is also supported on the Gateway for SAP HANA data source. More information is available in overview of single sign-on for gateways.

If the data source is Azure Analysis Services or on-premises Analysis Services and RLS and/or OLS is configured, the Power BI service will apply that level security, and users who don't have sufficient credentials to access the underlying data (which could be a query used in a dashboard, report, or other data artifact) won't see data for which the user doesn't have sufficient privileges.

RLS with Power BI can be used to restrict data access for given users. Filters restrict data access at the row level, and you can define filters within a role.

OLS can be used to secure sensitive tables or columns. However, unlike RLS, OLS also secures object names and metadata. This helps prevent malicious users from discovering even the existence of such objects. Secured tables and columns are obscured in the field list when using reporting tools like Excel or Power BI, and moreover, users without permissions can't access secured metadata objects via DAX or any other method. From the standpoint of users without proper access permissions, secured tables and columns simply don't exist.

OLS, together with RLS, enables enhanced enterprise-grade security on reports and semantic models, ensuring that only users with the requisite permissions have access to view and interact with sensitive data.

How is data transferred to Power BI?

- All data requested and transmitted by Power BI is encrypted in transit using HTTPS (except when the data source chosen by the customer doesn't support HTTPS) to connect from the data source to the Power BI service. A secure connection is established with the data provider, and only once that connection is established will data traverse the network.

How does Power BI cache report, dashboard, or model data, and is it secure?

- When a data source is accessed, the Power BI service follows the process outlined in Authentication to data sources.

Do clients cache web page data locally?

- When browser clients access Power BI, the Power BI web servers set the *Cache-Control* directive to *no-store*. The *no-store* directive instructs browsers not to cache the web page being viewed by the user, and not to store the web page in the client's cache folder.

What about role-based security, sharing reports or dashboards, and data connections? How does that work in terms of data access, dashboard viewing, report access or refresh?

- For **non-RLS-enabled** data sources, if a dashboard, report, or data model is shared with other users through Power BI, the data is then available for users with whom it's shared to view and interact with. Power BI *does not* reauthenticate users against the original source of the data; once data is uploaded into Power BI, the user who authenticated against the source data is responsible for managing which other users and groups can view the data.

  When data connections are made to an **RLS-capable** data source, such as an Analysis Services data source, only dashboard data is cached in Power BI. Each time a report or semantic model is viewed or accessed in Power BI that uses data from the RLS-capable data source, the Power BI service accesses the data source to get data based on the user's credentials, and if sufficient permissions exist, the data is loaded into the report or data model for that user. If authentication fails, the user will see an error.

  For more information, see Authentication to data sources.

Our users connect to the same data sources all the time, some of which require credentials that differ from their domain credentials. How can they avoid having to input these credentials each time they make a data connection?

- Power BI offers the Power BI Personal Gateway, which is a feature that lets users create credentials for multiple different data sources, then automatically use those credentials when subsequently accessing each of those data sources. For more information, see Use a personal gateway in Power BI.

Which ports are used by an on-premises data gateway and personal gateway? Are there any domain names that need to be allowed for connectivity purposes?

- The detailed answer to this question is available in the following article: Adjust communication settings for the on-premises data gateway.

When working with the on-premises data gateway, how are recovery keys used and where are they stored? What about secure credential management?

- During gateway installation and configuration, the administrator types in a gateway **Recovery Key**. That **Recovery Key** is used to generate a strong AES symmetric key. An RSA asymmetric key is also created at the same time.

  Those generated keys (RSA and AES) are stored in a file located on the local machine. That file is also encrypted. The contents of the file can only be decrypted by that particular Windows machine, and only by that particular gateway service account.

  When a user enters data source credentials in the Power BI service UI, the credentials are encrypted with the public key in the browser. The gateway decrypts the credentials using the RSA private key and re-encrypts them with an AES symmetric key before the data is stored in the Power BI service. With this process, the Power BI service never has access to the unencrypted data.

Which communication protocols are used by the on-premises data gateway, and how are they secured?

- The gateway supports the following two communications protocols:
  - AMQP 1.0 – TCP + TLS: This protocol requires ports 443, 5671-5672, and 9350-9354 to be open for outgoing communication. This protocol is preferred, since it has lower communication overhead.
  - HTTPS – WebSockets over HTTPS + TLS: This protocol uses port 443 only. The WebSocket is initiated by a single HTTP CONNECT message. Once the channel is established, the communication is essentially TCP+TLS. You can force the gateway to use this protocol by modifying a setting described in the on-premises gateway article.

What is the role of Azure CDN in Power BI?

- As mentioned previously, Power BI uses the Azure CDN to efficiently distribute the necessary static content and files to users based on geographical locale. To go into further detail, the Power BI service uses multiple CDNs to efficiently distribute necessary static content and files to users through the public internet. These static files include product downloads (such as Power BI Desktop, the on-premises data gateway, or Power BI apps from various ISVs), browser configuration files used to initiate and establish any subsequent connections with the Power BI service, as well as the initial secure Power BI sign-in page.

Based on information provided during an initial connection to the Power BI service, a user's browser contacts the specified Azure CDN (or for some files, the WFE) to download the collection of specified common files necessary to enable the browser's interaction with the Power BI service. The browser page then includes the Microsoft Entra token, session information, the location of the associated back-end cluster, and the collection of files downloaded from the Azure CDN and WFE cluster, for the duration of the Power BI service browser session.

For Power BI visuals, does Microsoft perform any security or privacy assessment of the custom visual code prior to publishing items to the Gallery?

- No. It's the customer's responsibility to review and determine whether custom visual code should be relied upon. All custom visual code is operated in a sandbox environment, so that any errant code in a custom visual doesn't adversely affect the rest of the Power BI service.

Are there other Power BI visuals that send information outside the customer network?

- Yes. Bing Maps and ESRI visuals transmit data out of the Power BI service for visuals that use those services.

For template apps, does Microsoft perform any security or privacy assessment of the template app prior to publishing items to the Gallery?

- No. The app publisher is responsible for the content while it's the customer's responsibility to review and determine whether to trust the template app publisher.

Are there template apps that can send information outside the customer network?

- Yes. It's the customer's responsibility to review the publisher's privacy policy and determine whether to install the template app on tenant. The publisher is responsible for informing the customer about the app's behavior and capabilities.

What about data sovereignty? Can we provision tenants in datacenters located in specific geographies, to ensure data doesn't leave the country or region borders?

- Some customers in certain geographies have an option to create a tenant in a national/regional cloud, where data storage and processing is kept separate from all other datacenters. National/Regional clouds have a slightly different type of security, since a separate data trustee operates the national/regional cloud Power BI service on behalf of Microsoft.

  Alternatively, customers can also set up a tenant in a specific region. However, such tenants do not have a separate data trustee from Microsoft. Pricing for national/regional clouds is different from the generally available commercial Power BI service. For more

information about Power BI service availability for national/regional clouds, see Power BI national/regional clouds.

How does Microsoft treat connections for customers who have Power BI Premium subscriptions? Are those connections different than those established for the non-Premium Power BI service?

- The connections established for customers with Power BI Premium subscriptions implement a Microsoft Entra business-to-business (B2B) authorization process, using Microsoft Entra ID to enable access control and authorization. Power BI handles connections from Power BI Premium subscribers to Power BI Premium resources just as it would any other Microsoft Entra user.

How does server-side authentication work in the WFE?

- The user authentication sequence service-side authentication occurs as described in the following steps, which are illustrated in the image that follows them.

1. A user initiates a connection to the Power BI service from a browser, either by typing in the Power BI address in the address bar or by selecting **Sign in** from the Power BI marketing page (https://powerbi.microsoft.com). The connection is established using TLS 1.2 and HTTPS, and all subsequent communication between the browser and the Power BI service uses HTTPS.

2. The Azure Traffic Manager checks the user's DNS record to determine the most appropriate (usually nearest) datacenter where Power BI is deployed, and responds to the DNS with the IP address of the WFE cluster to which the user should be sent.

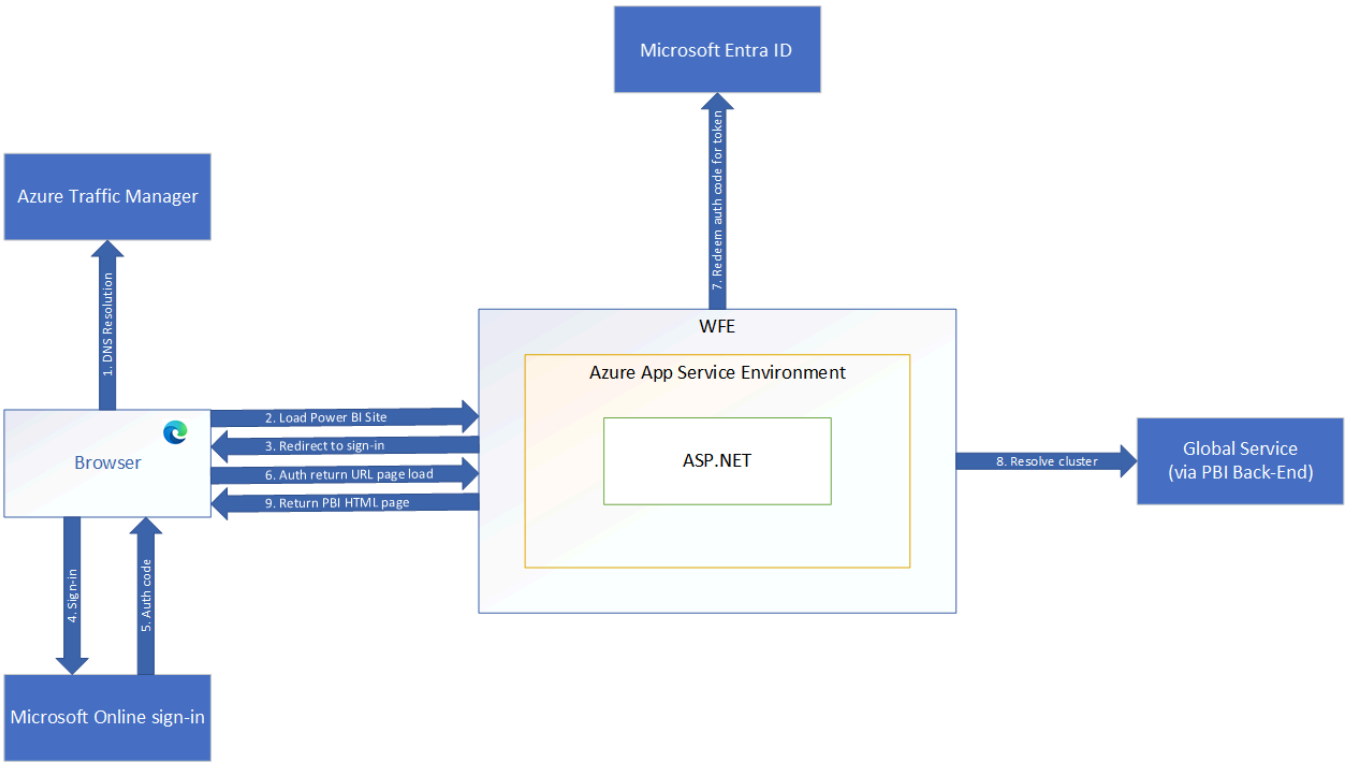
3. WFE then redirects the user to the Microsoft Online Services sign-in page.

4. After the user has been authenticated, the sign-in page redirects the user to the previously determined nearest Power BI service WFE cluster with an auth code.

5. The WFE cluster checks with Microsoft Entra ID to obtain a Microsoft Entra security token by using the auth code. When Microsoft Entra ID returns the successful authentication of the user and returns a Microsoft Entra security token, the WFE cluster consults the Power BI Global Service, which maintains a list of tenants and their Power BI back-end cluster locations and determines which Power BI back-end service cluster contains the user's tenant. The WFE cluster then returns an application page to the user's browser with the session, access, and routing information required for its operation.

6. Now, when the client's browser requires customer data, it will send requests to the back-end cluster address with the Microsoft Entra access token in the Authorization header. The Power BI back-end cluster reads the Microsoft Entra access token and validates the

signature to ensure that the identity for the request is valid. The Microsoft Entra access token will have an expiry date set according to Microsoft Entra policies, and to maintain the current session the Power BI Client in the user's browser will make periodic requests to renew the access token before it expires.



Additional resources

For more information on Power BI, see the following resources.

- Getting started with Power BI Desktop
- Power BI API reference
- On-premises data gateway
- Power BI national/regional clouds
- Microsoft Fabric security fundamentals
- Privacy, security, and responsible use for Copilot in Power BI

Deploying and Managing Power BI Premium Capacities

Article • 12/30/2024

We have retired the Power BI Premium whitepaper in favor of providing up-to-date information in separate articles. Use the following table to find content from the whitepaper.

Expand table

Articles	Description
• Basic concepts for designers in the Power BI service • Semantic models in the Power BI service • Semantic model modes in the Power BI service	Background information about Power BI service capacities, workspaces, dashboards, reports, workbooks, semantic models, and dataflows.
• What is Power BI Premium?	An overview of Power BI Premium, covering the basics of reserved capacities, supported workloads, unlimited content sharing, and other features.
• Managing Premium capacities • Configure and manage capacities in Power BI Premium • Configure workloads in a Premium capacity	Detailed information about configuring and managing capacities and workloads.
• Use the Fabric Capacity Metrics app	Monitoring with Fabric Capacity Metrics app, and interpreting the metrics you see in the app.
• Power BI Premium FAQ	Answers to questions around purchase and licensing, features, and common scenarios.

Feedback

Was this page helpful? Yes No

Provide product feedback | Ask the community