

Contents

Power BI guidance documentation

Overview

Overview

Optimization guide for Power BI

Concepts

Transform and shape data

The importance of query folding

Referencing Power Query queries

Disable Power Query background refresh

Best practices for dataflows

Data modeling

What is a star schema?

Reduce data model size

Create date tables

Auto date/time guidance

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

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
- Sample model
- Power BI reports
 - Separate reports from models
 - Extend visuals with report page tooltips
 - Use report page drillthrough
 - Tips to manage axes
 - Tips to control chart gridlines
 - Tips to optimize the use of labels
 - Tips to format and implement legends
 - Tips to optimize visual colors
 - Tips to sort and distribute data plots
 - Tips to work with shapes, images, and icons
- Power BI paginated reports
 - When to use paginated reports
 - Data retrieval guidance
 - Image use guidance
 - Use cascading parameters
 - Avoid blank pages when printing
 - Migrate SSRS reports to Power BI
- Admin and deployment
 - Tenant settings
 - On-premises data gateway sizing
 - Monitor report performance
 - Troubleshoot report performance
 - Deployment pipelines best practices
 - Access the Power BI activity log
- Migrate to Power BI
 - Migration overview
 - Prepare to migrate

- Gather requirements
- Plan deployment
- Conduct proof of concept
- Create content
- Deploy to Power BI
- Learn from customer Power BI migrations
- Adoption roadmap
 - Overview
 - Maturity levels
 - Data culture
 - Executive sponsorship
 - Content ownership
 - Content delivery scope
 - Center of Excellence
 - Governance
 - Mentoring and user enablement
 - Community of practice
 - User support
 - System oversight
 - Conclusion
- Implementation planning
 - Introduction
 - Usage scenarios
 - Usage scenarios overview
 - Personal BI
 - Team BI
 - Departmental BI
 - Enterprise BI
 - Managed self-service BI
 - Customizable managed self-service BI
 - Self-service data preparation
 - Prototyping and sharing

- Self-service content publishing
- On-premises reporting
- Microsoft's Center of Excellence (COE)
 - Microsoft's BI transformation
 - How Microsoft established a COE
 - BI solution architecture in the COE
- Whitepapers
 - Whitepapers overview
 - Power BI security whitepaper
 - Power BI enterprise deployment whitepaper
 - Power BI Premium deployment
 - Distribute Power BI externally using Azure Active Directory B2B

Guidance for Power BI

3/24/2022 • 2 minutes to read • [Edit Online](#)

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

3/24/2022 • 6 minutes to read • [Edit Online](#)

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 external-hosted or internal-hosted, and in Power BI they are referred to as *datasets*. It's important to understand your options, and to choose the appropriate dataset type for your solution. There are three dataset modes: Import, DirectQuery, and Composite. For more information, see [Datasets in the Power BI service](#), and [Dataset modes in the Power BI service](#).

For specific dataset mode guidance, see:

- [Data reduction techniques for Import modeling](#)
- [DirectQuery model guidance in Power BI Desktop](#)
- [Composite model guidance in Power BI Desktop](#)

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 dataset 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 back-end 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 datasets, 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 dataset 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 dataset, 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 dataset. Atop of that dataset, 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 [Managing Premium capacities](#). For guidance on how to optimize your capacity, see [Optimizing Premium capacities](#).

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 cluster 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 cluster 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](#).

Next steps

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

- [Power BI guidance](#)
- [Monitoring report performance](#)
- [Power BI adoption roadmap](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Query folding guidance in Power BI Desktop

3/24/2022 • 3 minutes to read • [Edit Online](#)

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 datasets.

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 transformation 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.

Next steps

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

- Power Query [Query folding](#) concept article
- [Incremental refresh for datasets](#)
- Questions? [Try asking the Power BI Community](#)

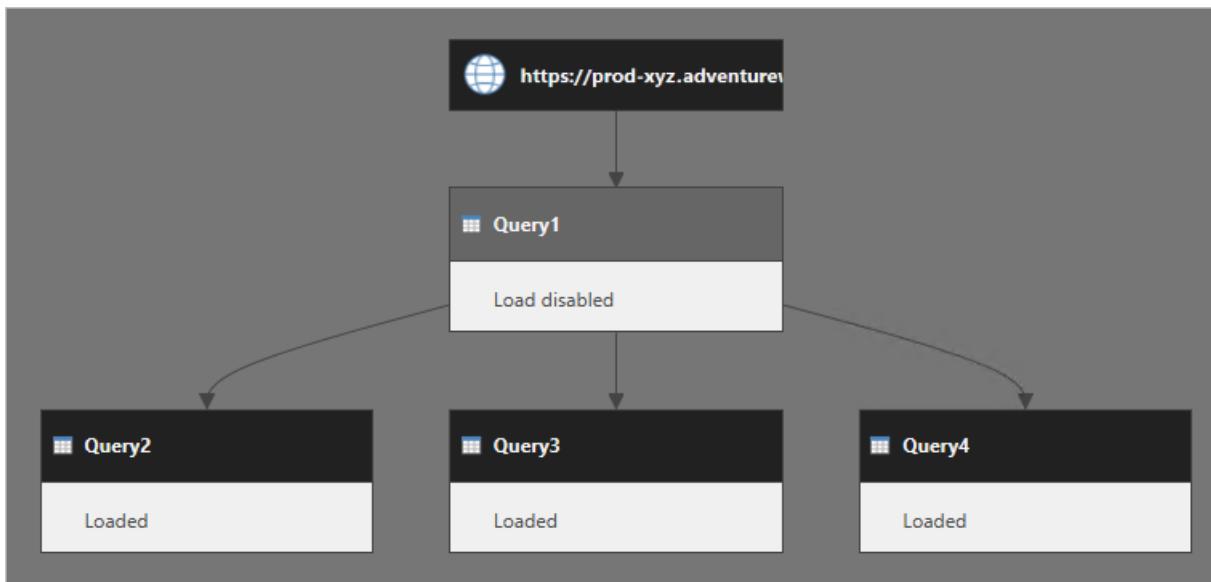
Referencing Power Query queries

3/24/2022 • 2 minutes to read • [Edit Online](#)

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.

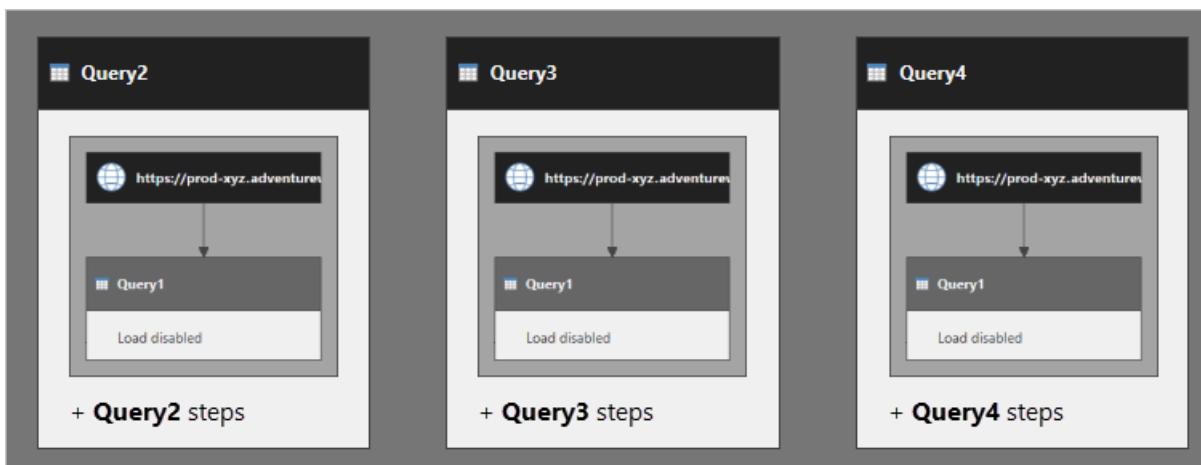
Let's be clear about what this means: *When a query references a second query, it's as though the steps in the second query are combined with, and run before, the steps in the first query.*

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 on 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'll themselves 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.

Next steps

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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Disable Power Query background refresh

3/24/2022 • 2 minutes to read • [Edit Online](#)

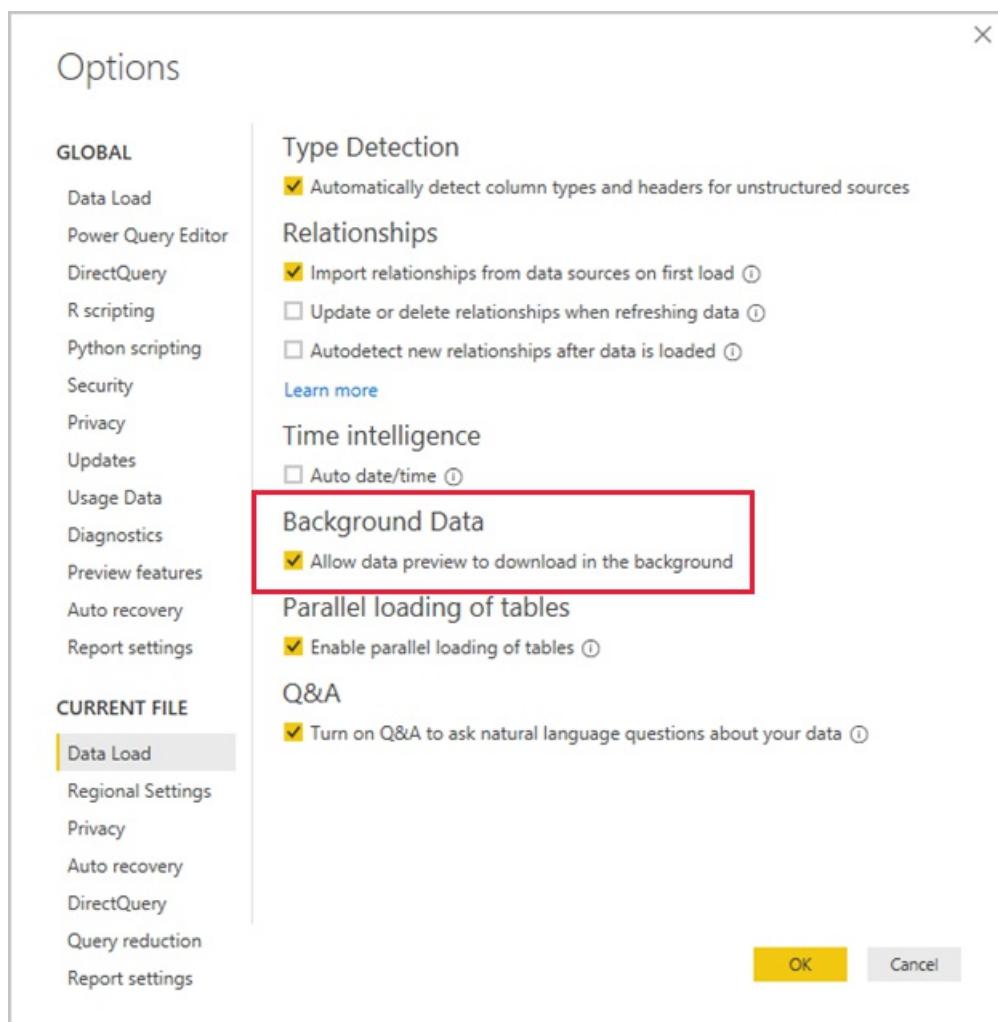
This article targets Import data modelers working with Power BI Desktop.

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.



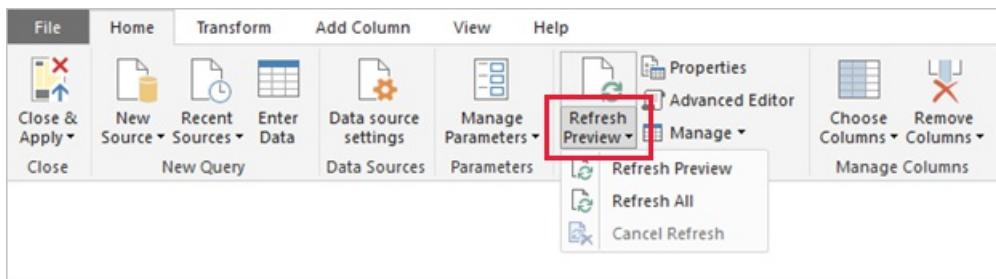
Enabling background refresh can result in preview data becoming out of date. If it occurs, the Power Query Editor will notify you with the following warning:



This preview may be up to 7 days old.

[Refresh](#)

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.



Next steps

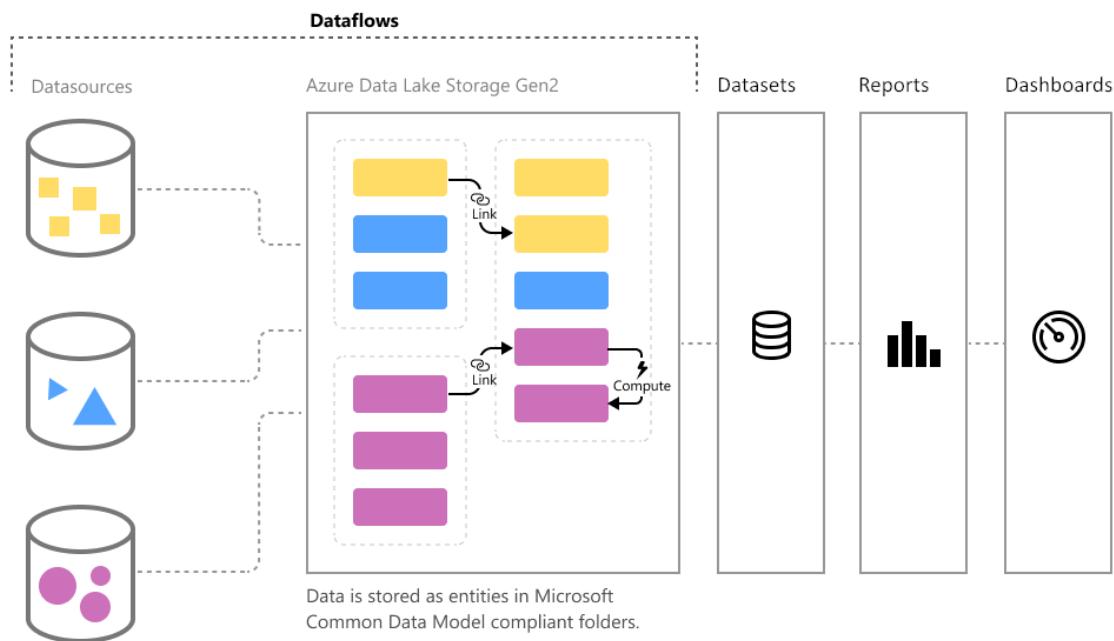
For more information related to this article, check out the following resources:

- [Power Query Documentation](#)
- Questions? [Try asking the Power BI Community](#)

Introduction to dataflows and self-service data prep

3/24/2022 • 2 minutes to read • [Edit Online](#)

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 clicks.



When to use dataflows

Dataflows are designed to support the following scenarios:

- Create reusable transformation logic that can be shared by many datasets and reports inside Power BI. Dataflows promote reusability of the underlying data elements, preventing the need to create separate connections with your cloud or on-premise data sources.
- Expose the data in your own Azure Data Lake Gen 2 storage, enabling you to connect other Azure services to the raw underlying data.
- Create a single source of the truth by forcing analysts to connect to the dataflows, rather than connecting to the underlying systems, providing you with control over which data is accessed, and how data is exposed to report creators. You can also map the data to industry standard definitions, enabling you to create tidy curated views, which can work with other services and products in the Power Platform.
- 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-premise sources.
- Prevent analysts from having direct access to the underlying data source. Since report creators can build on top of dataflows, it may be more convenient for you to allow access to underlying data sources only to a few individuals, and then provide access to the dataflows for analysts to build on top of. This approach

reduces the load to the underlying systems, and gives administrators finer control of when the systems get loaded from refreshes.

Once you've created a dataflow, you can use Power BI Desktop and the Power BI service to create datasets, reports, dashboards, and apps that leverage the Common Data Model to drive deep insights into your business activities. Dataflow refresh scheduling is managed directly from the workspace in which your dataflow was created, just like your datasets.

Next steps

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](#)

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

- [Common Data Model - overview](#)

Understand star schema and the importance for Power BI

3/24/2022 • 18 minutes to read • [Edit Online](#)

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

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

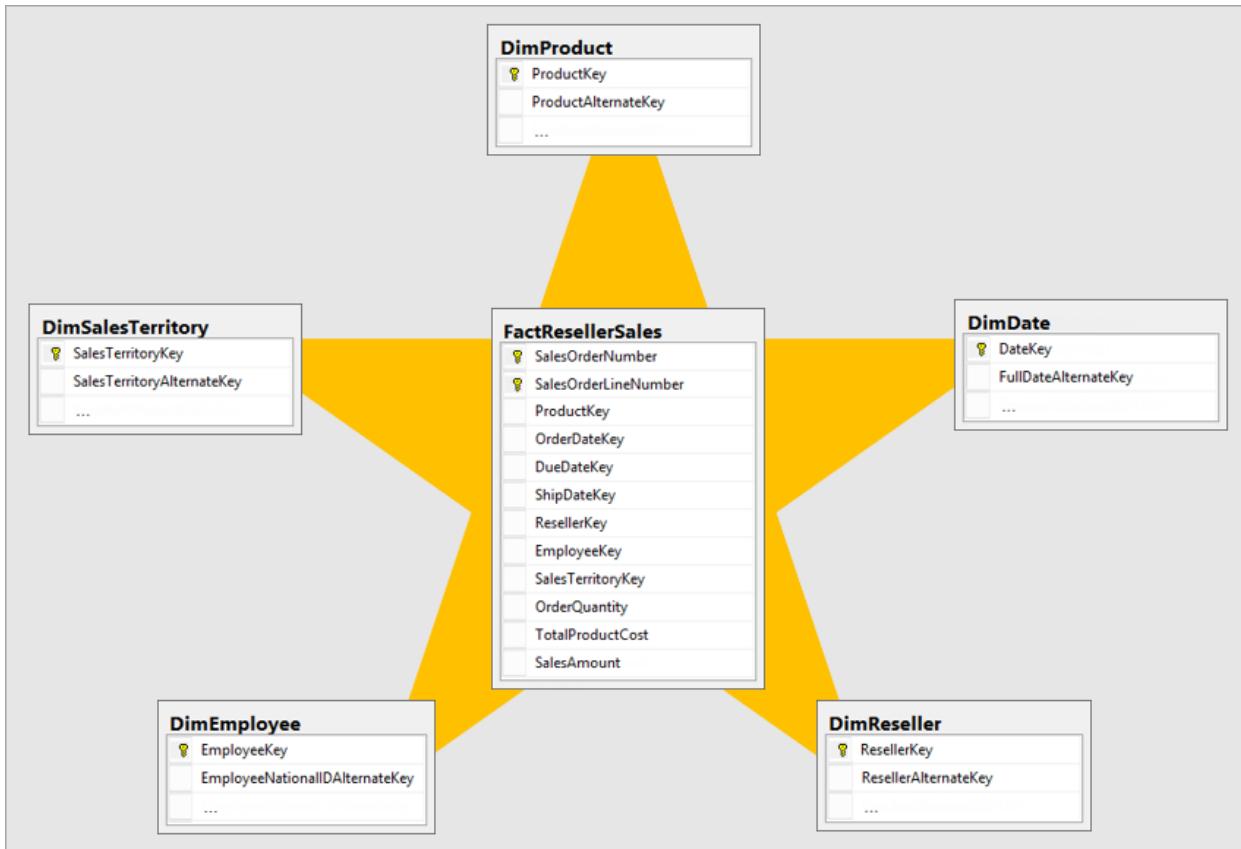
Star schema overview

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 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 descriptive columns.

Fact tables store observations or events, and can be sales orders, stock balances, exchange rates, temperatures, etc. 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 very 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 repetitive data. Consider a table of products that has a unique key value column, like the product key, and additional columns describing product characteristics, including 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.

	SalesOrderNumber	OrderDate	ProductKey	ResellerKey	SalesAmount
1	SO69561	2020-05-31	594	546	226.00
2	SO69560	2020-05-30	513	100	218.45
3	SO69560	2020-05-30	594	100	113.00
4	SO69539	2020-05-28	243	529	858.90
5	SO69539	2020-05-28	378	529	1466.01
6	SO69541	2020-05-28	594	661	113.00
7	SO69542	2020-05-28	243	317	1717.80
8	SO69544	2020-05-28	243	666	3435.60
9	SO69545	2020-05-28	378	436	5864.04
10	SO69532	2020-05-27	594	312	113.00
11	SO69532	2020-05-27	513	312	436.90
12	SO69533	2020-05-27	594	476	226.00

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.

	SalesOrderNumber	OrderDate	ProductKey	Product	Category	Color	Size	ResellerKey	SalesAmount
1	SO69561	2020-05-31	594	Mountain-500 Silver, 48	Bikes	Silver	48	546	226.00
2	SO69560	2020-05-30	513	ML Mountain Frame-W - Silver, 46	Components	Silver	46	100	218.45
3	SO69560	2020-05-30	594	Mountain-500 Silver, 48	Bikes	Silver	48	100	113.00
4	SO69539	2020-05-28	243	HL Road Frame - Red, 44	Components	Red	44	529	858.90
5	SO69539	2020-05-28	378	Road-250 Black, 52	Bikes	Black	52	529	1466.01
6	SO69541	2020-05-28	594	Mountain-500 Silver, 48	Bikes	Silver	48	661	113.00
7	SO69542	2020-05-28	243	HL Road Frame - Red, 44	Components	Red	44	317	1717.80
8	SO69544	2020-05-28	243	HL Road Frame - Red, 44	Components	Red	44	666	3435.60
9	SO69545	2020-05-28	378	Road-250 Black, 52	Bikes	Black	52	436	5864.04
10	SO69532	2020-05-27	594	Mountain-500 Silver, 48	Bikes	Silver	48	312	113.00
11	SO69532	2020-05-27	513	ML Mountain Frame-W - Silver, 46	Components	Silver	46	312	436.90
12	SO69533	2020-05-27	594	Mountain-500 Silver, 48	Bikes	Silver	48	476	226.00

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 data models with tables that represent normalized fact and dimension data. However, there's one exception where a [snowflake dimension](#) should be denormalized to produce a single model table.

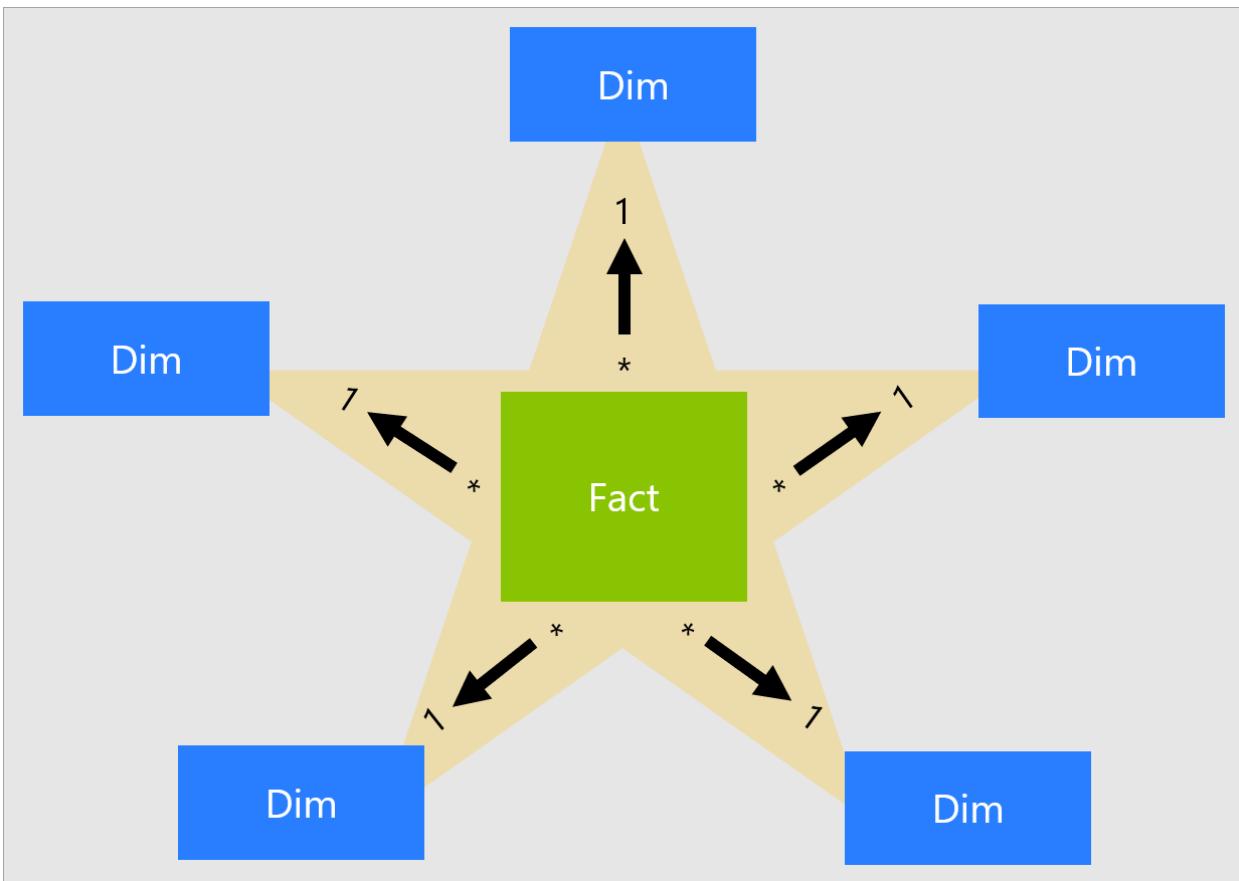
Star schema relevance to Power BI models

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 is sent to the Power BI model (which the Power BI service calls a dataset). These queries are used to 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 support *filtering* and *grouping*
- Fact tables support *summarization*

There's no table property that modelers set to configure 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-type table while the "many" side is always a fact-type table. For more information about relationships, see [Model relationships in Power BI Desktop](#).



A well-structured model design should include tables that are either dimension-type tables or fact-type tables. Avoid mixing the two types together for a single table. We also recommend that you should strive to deliver the right number of tables with the right relationships in place. It's also important that fact-type 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 additional concepts related to star schema design that can be applied to a Power BI 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 model, a **measure** has a different—but similar—definition. It's a formula written in [Data Analysis Expressions \(DAX\)](#) that achieves summarization. Measure expressions often leverage DAX aggregation functions like SUM, MIN, MAX, AVERAGE, etc. 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 the [DAX Basics in Power BI Desktop](#) article.

It's important to understand that Power BI models support a second method for achieving summarization. Any column—and typically numeric columns—can be summarized by a report visual or Q&A. These columns are referred to as *implicit measures*. They offer a convenience for you as a model developer, as in many instances you do not need to create measures. For example, the Adventure Works reseller sales **Sales Amount** column could be summarized in numerous ways (sum, count, average, median, min, max, etc.), without the need to create a measure for each possible aggregation type.

The screenshot shows the 'Fields' pane in Power BI. A tree view shows the 'Reseller Sales' table expanded. Under it, five columns are listed with their respective DAX aggregation functions: 'Order Quantity' (Σ), 'Sales' (with a red arrow pointing to it labeled 'Measure'), 'Sales Amount' (with a red arrow pointing to it labeled 'Summarizable column'), 'Total Product Cost' (Σ), and 'Unit Price' (Σ). The 'Sales' and 'Sales Amount' columns are highlighted with red arrows and labels.

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

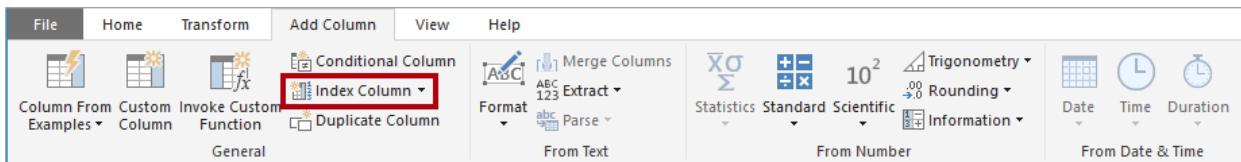
- When you know your report authors will query the model by using [Multidimensional Expressions \(MDX\)](#), the model must include *explicit measures*. Explicit measures are defined by using DAX. This design approach is highly relevant when a Power BI dataset is queried by using MDX, because MDX can't achieve summarization of column values. Notably, MDX will be used when performing [Analyze in Excel](#), because PivotTables issue MDX queries.
- When you know your report authors will create Power BI paginated reports using the MDX query designer, the 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 need to ensure that your report authors can only 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, average, etc. 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 **Fields** 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 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-type table in your 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 easily achieve this requirement by creating a [Power Query index column](#).

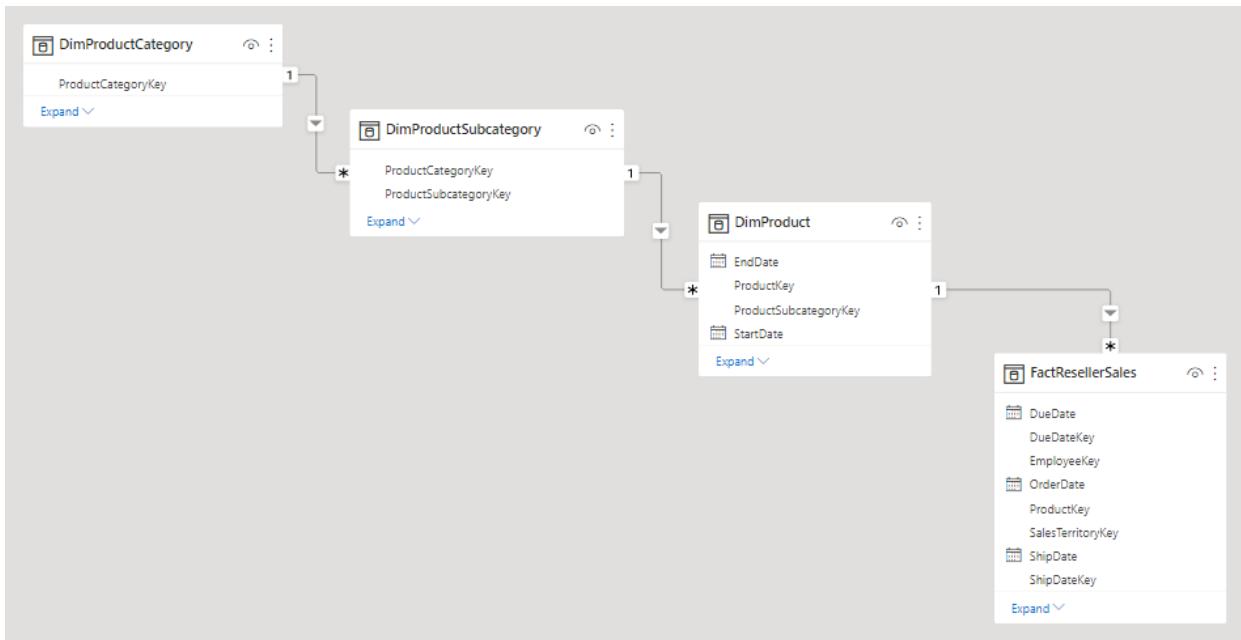


You must merge this query with the "many"-side query so that you can add the index column to it also. When you load these queries to the model, you can then create a one-to-many relationship between the model tables.

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. Categories are assigned to subcategories, and products are in turn assigned to subcategories. 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.

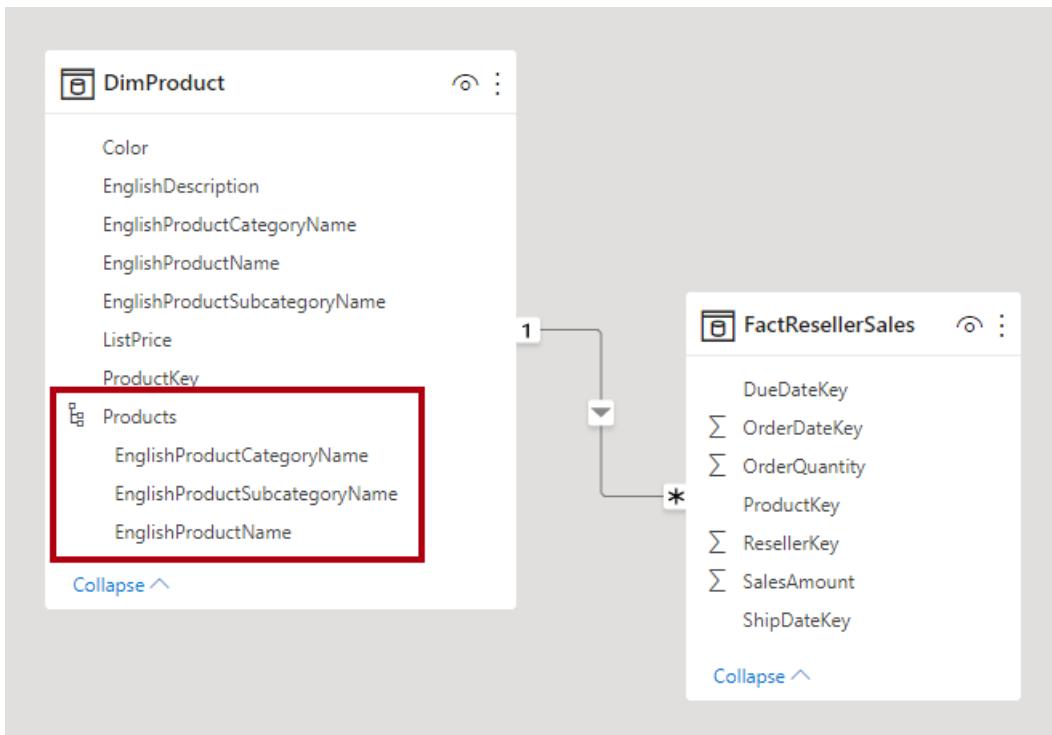


In Power BI Desktop, you can choose to mimic a snowflake dimension design (perhaps because your source data does) or integrate (denormalize) the source tables into a single model table. Generally, the benefits of a single model table outweigh the benefits of multiple model tables. The most optimal decision 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 will need to be traversed, which will likely be less efficient than filters applied to a single table.
- The **Fields** pane presents more model tables to report authors, which can result in a less intuitive experience, especially when snowflake dimension tables contain just one or two columns.
- It's not possible to create a hierarchy that spans the tables.

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 very large dimension tables.



Slowly changing dimensions

A **slowly changing dimension** (SCD) is one that appropriately manages change of dimension members over time. It applies when business entity values change over time, and in an ad hoc manner. A good example of a *slowly* changing dimension is a customer dimension, specifically its contact detail columns like email address and phone number. In contrast, some dimensions are considered to be *rapidly* changing when a dimension attribute changes often, like a stock's market price. 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-type 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-type 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, then 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 salespeople 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

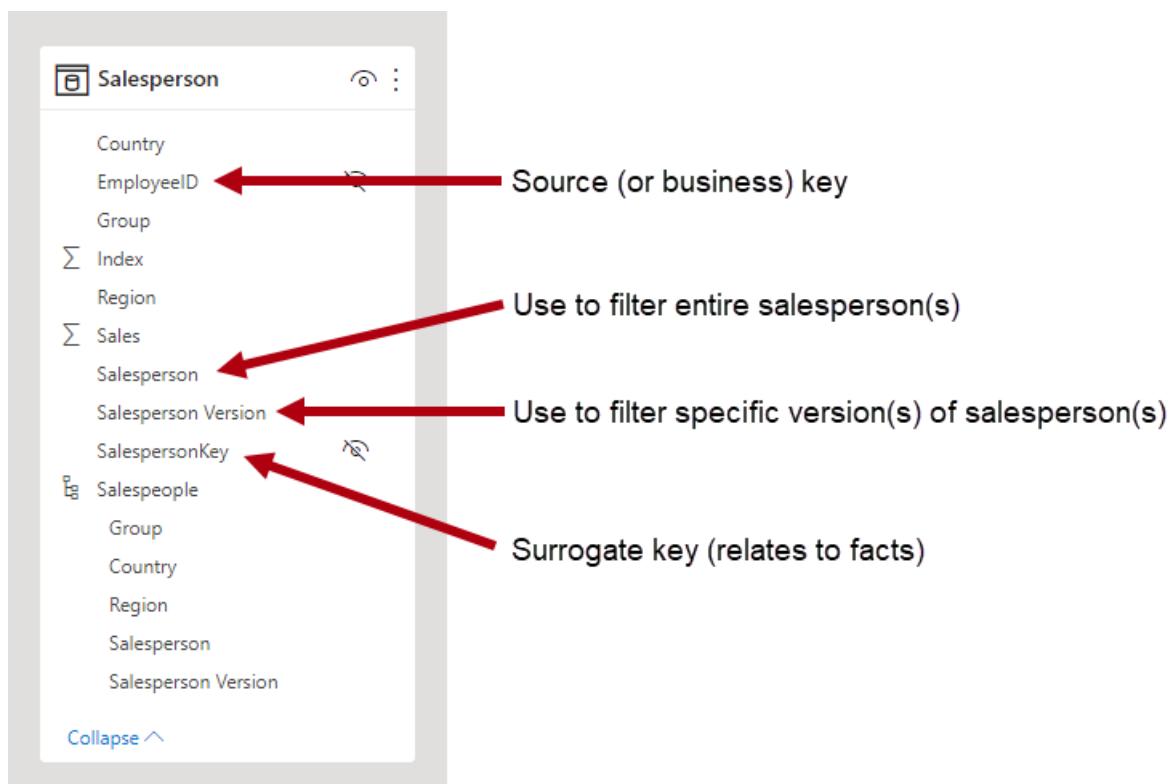
time validity. Current versions may define an empty end date (or 12/31/9999), which indicates that the row is the current version. The table must also define a surrogate key because the business key (in this instance, employee ID) won't be unique.

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 model using Power Query can't produce this result. It can, however, load data from a pre-loaded SCD Type 2 dimension table.

The Power BI 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 model dimension-type 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 "Michael Blythe (12/15/2008-06/26/2019)" or "Michael Blythe (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 also a good design practice to include a hierarchy that allows visuals to drill down to the version level.



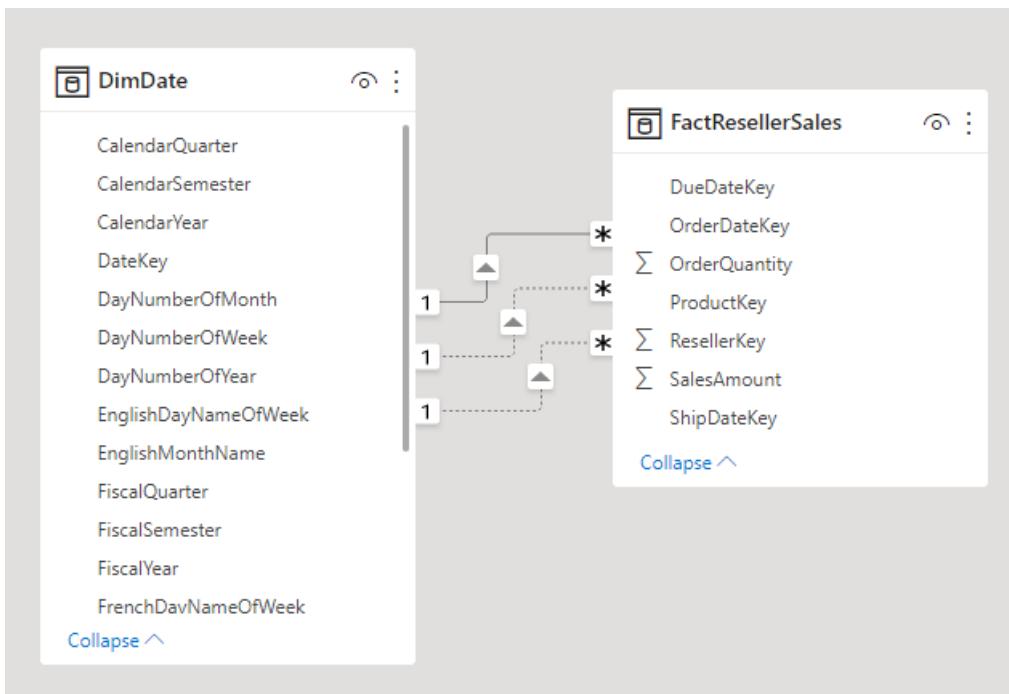
Group	Country	Region	Salesperson	Salesperson Version	Sales
North America	United States	Northeast	David Campbell	David Campbell (12/15/2008-06/26/2019)	\$52,307.11
				David Campbell (06/27/2019-Current)	\$119,807.56
				Total	\$172,114.67
				Pamela Ansman-Wolfe	Pamela Ansman-Wolfe (03/03/2007-Current)
				Total	\$222,087.01
			Tete Mensa-Annan	Tete Mensa-Annan (01/09/2005-Current)	\$23,098.4
				Total	\$23,098.4
				Total	\$417,300.08
			Total		\$417,300.08
			Total		\$417,300.08
Total					\$417,300.08

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.

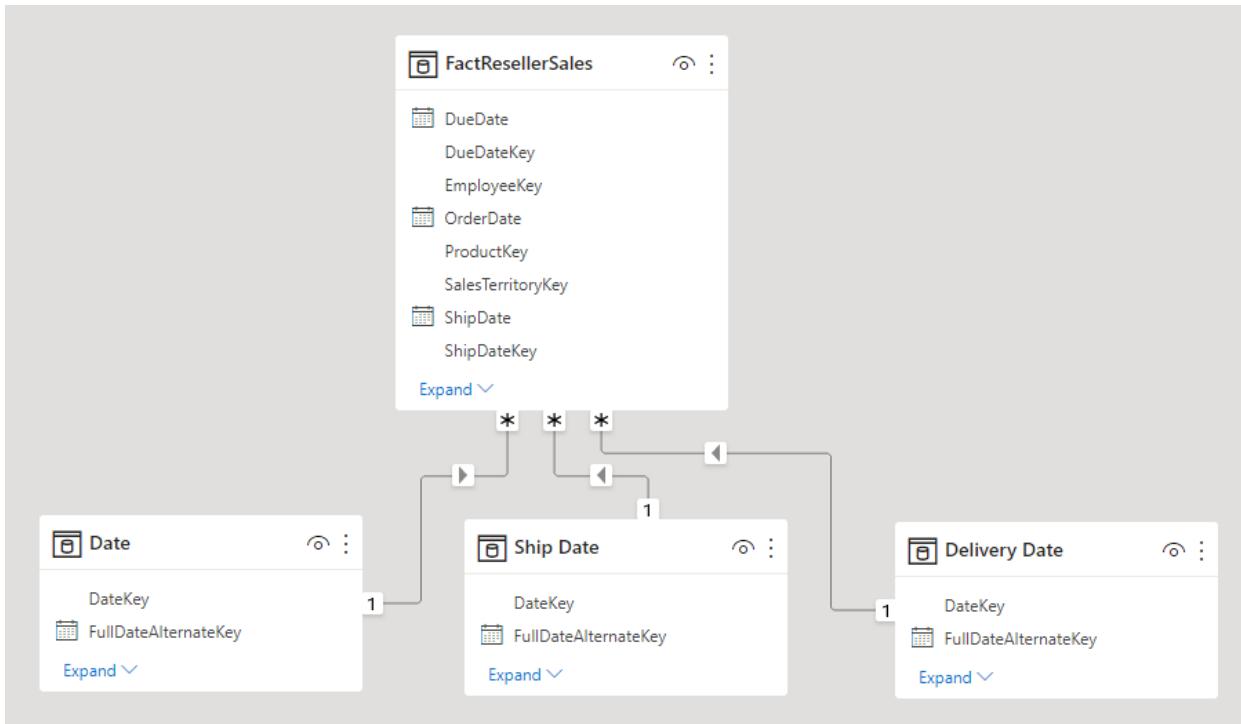
In a Power BI model, this design can be imitated by creating multiple relationships between two tables. In the Adventure Works example, the date and reseller sales tables would have three relationships. While this design is possible, it's important to understand that there can only be one active relationship between two Power BI model tables. All remaining relationships must be set to inactive. Having a single active relationship means there is a default filter propagation from date to reseller sales. In this instance, the active relationship is set to the most common filter that is 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 **Fields** pane clutter, with 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-type table for each role-playing instance. You typically create the additional dimension tables as [calculated tables](#), using DAX. Using calculated tables, 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, however, with this design approach is that there will be duplication of the date dimension table resulting in an increased model storage size. As dimension-type tables typically store fewer rows relative to fact-type tables, it is rarely a concern.

Observe the following good design practices when you create model dimension-type 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**, etc.
- When relevant, ensure that table descriptions provide feedback to report authors (through **Fields** pane tooltips) about how filter propagation is configured. This clarity is important when the model contains a generically named table, like **Date**, which is used to filter many fact-type 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 (gender, age group, etc.).

The design objective of a junk dimension is to consolidate many "small" dimensions into a single dimension to both reduce the model storage size and also reduce **Fields** 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. The surrogate key provides a unique reference to each row in the table. 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).

Order Status	ResellerSalesJunkKey	Order Status	Delivery Status
Quote	1	1 Quote	Not Delivered
Ordered	2	2 Quote	Delivered
Cancelled	3	3 Ordered	Not Delivered
	4	4 Ordered	Delivered
Not Delivered	5	5 Cancelled	Not Delivered
Delivered	6	6 Cancelled	Delivered

You load this query to the model as a dimension-type 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 is required for filtering. At Adventure Works, the reseller sales order number is a good example. In this case, it doesn't make good model design sense to create an independent table consisting of just this one column, because it would increase the model storage size and result in **Fields** pane clutter.

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

The screenshot shows the 'Fields' pane in Power BI. Under the 'Reseller Sales' fact table, the 'Order Number' column is highlighted with a red arrow. The other columns listed are Order Quantity, Sales, Sales Amount, Total Product Cost, and Unit Price.

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

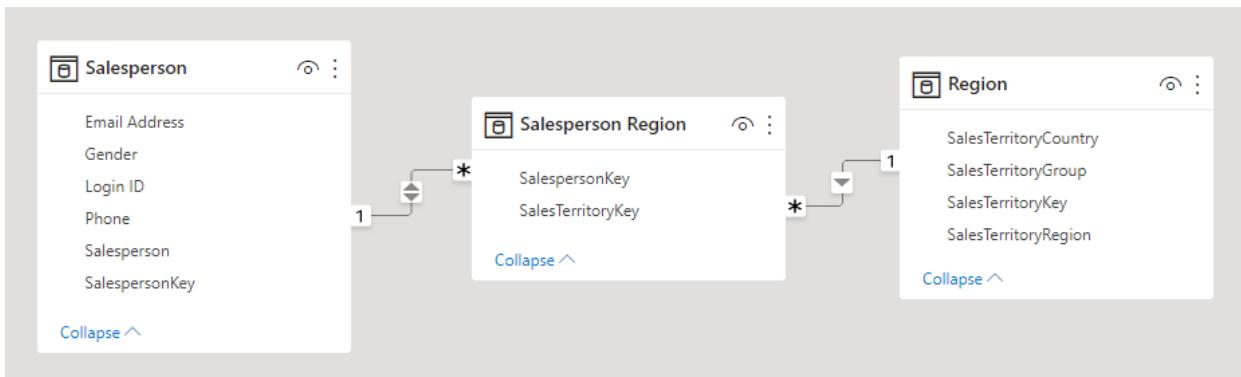
Factless fact tables

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 logged into 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 have logged in.

A more compelling use of a factless fact table is to store relationships between dimensions, and it's the Power BI model design approach we recommend 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 \(Relate two dimension-type tables\)](#).

Next steps

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

- [Dimensional modeling Wikipedia article](#)
- [Create and manage relationships in Power BI Desktop](#)
- [One-to-one relationship guidance](#)
- [Many-to-many relationship guidance](#)
- [Bi-directional relationship guidance](#)
- [Active vs inactive relationship guidance](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Data reduction techniques for Import modeling

3/24/2022 • 7 minutes to read • [Edit Online](#)

This article targets Power BI Desktop data modelers developing Import models. It describes different techniques to help reduce the data loaded into Import models.

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

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

- Larger model sizes may not be supported by your capacity. Shared capacity can host models up to 1 GB in size, while Premium capacities can host models up to 13 GB in size. For further information, read the [Power BI Premium support for large datasets](#) article.
- Smaller model sizes reduce contention for capacity resources, in particular memory. It allows more models to be concurrently loaded for longer periods of time, resulting in lower eviction rates. For more information, see [Managing Premium capacities](#).
- Smaller models achieve faster data refresh, resulting in lower latency reporting, higher dataset refresh throughput, and less pressure on source system and capacity resources.
- Smaller table row counts can result in faster calculation evaluations, which can deliver better overall query performance.

There are eight different data reduction techniques covered in this article. These techniques include:

- [Remove unnecessary columns](#)
- [Remove unnecessary rows](#)
- [Group by and summarize](#)
- [Optimize column data types](#)
- [Preference for custom columns](#)
- [Disable Power Query query load](#)
- [Disable auto date/time](#)
- [Switch to Mixed mode](#)

Remove unnecessary columns

Model table columns serve two main purposes:

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

Columns that don't serve these purposes can probably be removed. Removing columns is referred to as *vertical filtering*.

We recommend that you design models with exactly the right number of columns based on the known reporting requirements. Your requirements may 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

Model tables should be loaded with as few rows as possible. It can be achieved by loading filtered rowsets into model tables for two different reasons: to filter by entity or by time. Removing rows is referred to as *horizontal filtering*.

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 will result in many smaller models, and it can also eliminate the need to define row-level security (but will require granting specific dataset permissions in the Power BI service, and creating "duplicate" reports that connect to each dataset). You can leverage the use of Power Query parameters and Power BI Template files to simplify management and publication. For further information, read the [Deep Dive into Query Parameters and Power BI Templates](#) blog entry.

Filtering by time involves limiting the amount of *data history* loaded into fact-type tables (and limiting the date rows loaded into the model date tables). We suggest you don't automatically load all available history, unless it is a known reporting requirement. It is helpful to understand that time-based Power Query filters can be parameterized, and even set to use relative time periods (relative to the refresh date, for example, the past five years). Also, bear in mind that retrospective changes to time filters will not break reports; it will just result in less (or more) data history available in reports.

Group by and summarize

Perhaps the most effective technique to reduce a model size is to load pre-summarized data. This technique can be used to raise the grain of fact-type tables. There is a distinct trade-off, however, resulting in loss of detail.

For example, a source sales fact table stores one row per order line. Significant data reduction could be achieved by summarizing all sales metrics, grouping by date, customer, and product. Consider, then, that an even more significant data reduction could be achieved by grouping by date *at month level*. It could achieve a possible 99% reduction in model size, but reporting at day level—or individual order level—is no longer possible. Deciding to summarize fact-type data always involves tradeoffs. Tradeoff could be mitigated by a Mixed model design, and this option is described in the [Switch to Mixed mode](#) technique.

Optimize column data types

The VertiPaq storage engine uses separate data structures for each column. By design, these data structures achieve the highest optimizations for numeric column data, which use value encoding. Text and other non-numeric data, however, uses hash encoding. It requires the storage engine to assign a numeric identifier to each unique text value contained in the column. It is the numeric identifier, then, that is then 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 may be consistently prefixed by a text value (e.g. "SO123456"). The prefix could be removed, and the order number value converted to whole number. For large tables, it 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 will help to minimize the 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 data structures are stored slightly differently, and typically achieve less efficient compression. Also, they are built once all Power Query tables are loaded, which can result in extended data refresh times. It is therefore less efficient to add table columns as *calculated columns* than Power Query

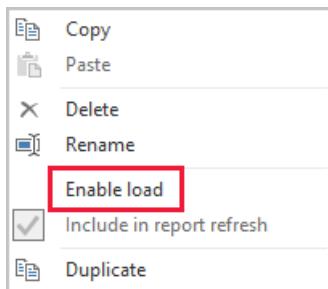
computed columns (defined in M).

Preference should be 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 may be the better choice. It can be the case when the formula involves evaluating measures, or it requires specific modeling functionality only supported in DAX functions. For information on one such example, refer to the [Understanding functions for parent-child hierarchies in DAX](#) article.

Disable Power Query query load

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



Disable auto date/time

Power BI Desktop includes an option called *Auto date/time*. When enabled, it creates a hidden auto date/time table for date columns to support report authors when configuring filters, grouping, and drill-down actions for calendar time periods. The hidden tables are in fact calculated tables that will increase the size of the model. For guidance about using this option, refer to the [Auto date/time guidance in Power BI Desktop](#) article.

Switch to Mixed mode

In Power BI Desktop, a Mixed mode design produces a Composite model. Essentially, it allows you to determine storage mode *for each table*. Therefore, each table can have its Storage Mode property set as Import or DirectQuery (Dual is another option).

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

There are, however, many security and performance implications related to Composite models. For further information, read the [Use composite models in Power BI Desktop](#) article.

Next steps

For more information about Power BI Import model design, see the following articles:

- [Use composite models in Power BI Desktop](#)
- [Storage mode in Power BI Desktop](#)
- Questions? [Try asking the Power BI Community](#)

Create date tables in Power BI Desktop

3/24/2022 • 4 minutes to read • [Edit Online](#)

This article targets you as a data modeler working with Power BI Desktop. It describes good design practices for creating date tables in your data models.

To work with Data Analysis Expressions (DAX) [time intelligence functions](#), there's a prerequisite model requirement: You must have at least one *date table* in your model. A date table is a 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:

- The Auto date/time option
- Power Query to connect to a date dimension table
- Power Query to generate a date table
- DAX to generate a date table
- DAX to clone an existing date table

TIP

A date table is perhaps the most consistent feature you'll add to any of your models. What's more, 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 modelers in your organization. So, whenever someone develops a new 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. Using this option can also be convenient 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 will have a date dimension table. This way, your model leverages 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 leverage the DAX time intelligence capabilities.

Generate with Power Query

You can generate a date table 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 functions. 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—it's a requirement for a marked date table. What's more, you don't need to manage extending the table to future years: When a data refresh completes, it triggers the recalculation of the table. A recalculation will automatically extend the table's date range when dates for a new year are loaded into the model.

Clone with DAX

When your model already has a date table and you need an additional date table, you can easily clone the existing date table. It's the case when date is a [role playing dimension](#). You can clone a table by creating a calculated table. The calculated table expression is simply the name of the existing date table.

Next steps

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 Power BI Community](#)

- Suggestions? Contribute ideas to improve Power BI

Auto date/time guidance in Power BI Desktop

3/24/2022 • 3 minutes to read • [Edit Online](#)

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. 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, etc.). The values are not fully self-describing, and in some report designs may 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 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-type 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 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. 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 may allow you to create a single date table that propagates to all fact-type tables, possibly resulting 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.

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

Next steps

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

- [Create 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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

One-to-one relationship guidance

3/24/2022 • 7 minutes to read • [Edit Online](#)

This article targets you as a data modeler working 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-type 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-type 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-type tables with a one-to-one relationship. It's because both fact-type tables would need to have the same dimensionality and granularity. Also, each fact-type table would need unique columns to allow the model relationship to be created.

Degenerate dimensions

When columns from a fact-type table are used for filtering or grouping, you can consider making them available in a separate table. This way, you separate columns used for filter 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 **Fields** pane experience to your report authors

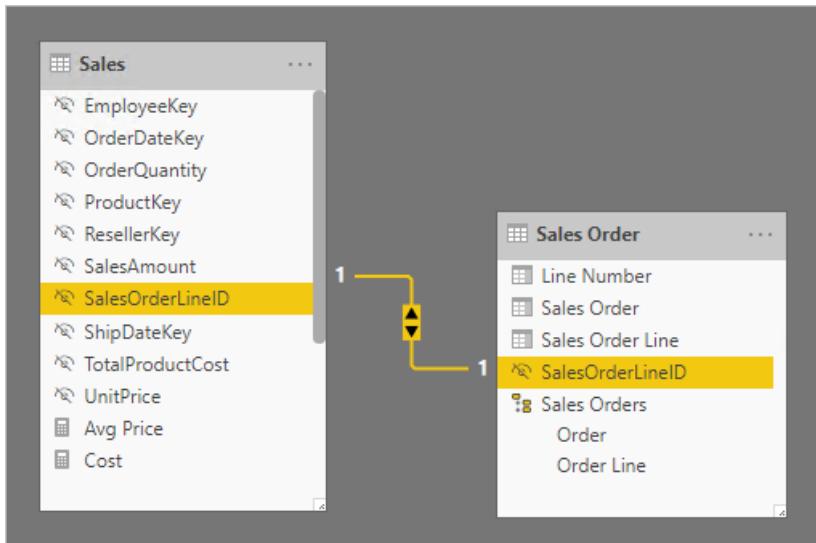
Consider a source sales table that stores sales order details in two columns.

	OrderNumber	OrderLineNumber	OrderDate	DueDate	ShipDate	ProductID	CustomerID
1	43659	1	2018-12-29	2019-01-10	2019-01-05	349	676
2	43659	2	2018-12-29	2019-01-10	2019-01-05	350	676
3	43659	3	2018-12-29	2019-01-10	2019-01-05	351	676
4	43660	1	2018-12-29	2019-01-10	2019-01-05	326	117
5	43660	2	2018-12-29	2019-01-10	2019-01-05	319	117

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

In the following model diagram, 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 **SalesOrderLineID**. (The key value is calculated by multiplying the order number by 1000, and then adding the order line number.)



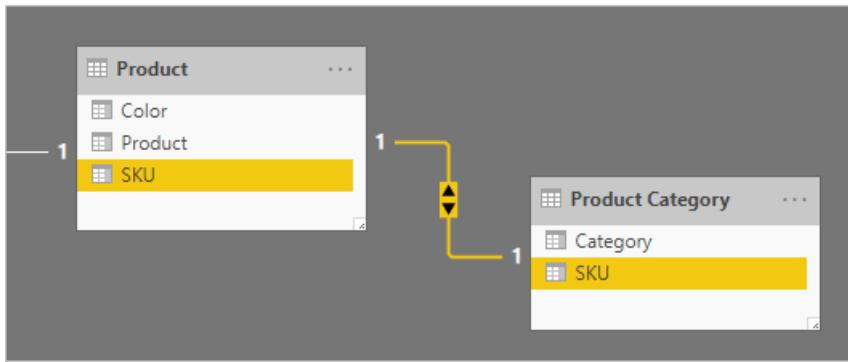
The **Sales Order** table provides a rich experience for report authors with three columns: **Sales Order**, **Sales Order Line**, and **Line Number**. It also includes a hierarchy. These table resources support report designs that need to filter, group by, or drill down through orders and order lines.

As 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 **SalesOrderLineID** column.

Row data spans across tables

Consider an example involving two one-to-one related dimension-type tables: **Product**, and **Product Category**. Each table represents imported data and has a **SKU** (Stock-Keeping Unit) column containing unique values.

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

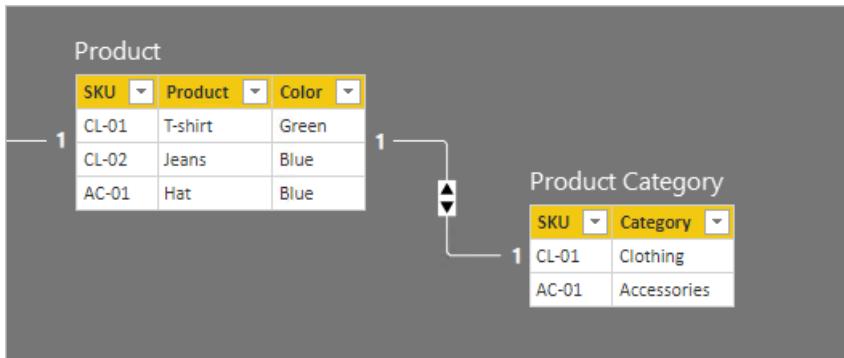


The first table is named **Product**, and it contains three columns: **Color**, **Product**, and **SKU**. The second table is named **Product Category**, and it contains two columns: **Category**, and **SKU**. A one-to-one relationship relates the two **SKU** columns. The relationship filters in both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

To help describe how the relationship filter propagation works, the model diagram has been modified to reveal the table rows. All examples in this article are based on this data.

NOTE

It's not possible to display table rows in the Power BI Desktop model diagram. It's done in this article to support the discussion with clear examples.

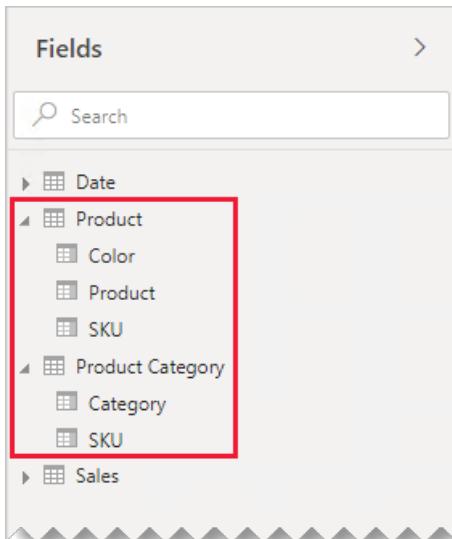


The row details for the two tables are described in the following bulleted list:

- 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 **Fields** pane, report authors will 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.

Product List			
SKU	Product	Color	Category
AC-01	Hat	Blue	Accessories
CL-01	T-shirt	Green	Clothing
CL-02	Jeans	Blue	

Notice that the **Category** value for product SKU CL-02 is BLANK. It's because there's no 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. It's because this design can:

- Contribute to **Fields** 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 \(Relationship evaluation\)](#).

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. It's done by merging the Power Query queries.

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

- 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. Configure 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.
- 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 **Fields** pane.

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

The screenshot shows the 'Fields' pane in Power BI. At the top, there are navigation arrows and a search bar. Below the search bar, there is a tree view of data sources. The 'Product' table is expanded, showing its four columns: 'Category', 'Color', 'Product', and 'SKU'. The 'Date' and 'Sales' tables are collapsed, indicated by a minus sign icon. A red box highlights the 'Product' table in the tree view.

3. **Replace missing values:** If the second query has unmatched rows, NULLs will appear in the columns introduced from it. When appropriate, consider replacing NULLs 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 table visual, notice that the category for product SKU CL-02 now reads *[Undefined]*. In the query, null categories were replaced with this token text value.

Product List			
SKU	Product	Color	Category
AC-01	Hat	Blue	Accessories
CL-01	T-shirt	Green	Clothing
CL-02	Jeans	Blue	[Undefined]

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.

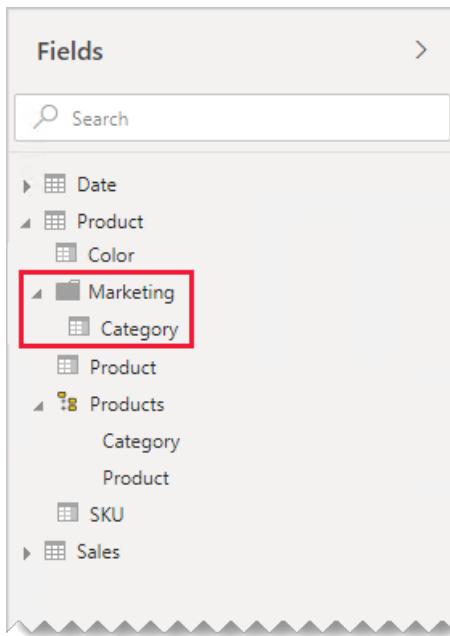
In our example, report authors now can use a hierarchy that has two levels: **Category** and **Product**.

The screenshot shows the 'Fields' pane in Power BI. It lists various fields under different display folders. A red box highlights the 'Products' folder, which contains the 'Category' and 'Product' fields. Other visible fields include Date, Product, Color, SKU, and Sales.

Display Folder	Field
Date	Date
Product	Category
Product	Color
Product	Product
Products	Category
Products	Product
SKU	SKU
Sales	Sales

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 sources. 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 will be 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.

Product List			
SKU	Product	Color	Category
AC-01	Hat	Blue	Accessories
CL-01	T-shirt	Green	Clothing

The table displays two rows only. Product SKU CL-02 is missing because there's no matching row in the **Product Category** table.

Next steps

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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Many-to-many relationship guidance

3/24/2022 • 14 minutes to read • [Edit Online](#)

This article targets you as a data modeler working 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, in fact, three many-to-many scenarios. They can occur when you're required to:

- [Relate two dimension-type tables](#)
- [Relate two fact-type tables](#)
- [Relate higher grain fact-type tables](#), when the fact-type table stores rows at a higher grain than the dimension-type table rows

NOTE

Power BI now natively supports many-to-many relationships. For more information, see [Apply many-many relationships in Power BI Desktop](#).

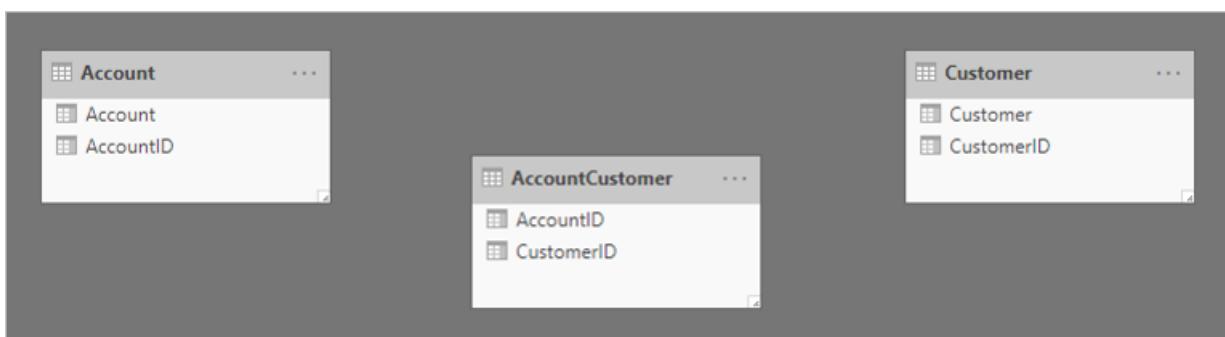
Relate many-to-many dimensions

Let's consider the first many-to-many scenario type with an example. The classic scenario relates two entities: 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-type table stores accounts, and another dimension-type table stores customers. As is characteristic of dimension-type tables, there's an 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, its purpose is to store one row for each customer-account association.

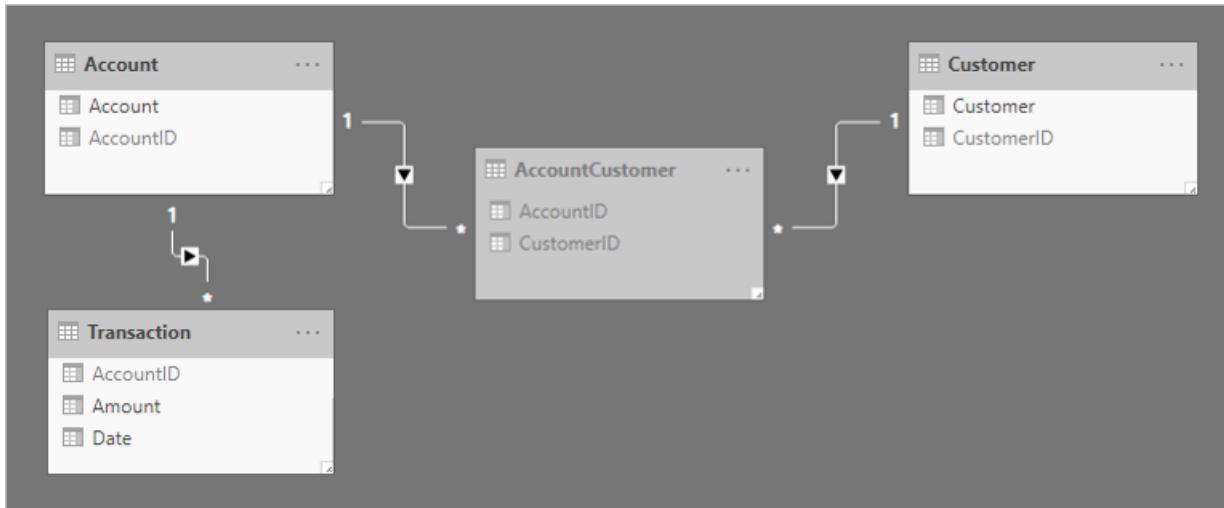
Interestingly, when this table only contains ID columns, it's called a *factless fact table*.

Here's a simplistic model diagram of the three 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.

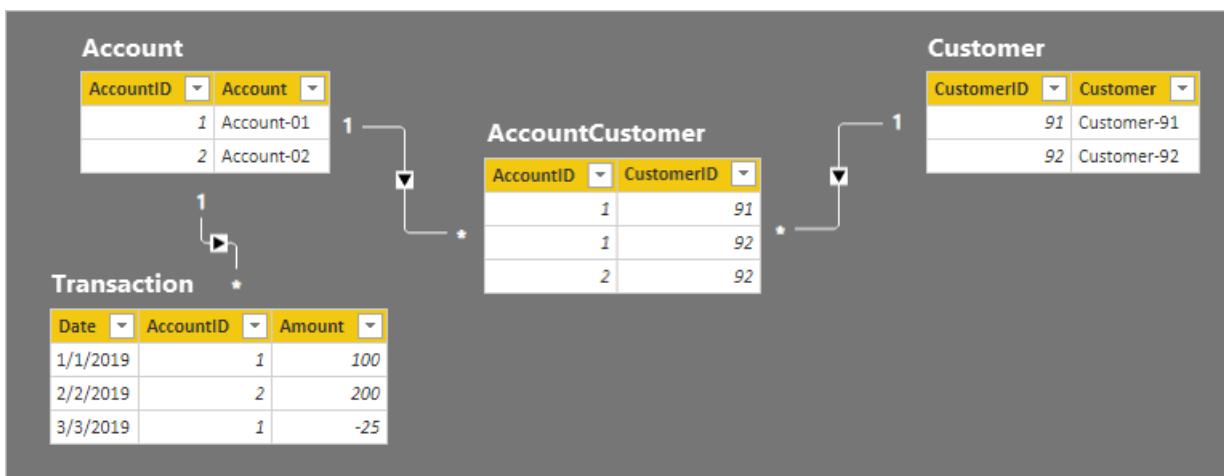
Two one-to-many relationships are added to relate the tables. Here's an updated model diagram of the related tables. A fact-type table named **Transaction** has been added. It records account transactions. The bridging table and all ID columns have been hidden.



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

NOTE

It's not possible to display table rows in the Power BI Desktop model diagram. It's done in this article to support the discussion with clear examples.



The row details for the four tables are described 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.

Below are two visuals that summarize the **Amount** column from 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*.

Account Balance		Customer Balance	
Account	Amount	Customer	Amount
Account-01	75	Customer-91	275
Account-02	200	Customer-92	275
Total	275	Total	275

The first visual is titled **Account Balance**, and it 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

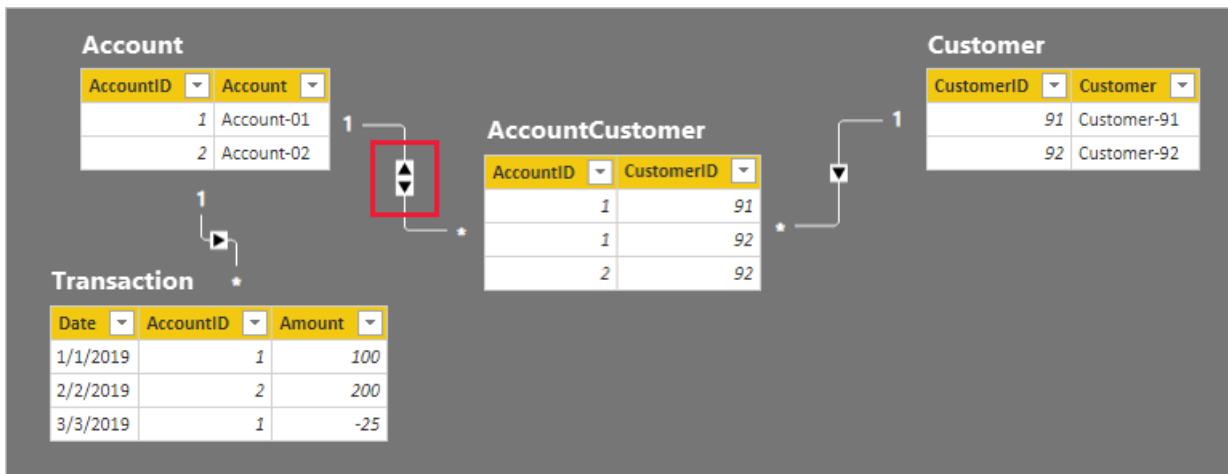
The second visual is titled **Customer Balance**, and it has two columns: **Customer** and **Amount**. It displays the following result:

- 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. It'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 the **Customer Balance** 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. The issue is related to filter propagation. It's not flowing all the way to the **Transaction** table.

Follow the relationship filter directions from the **Customer** table to the **Transaction** table. It should be apparent that the relationship between the **Account** and **AccountCustomer** table is propagating in the wrong direction. The filter direction for this relationship must be set to **Both**.



Account Balance		Customer Balance	
Account	Amount	Customer	Amount
Account-01	75	Customer-91	75
Account-02	200	Customer-92	275
Total	275	Total	275

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-type tables, we provide the following guidance:

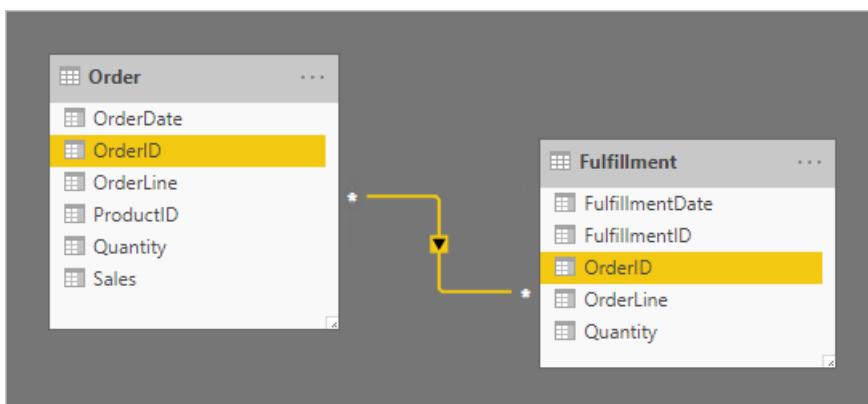
- Add each many-to-many related entity as a model table, ensuring it has a unique identifier (ID) column
- Add a bridging table to store associated entities
- Create one-to-many relationships between the three tables
- Configure **one** bi-directional relationship to allow filter propagation to continue to the fact-type tables
- When it isn't appropriate to have missing ID values, set the **Is Nullable** property of ID columns to FALSE—data refresh will then fail if missing values are sourced
- Hide the bridging table (unless it contains additional columns or measures required for reporting)
- Hide any ID columns that aren't suitable for reporting (for example, when IDs are surrogate keys)
- 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. It results in the best 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 you relate many-to-many dimension-type tables directly. This design approach requires configuring a relationship with a many-to-many cardinality. Conceptually it can be achieved, yet it implies that the related columns will contain duplicate values. It's a well-accepted design practice, however, that dimension-type tables have an ID column. Dimension-type tables should always use the ID column as the "one" side of a relationship.

Relate many-to-many facts

The second many-to-many scenario type involves relating two fact-type tables. Two fact-type 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-type tables: **Order** and **Fulfillment**. The **Order** table contains one row per order line, and the **Fulfillment** table can contain 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 two **OrderID** columns, with filter propagation only from the **Order** table (**Order** filters **Fulfillment**).



The relationship cardinality is set to many-to-many to support storing duplicate **OrderID** values in both tables. In the **Order** table, duplicate **OrderID** values can exist because an order can have multiple lines. In the **Fulfillment** table, duplicate **OrderID** values can exist because orders may 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 screenshot shows the Power BI Data View with two tables: **Order** and **Fulfillment**.

Order Table Data:

OrderDate	OrderID	OrderLine	ProductID	OrderQuantity	Sales
01/01/2019	1	1	Prod-A	5	50
01/01/2019	1	2	Prod-B	10	80
02/02/2019	2	1	Prod-B	5	40
02/02/2019	2	2	Prod-C	1	20
03/03/2019	3	1	Prod-C	5	100

Fulfillment Table Data:

FulfillmentDate	FulfillmentID	OrderID	OrderLine	FulfillmentQuantity
01/01/2019	50	1	1	2
02/02/2019	51	2	1	5
02/02/2019	52	1	1	3
02/02/2019	53	1	2	10

A relationship line connects the **OrderID** column in the **Order** table to the **OrderID** column in the **Fulfillment** table. The cardinality for the **Order** side is indicated by an asterisk (*), and the cardinality for the **Fulfillment** side is also indicated by an asterisk (*). The relationship line is highlighted with a yellow arrowhead pointing from the **Order** table to the **Fulfillment** table.

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.

Order and Fulfillment Quantities		
OrderID	OrderQuantity	FulfillmentQuantity
1	15	15
2	6	5
3	5	
Total	26	20

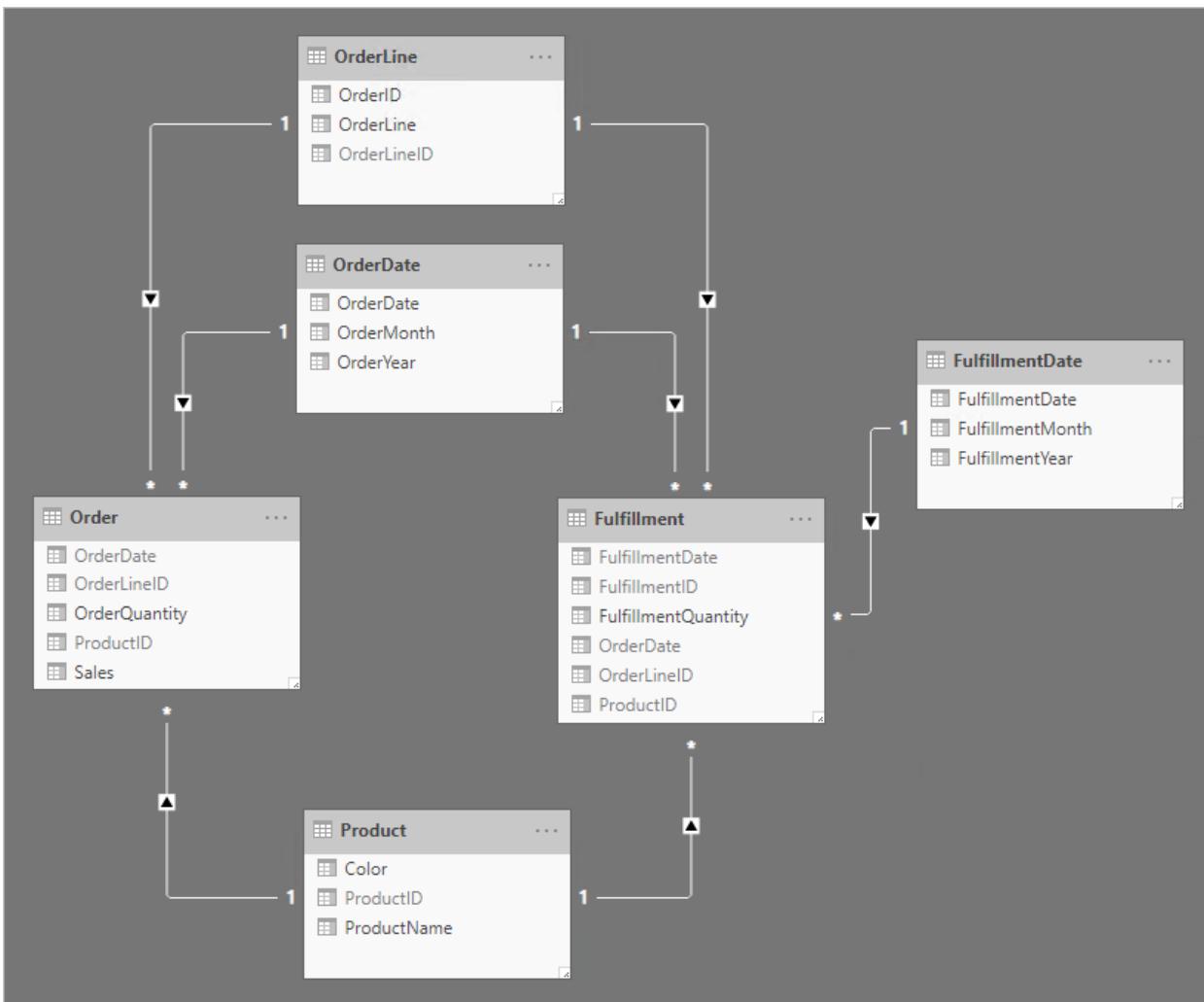
The visual presents an accurate result. However, the usefulness of the model is limited—you can only filter or group by the Order table OrderID column.

Relate many-to-many facts guidance

Generally, we don't recommend relating two fact-type tables directly using many-to-many cardinality. The main reason is because the model won't provide flexibility in the ways you report visuals filter or group. In the example, it's only possible for visuals to filter or group by the Order table OrderID column. An additional reason relates to the quality of your data. If your data has integrity issues, it's possible some rows may be omitted during querying due to the nature of the *limited relationship*. For more information, see [Model relationships in Power BI Desktop \(Relationship evaluation\)](#).

Instead of relating fact-type tables directly, we recommend you adopt [Star Schema](#) design principles. You do it by adding dimension-type tables. The dimension-type tables then relate to the fact-type tables by using one-to-many relationships. This design approach is robust as it delivers flexible reporting options. It lets you filter or group using any of the dimension-type columns, and summarize any related fact-type table.

Let's consider a better solution.



Notice the following design changes:

- The model now has four additional tables: **OrderLine**, **OrderDate**, **Product**, and **FulfillmentDate**
- The four additional tables are all dimension-type tables, and one-to-many relationships relate these tables to the fact-type tables
- The **OrderLine** table contains an **OrderLineID** column, which represents the **OrderID** value multiplied by 100, plus the **OrderLine** value—a unique identifier for each order line
- The **Order** and **Fulfillment** tables now 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 relates only to the **Fulfillment** table
- All unique identifier columns are hidden

Taking the time to apply star schema design principles delivers the following benefits:

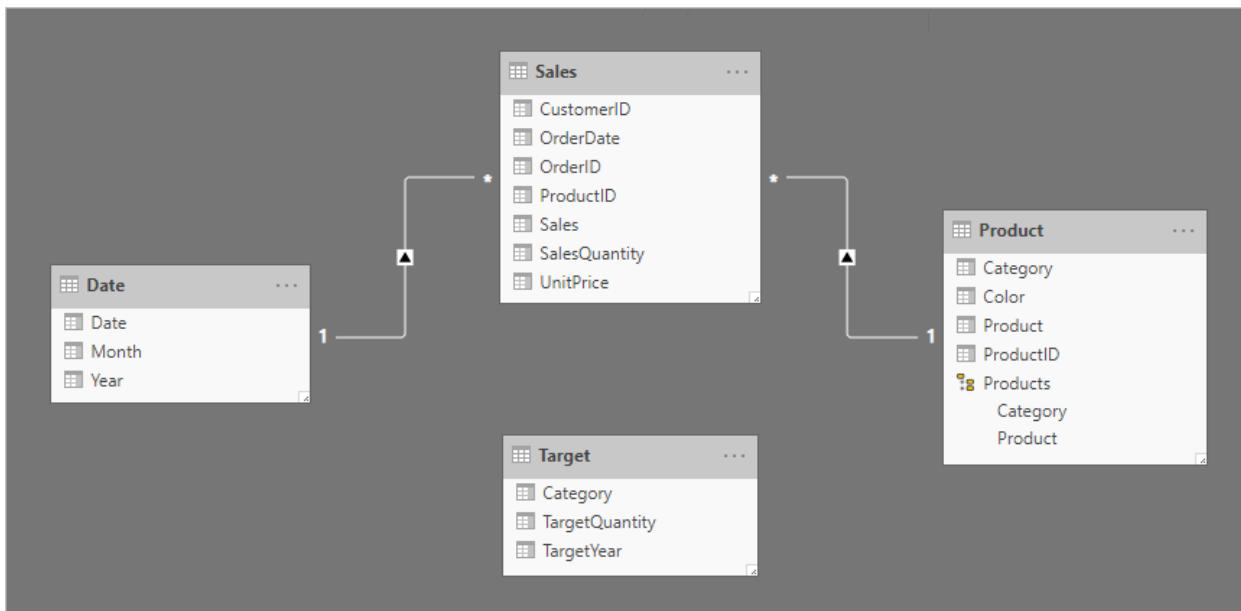
- Your report visuals can *filter or group* by any visible column from the dimension-type tables
- Your report visuals can *summarize* any visible column from the fact-type tables
- Filters applied to the **OrderLine**, **OrderDate**, or **Product** tables will propagate to both fact-type tables
- All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

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** are

dimension-type tables, and one-to-many relationships relate each to the **Sales** fact-type 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.

TargetYear	Category	TargetQuantity
01/01/2019	Clothing	10
01/01/2019	Accessories	20
01/01/2019	Clothing	30
01/01/2020	Accessories	40
01/01/2020	Food	100
01/01/2020	Food	200

Because the **Target** table stores data at a higher level than the dimension-type tables, a one-to-many relationship cannot 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-type tables.

Relate higher grain time periods

A relationship between the **Date** and **Target** tables should be a one-to-many relationship. It's because the **TargetYear** column values are dates. In this example, each **TargetYear** column value is 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 may 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 from a year into its months. The visual is summarizing the **TargetQuantity** column. (The [Show items with no data](#) option has been enabled for the matrix rows.)

Targets	
Year	TargetQuantity
2019	130
2020	270
2020-01	270
2020-02	
2020-03	
2020-04	

To avoid this behavior, we recommend 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** or **Month** columns aren't filtered.

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

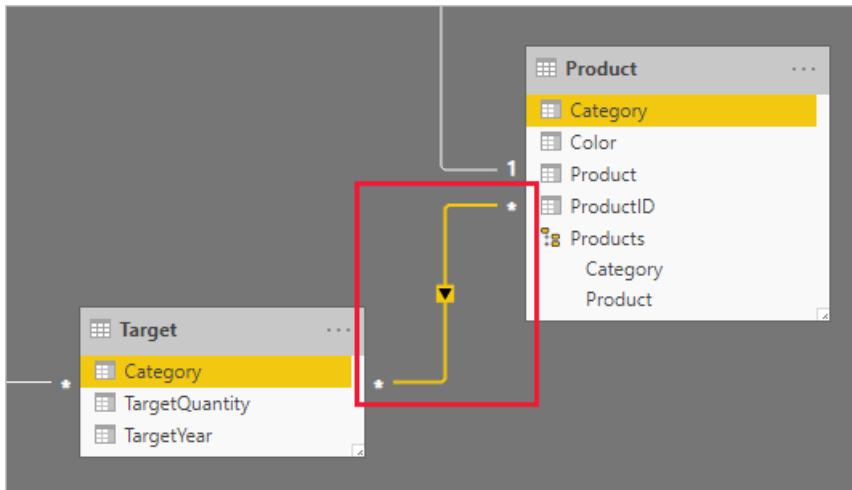
The following matrix visual now uses the **Target Quantity** measure. It shows that all monthly target quantities are BLANK.

Targets	
Year	Target Quantity
2019	130
2020	270
2020-01	
2020-02	
2020-03	
2020-04	

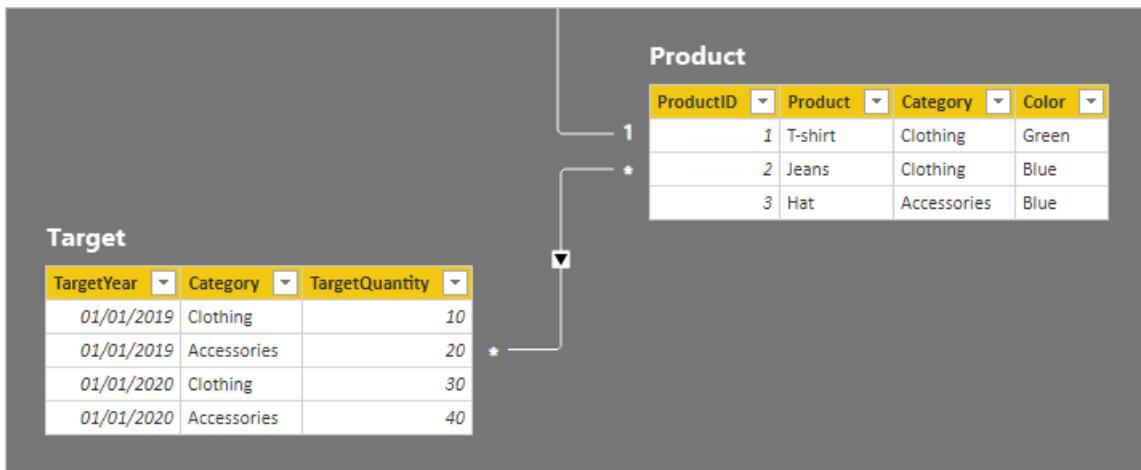
Relate higher grain (non-date)

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

The **Category** columns (from both the **Product** and **Target** tables) contains duplicate values. So, there's no "one" 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-type table to the fact-type table.



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



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.

Category Targets	
Category	TargetQuantity
Accessories	60
Clothing	40
Total	100

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.

Color Targets?	
Color	TargetQuantity
Blue	100
Green	40
Total	100

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 you control the summarization of your fact data by using measures.

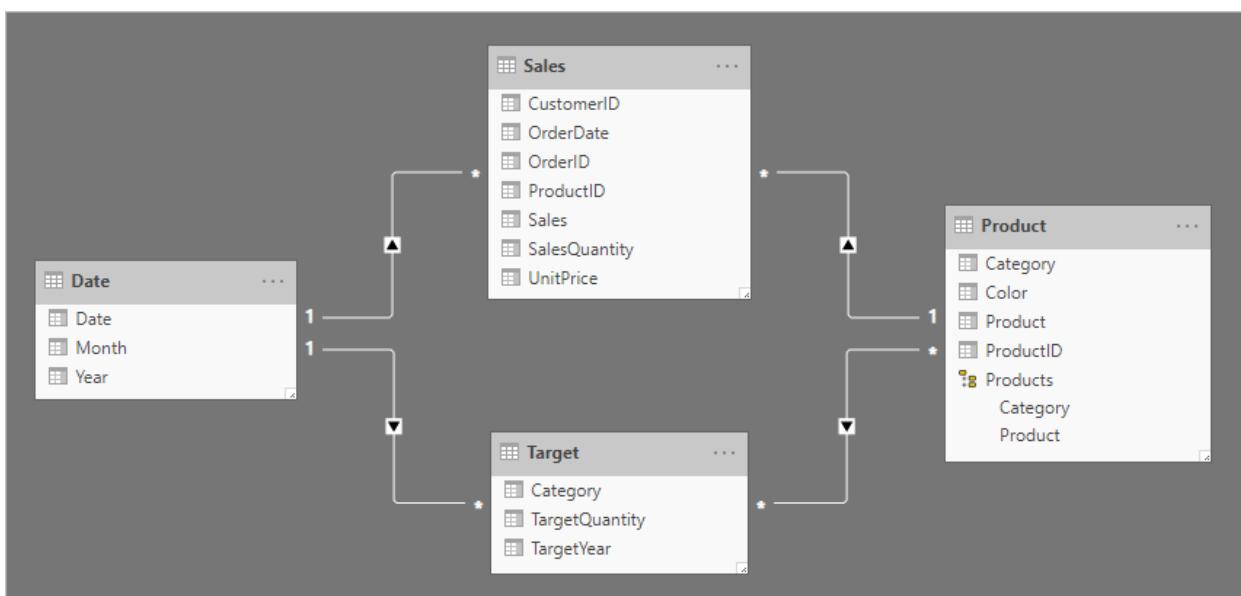
Consider the following measure definition. Notice that all **Product** table columns that are beneath the category level are tested for filters.

```
Target Quantity =  
IF(  
    NOT ISFILTERED('Product'[ProductID])  
    && NOT ISFILTERED('Product'[Product])  
    && NOT ISFILTERED('Product'[Color]),  
    SUM(Target[TargetQuantity])  
)
```

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

Color Targets?	
Color	Target Quantity
Blue	
Green	
Total	100

The final model design looks like the following.



Relate higher grain facts guidance

When you need to relate a dimension-type table to a fact-type table, and the fact-type table stores rows at a higher grain than the dimension-type table rows, we provide the following guidance:

- For higher grain fact dates:
 - In the fact-type table, store the first date of the time period

- Create a one-to-many relationship between the date table and the fact-type table
- For other higher grain facts:
 - Create a many-to-many relationship between the dimension-type table and the fact-type table
- For both types:
 - Control summarization with measure logic—return BLANK when lower-level dimension-type columns are used to filter or group
 - Hide summarizable fact-type table columns—this way, only measures can be used to summarize the fact-type table

Next steps

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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Active vs inactive relationship guidance

3/24/2022 • 6 minutes to read • [Edit Online](#)

This article targets you as a data modeler working 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.

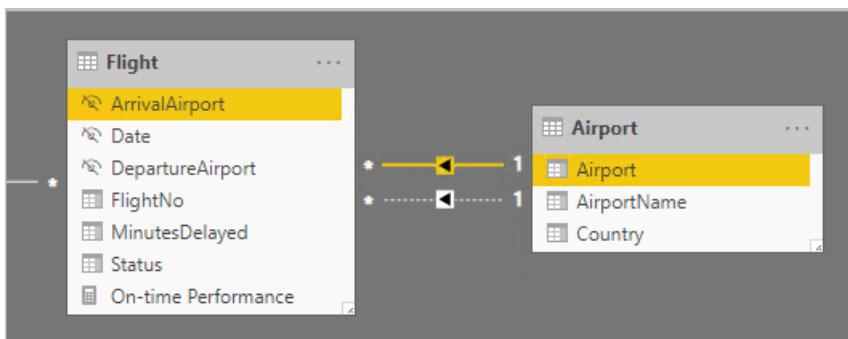
It's also important that you have an understanding of star schema design. For more information, see [Understand star schema and the importance for Power BI](#).

Active relationships

Generally, we recommend defining 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-type table storing 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-type table storing one row per airport. Each row describes the airport code, airport name, and the country.

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

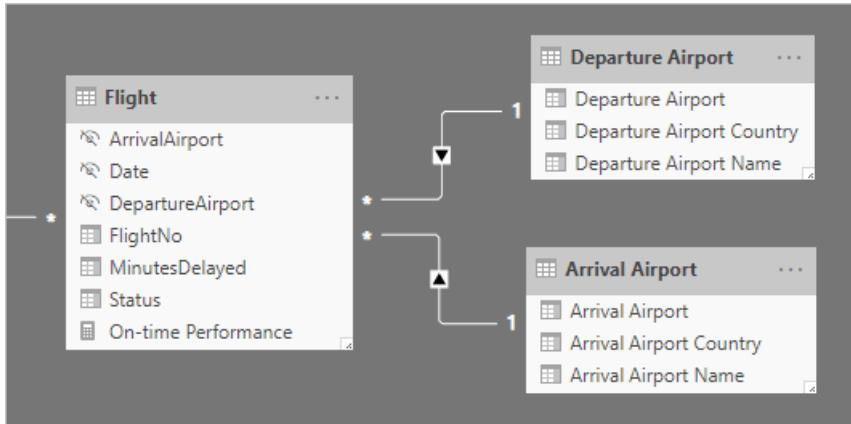


There are two model relationships between the **Flight** and **Airport** tables. In the **Flight** table, the **DepartureAirport** and **ArrivalAirport** columns relate to the **Airport** column of the **Airport** table. In star schema design, the **Airport** table is described as a [role-playing dimension](#). 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 for Power BI models. It's because model relationships are paths for filter propagation, and these paths must be deterministic. For this reason, a model cannot have multiple active relationships between two tables. Therefore—as described 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. This 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 reporting requirements involve filtering (or grouping) by departure and arrival airports *at the same time*, two active relationships are needed. 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**. The model relationships between these tables and the **Flight** table are 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.

On-time Departures			
Month	Arrival Airport	Flights	On-time Performance
2019 December	Perth	61	99.12%
	Adelaide	30	98.21%
	Brisbane	62	95.48%
	Canberra	75	95.05%
	Hobart	29	94.14%
	Sydney	119	92.31%
	Darwin	28	91.75%
	Total	404	94.87%

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

NOTE

For Import models, the additional 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-type tables contain low row counts relative to fact-type 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-type table, to a design with *one table per role*.

1. Remove any inactive relationships.

2. Consider renaming the role-playing dimension-type table to better describe its role. In the example, 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 defining 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.

```
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, 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 their questions.

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

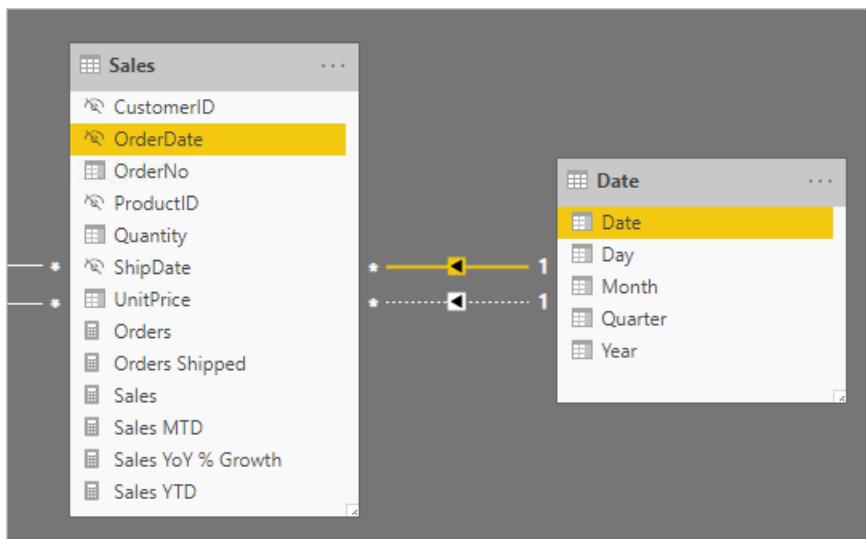
Inactive relationships

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

Let's now 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 by) 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 will propagate to the **OrderDate** column.

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

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

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

This model design supports producing the following report design.

Quarter	Monthly Sales			
	Month	Sales	Orders	Orders Shipped
□ (Blank)				
□ 2019 Q1				
□ 2019 Q2				
□ 2019 Q3				
■ 2019 Q4				
	2019 October	\$288,618	11,006	9,536
	2019 November	\$437,234	15,923	16,421
	2019 December	\$584,278	21,363	17,916
	Total	\$1,310,130	48,292	43,873

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 item. This slicer item appears as a result of [table expansion](#). While each **Sales** table row has an 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.

Recommendations

In summary, we recommend defining 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. It means that role-playing dimension-type tables should be duplicated in your model.

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

Next steps

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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Bi-directional relationship guidance

3/24/2022 • 5 minutes to read • [Edit Online](#)

This article targets you as a data modeler working 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 minimizing the use of bi-directional relationships. They can negatively impact on model query performance, and possibly deliver confusing experiences for your report users.

There are three scenarios when bi-directional filtering can solve specific requirements:

- [Special model relationships](#)
- [Slicer items "with data"](#)
- [Dimension-to-dimension analysis](#)

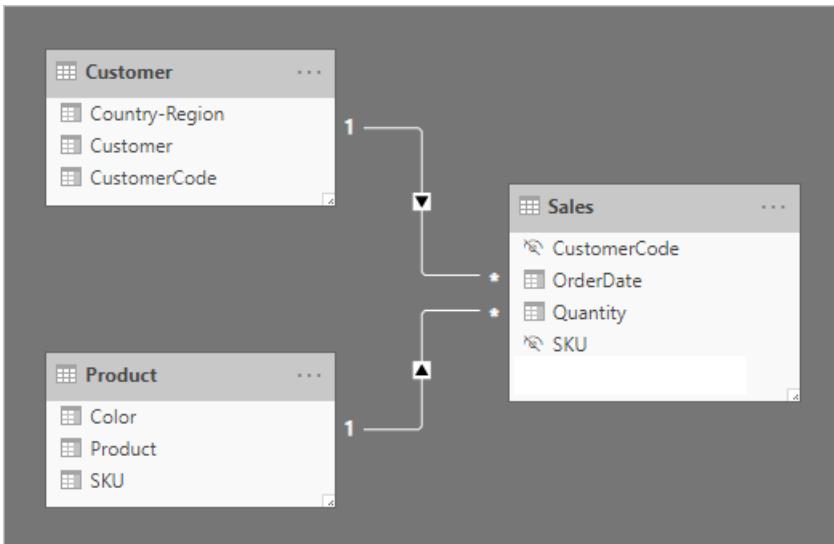
Special model relationships

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 designs, see [One-to-one relationship guidance](#).
- **Many-to-many:** When relating two dimension-type 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 \(Relate many-to-many dimensions\)](#).

Slicer items "with data"

Bi-directional relationships can deliver slicers that limit items 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 dataset, or an Analysis Services model.) To help explain what it means, first consider the following model diagram.

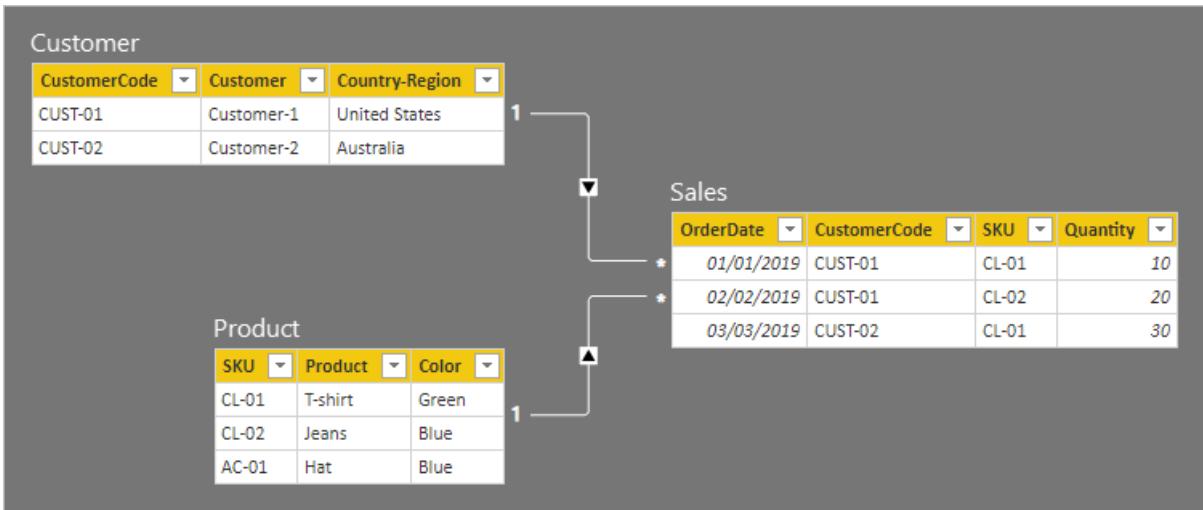


The first table is named **Customer**, and it contains three columns: **Country-Region**, **Customer**, and **CustomerCode**. The second table is named **Product**, and it contains three columns: **Color**, **Product**, and **SKU**. The third table is named **Sales**, and it contains four columns: **CustomerCode**, **OrderDate**, **Quantity**, and **SKU**. The **Customer** and **Product** tables are dimension-type 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.

NOTE

It's not possible to display table rows in the Power BI Desktop model diagram. It's done in this article to support the discussion with clear examples.



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:

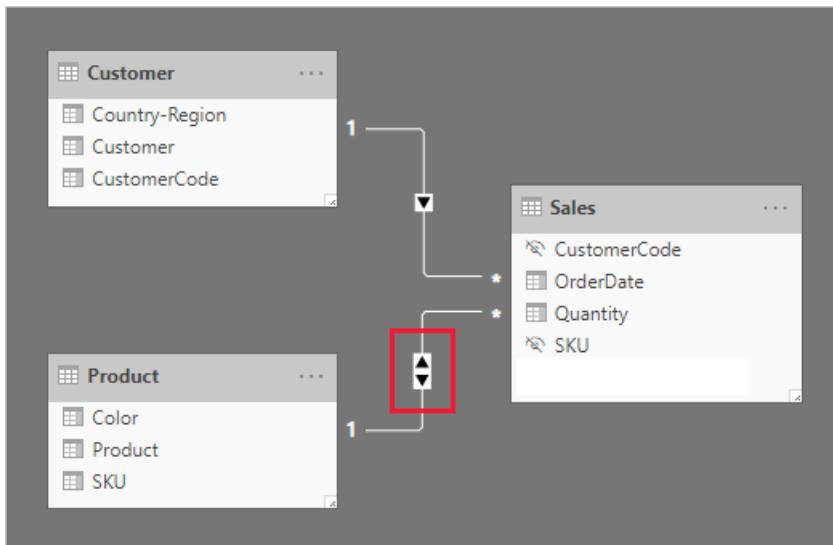
- o OrderDate January 1 2019, CustomerCode CUST-01, SKU CL-01, Quantity 10
- o OrderDate February 2 2019, CustomerCode CUST-01, SKU CL-02, Quantity 20
- o OrderDate March 3 2019, CustomerCode CUST-02, SKU CL-01, Quantity 30

Now consider the following report page.



The page consists of two slicers and a card visual. The first slicer is for **Country-Region** and it has two items: Australia and United States. It currently slices by Australia. The second slicer is for **Product**, and it has three items: 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 items where data *relates* to Australian sales. It's what's meant by showing slicer items "with data". You can achieve this behavior by configuring the relationship between the **Product** and **Sales** table to filter in both directions.



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



We first suggest you consider carefully whether this design works for your report users. Some report users find the experience confusing. They don't understand why slicer items dynamically appear or disappear when they interact with other slicers.

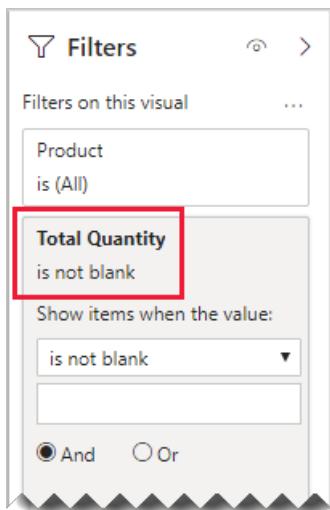
If you do decide to show slicer items "with data", we don't recommend you configure bi-directional relationships. Bi-directional relationships require more processing and so they can negatively impact on query performance—especially as the number of bi-directional relationships in your model increases.

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** table no longer filters in both directions. And, the following measure definition has been added to the **Sales** table.

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

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



Dimension-to-dimension analysis

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

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

As the fact-type table behaves like a bridging table, you can follow the many-to-many relationship guidance to relate two dimension-type tables. It will require configuring at least one relationship to filter in both directions. For more information, see [Many-to-many relationship guidance \(Relate many-to-many dimensions\)](#).

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 items "with data"*. So, we recommend that you activate bi-directional filtering *in a measure definition* by using the **CROSSFILTER** DAX function instead. The **CROSSFILTER** function can be used 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 configured to filter in a *single direction*.

```

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

```

During the evaluation of the **Different Countries Sold** measure expression, 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.

Product Sales		
Product	Quantity	Different Countries Sold
Jeans	20	1
T-shirt	40	2
Total	60	2

Next steps

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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Relationship troubleshooting guidance

3/24/2022 • 3 minutes to read • [Edit Online](#)

This article targets you as a data modeler working with Power BI Desktop. It provides you with guidance on how to troubleshoot specific issues you may 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 configured 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 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 Data 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 configured—they could be incorrect if a "many"-side column presently contains unique values, and has been incorrectly configured 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
10. Switch to Data view, and verify that matching values can be found in related columns

Troubleshooting guide

Here's a list of issues together with possible solutions.

ISSUE	POSSIBLE REASON(S)
-------	--------------------

ISSUE	POSSIBLE REASON(S)
The visual displays no result	<ul style="list-style-type: none"> - The model is yet to be loaded with data - No data exists within the filter context - Row-level security is enforced - Relationships aren't propagating between tables—<i>follow checklist above</i> - Row-level security 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	<ul style="list-style-type: none"> - Relationships don't exist - Relationships aren't propagating between tables—<i>follow checklist above</i>
The visual displays results, but they aren't correct	<ul style="list-style-type: none"> - Visual is incorrectly configured - Measure logic is incorrect - Model data needs to be refreshed - Source data is incorrect - Relationship columns are incorrectly related (for example, ProductID column maps to CustomerID) - 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	<ul style="list-style-type: none"> - 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 (Regular relationships) - It's a regular one-to-one relationship, and related columns contain BLANKs—see Model relationships in Power BI Desktop (Regular relationships) - An inactive relationship "many"-side column stores BLANKs, or has values not stored on the "one"-side
The visual is missing data	<ul style="list-style-type: none"> - Incorrect/unexpected filters are applied - Row-level security is enforced - It's a limited relationship, and there are BLANKs in related columns, or data integrity issues—see Model relationships in Power BI Desktop (limited relationships) - It's a relationship between two DirectQuery tables, the relationship is configured to assume referential integrity, but there are data integrity issues (mismatched values in related columns)
Row-level security is not correctly enforced	<ul style="list-style-type: none"> - Relationships aren't propagating between tables—<i>follow checklist above</i> - Row-level security is enforced, but a bi-directional relationship isn't enabled to propagate—see Row-level security (RLS) with Power BI Desktop

Next steps

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

- [Model relationships in Power BI Desktop](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

DirectQuery model guidance in Power BI Desktop

3/24/2022 • 17 minutes to read • [Edit Online](#)

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 is not 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.

This article does not directly cover Composite models. A Composite model will consist 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 are not in scope for this article. For more information, see [Use composite models in Power BI Desktop](#).

It is 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.

Design in Power BI Desktop

Both Azure Synapse Analytics (formerly SQL Data Warehouse) and Azure HDInsight Spark data sources can be connected to directly, without the need to use Power BI Desktop. It is achieved in the Power BI service by "Getting Data" and choosing the Databases tile. For more information, see [Azure Synapse Analytics \(formerly SQL Data Warehouse\) with DirectQuery](#).

While direct connect is convenient, we don't recommend that you use this approach. The main reason is that it is not possible to refresh the model structure should the underlying data source schema change.

We recommend that you use Power BI Desktop to create and manage all of your DirectQuery models. This approach provides you with complete control to define the model that you need, including the use of supported features like hierarchies, calculated columns, measures, and more. It will also allow you to revise the model design should the underlying data source schema change.

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 is especially important that dimension-type tables contain a column of unique values (dimension key) that maps to the fact-type table(s). It's also important that fact-type dimension columns contain valid dimension key values. They will 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 will ensure every **Sales** 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 leverage Massively Parallel Processing (MPP) architecture, consider configuring large fact-type tables as hash distributed, and dimension-type 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](#).

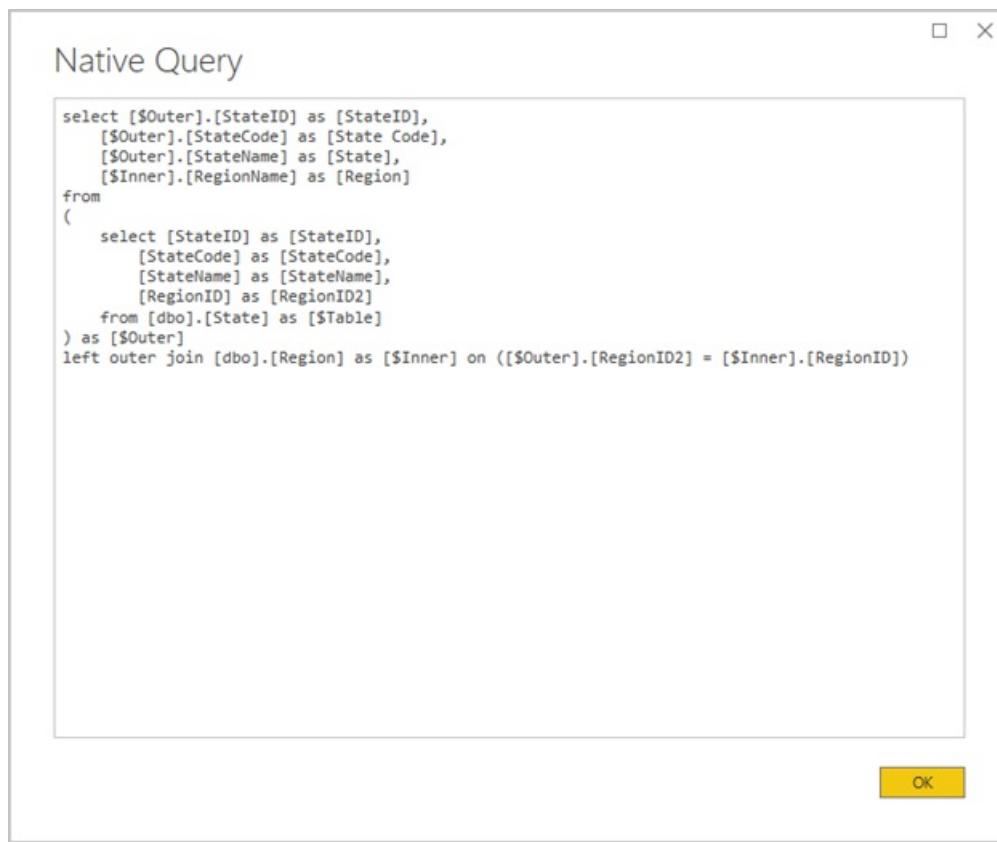
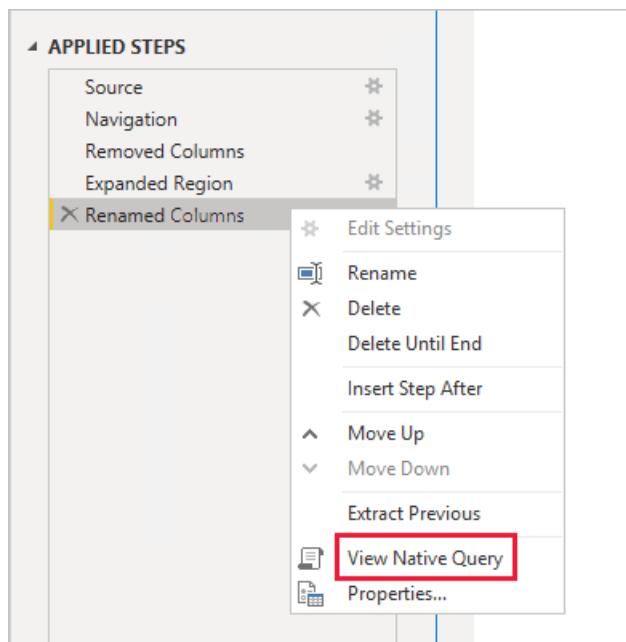
Consider also indexed views that can pre-aggregate fact table data at a higher grain. For example, if the **Sales** table stores data at order line level, you could create a view to summarize this data. The view could be based on a SELECT statement that groups the **Sales** table data by date (at month 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 is 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 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:

```

...
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.

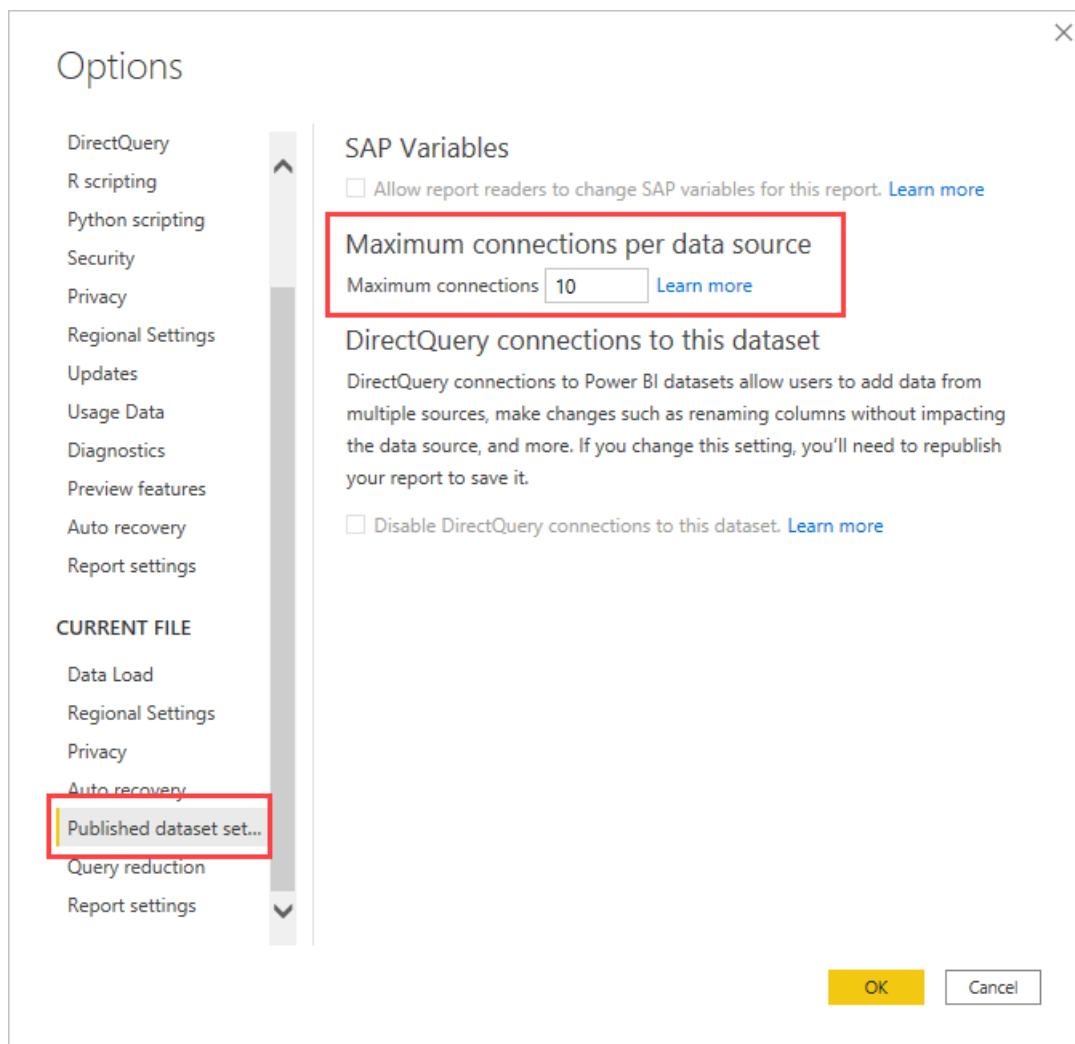
- **Keep measures simple:** At least initially, it's recommended to limit measures to simple aggregates. The aggregate functions include SUM, COUNT, MIN, MAX, and AVERAGE. Then, if the measures are sufficiently responsive, you can experiment with more complex measures, but paying attention to the performance 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 do not 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 is necessary to relate tables by using multiple columns. For example, the **Sales** and **Geography** tables are related by two columns: **Country** 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 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-type tables, which is a common practice in relational data warehouse designs.

There is 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 does not natively support the unique identifier (GUID) data type. When defining a relationship between columns of this type, Power BI will generate 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 is usually the primary key column of dimension-type tables.) When hidden, it is not available in the **Fields** pane and so cannot 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 will generate a query that does not 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 will generate 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.

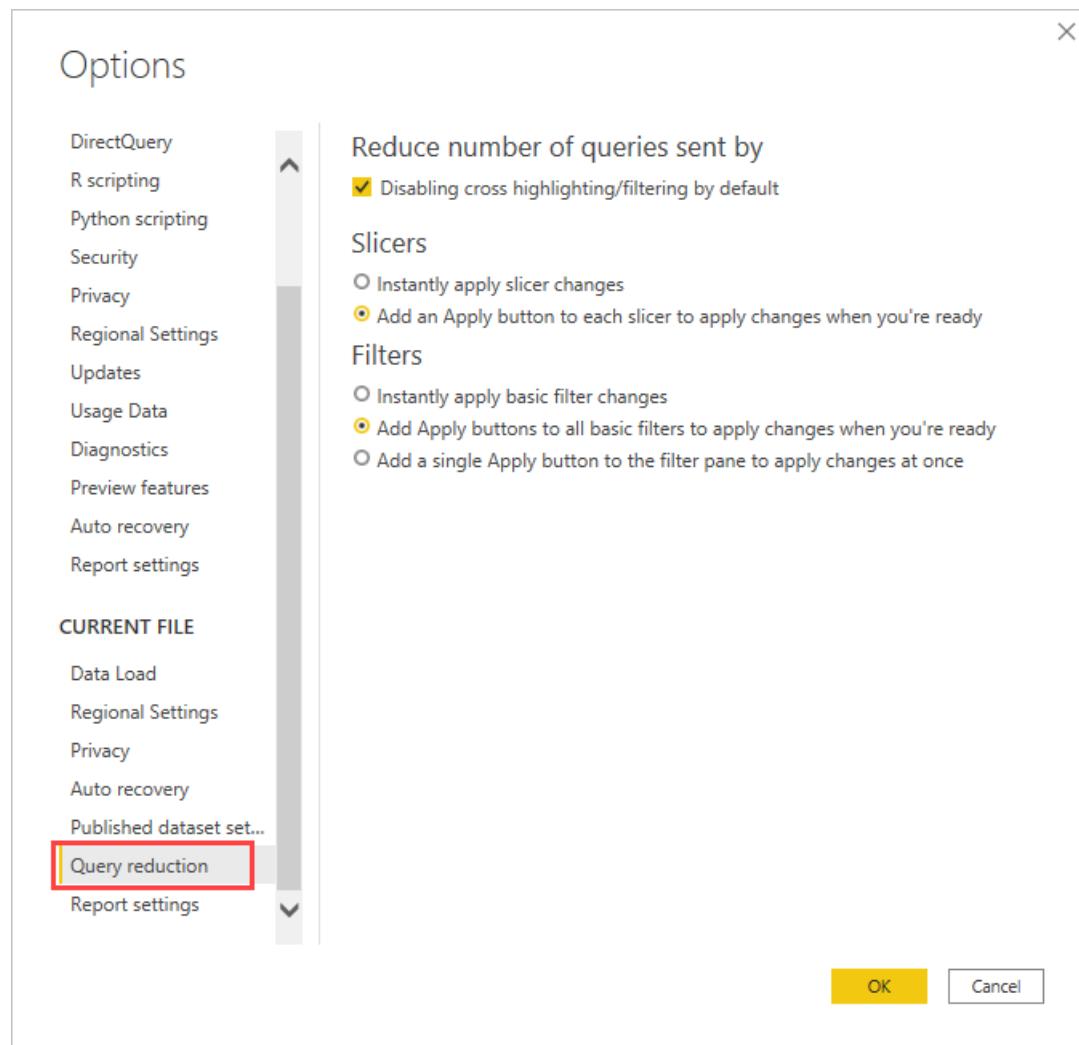
When the model is published to Power BI, the maximum number of concurrent queries sent to the underlying data source also depends on the environment. Different environments (such as Power BI, Power BI Premium, or Power BI Report Server) each can impose different throughput constraints. For more information about Power BI Premium capacity resource limitations, see [Deploying and Managing Power BI Premium Capacities](#).

Optimize report designs

Reports based on a DirectQuery dataset 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 is also possible to show an

Apply button on slicers and filters. The slicer or filter options will not 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 **Country** 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 whilst 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 is 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 is higher chance that they will 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.

The screenshot shows a Power BI interface. On the left is a table visual with the following data:

Category	Sales
Collective pitch	\$73,565,526.95
Trainer	\$22,801,448.90
Warbird	\$15,264,362.90
Total	\$111,631,338.75

On the right is the 'Filters' pane, which displays the filter applied to the 'Sales' measure:

Filters on this visual

Category
is (All)

Sales
is greater than \$15,000,000.00
Show items when the value:
is greater than
15000000
● And ○ Or

Add data fields here

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 is not 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 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 are interested in), each new selection results in a new query being sent to the underlying source. Whilst 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 is 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 is important to educate your users on how to efficiently work with reports based on DirectQuery datasets. 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 datasets. 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 are 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.

Next steps

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 Power BI Community](#)

Composite model guidance in Power BI Desktop

3/24/2022 • 6 minutes to read • [Edit Online](#)

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

NOTE

An introduction to Composite models is not covered in this article. If you're not completely familiar with Composite models, we recommend 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

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, cannot 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](#).

Further, you can consider developing a Composite model in the following situations.

- Your model could be a DirectQuery model, but you want to boost performance. In a Composite model, performance can be improved by configuring appropriate storage for each table. You can also add [aggregations](#). Both of these optimizations are discussed later in this article.
- You want to combine a DirectQuery model with additional data, which must be imported into the model. Imported data can be loaded from a different data source, or from calculated tables.
- You want to combine two or more DirectQuery data sources into a single model.

NOTE

Composite models cannot combine connections to external analytic databases. These include live connections to [external-hosted models](#), Power BI datasets, SAP Business Warehouse, and SAP HANA when [treating SAP HANA as a multidimensional source](#).

Optimize model design

A Composite model can be optimized by configuring table [storage modes](#), and by adding [aggregations](#).

Table storage mode

In a Composite model, you can configure the storage mode for each table (except calculated tables):

- **DirectQuery:** We recommend 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-type tables—tables used for summarization.

- **Import:** We recommend you set this mode for tables that are not 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 you set this mode for dimension-type tables, when there's a possibility they'll be queried together with DirectQuery fact-type tables from the same source.
- **Hybrid:** We recommend you configure this mode by adding Import partitions, as well as one DirectQuery partition to a fact table if you want to include the latest data changes in real time, or if 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):** All data is retrieved from the model cache. It will deliver the fastest possible performance. This scenario is common for dimension-type tables queried by filters or slicer visuals.
- **Queries Dual table(s) or DirectQuery table(s) from the same source:** All data is retrieved 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-type tables and DirectQuery fact-type 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 is a combination of the previous two scenarios. The data is retrieved from the model cache if it is available in Import partitions, otherwise by sending 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-type tables and DirectQuery fact-type 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 a virtual table. In this case, the native query can be inefficient, especially for large grouping sets. And, it has the potential to expose sensitive data in the native query.

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
- Set the storage mode to **DirectQuery** when a table is a fact-type table storing large data volumes, or 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 through TOM, TMSL, or a third-party tool. For more information, see [Incremental refresh and real-time data for datasets](#).
- Set the storage mode to **Dual** when a table is a dimension-type table, and it will be queried together with **DirectQuery** or **Hybrid** fact-type tables based on the same source
- Configure 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 data sources (including the model cache)—limited relationships will eliminate rows when related column values don't match
- Optimize DirectQuery data sources with appropriate indexes for efficient joins, filtering and grouping
- Don't load sensitive data into Import, Dual or Hybrid tables if there's risk of a native query being intercepted—for more information, see [Use composite models in Power BI Desktop \(Security implications\)](#)

Aggregations

You can add aggregations to DirectQuery tables in your Composite model. Aggregations are cached in the model, and so they behave as Import tables (although they can't be used like a model table). Their purpose is to improve performance for "higher grain" queries. For more information, see [Aggregations in Power BI Desktop](#).

NOTE

Hybrid tables do not support aggregations because some of the partitions operate in Import mode. It is not possible to add aggregations at the level of an individual DirectQuery partition.

We recommend that an aggregation table 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 there's an adequate performance gain relative to the cost of creating and maintaining the aggregation table.

Next steps

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](#)
- [DirectQuery model troubleshooting in Power BI Desktop](#)
- [Power BI data sources](#)
- [Storage mode in Power BI Desktop](#)
- [Aggregations in Power BI Desktop](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

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

3/24/2022 • 8 minutes to read • [Edit Online](#)

This article targets you as a data modeler working with Power BI Desktop. It describes good design practices for enforcing row-level 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:

```
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:

```
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-type tables, and not fact-type tables. And, rely on well-designed relationships to ensure RLS filters propagate to other model tables. 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 dataset roles. Only dataset 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 or security groups. Whenever possible, we recommend you map security groups to dataset roles. It involves managing security group memberships in Azure Active Directory. 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:

```
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 will result in the table returning no rows.

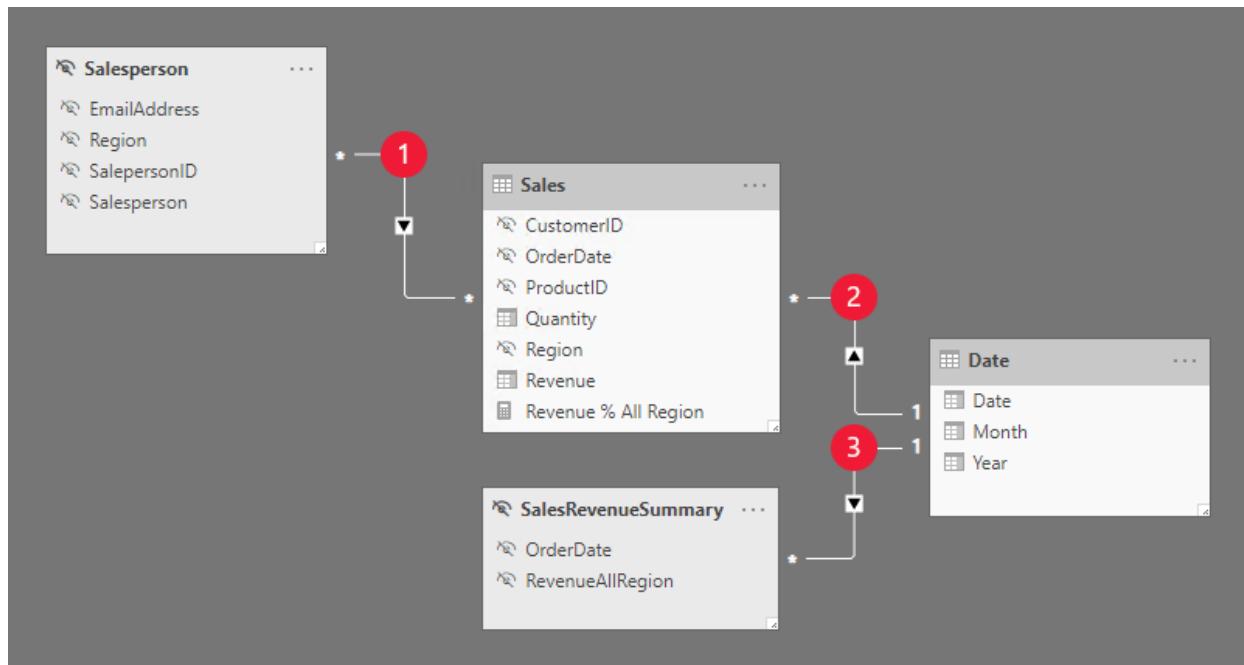
```
IF(
    USERNAME() = "Worker",
    [Type] = "Internal",
    IF(
        USERNAME() = "Manager",
        TRUE(),
        FALSE()
    )
)
```

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:

```

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:

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

Each of the three model relationships is described in the following 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 relationships between the Date and Sales tables.
3	There's a one-to-many relationships between the Date and SalesRevenueSummary tables.

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

```

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.

Avoid using RLS

Avoid using RLS, whenever it makes sense to do so. If you have only a small number of simplistic RLS rules that apply static filters, consider publishing multiple datasets instead. None of the datasets define roles because each dataset 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 dataset *for each sales region* to different workspaces. The datasets 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 dataset 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 are 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 beneath 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 additional 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.
- 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.

When a specific user can see all data, it's possible they're accessing reports directly in the workspace and they're the dataset owner. RLS is only enforced when:

- The report is opened in an app.
- The report is opened in a workspace, and the user is mapped to the [Viewer role](#).

Next steps

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
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Separate reports from models in Power BI Desktop

3/24/2022 • 3 minutes to read • [Edit Online](#)

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 a Power BI dataset 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

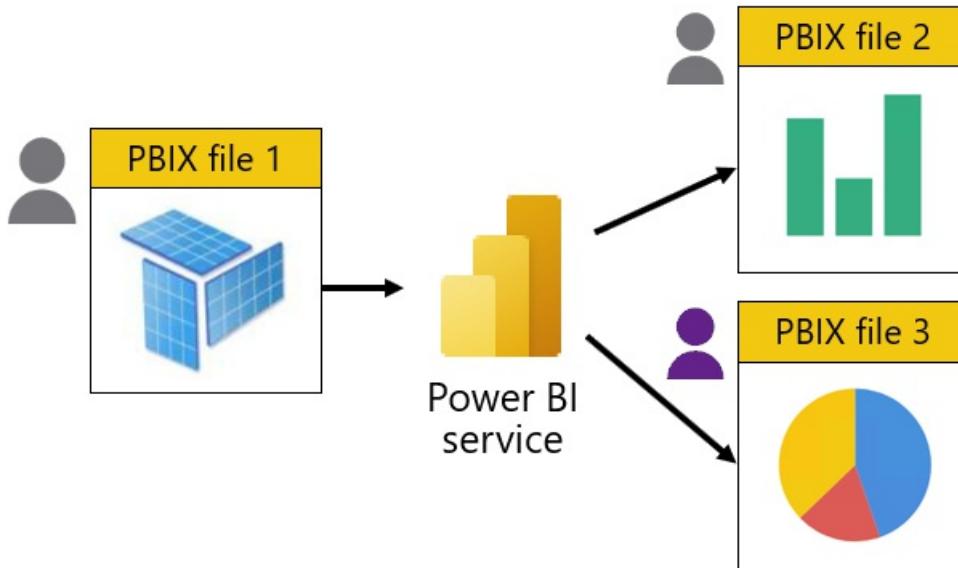
A *single file solution* works well when there's only ever a single report based on the model. In this case, it's likely that both the model and report are the efforts of the same person. We define it as a *Personal BI* solution, though the report could be shared with others. Such solutions can represent 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 dataset.

Preserve the model interface

Sometimes, model changes are inevitable. Data modelers must take care, then, not to 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 dataset](#) 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 artifact. In fact, it's a hyperlink that opens an email message addressed to the contact.

We recommend you contact the owner of each related artifact 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.

Next steps

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

- [Connect to datasets in the Power BI service from Power BI Desktop](#)
- [View related content in the Power BI service](#)
- [Data lineage](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Extend visuals with report page tooltips

3/24/2022 • 3 minutes to read • [Edit Online](#)

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.

Power BI visuals do not support page tooltips.

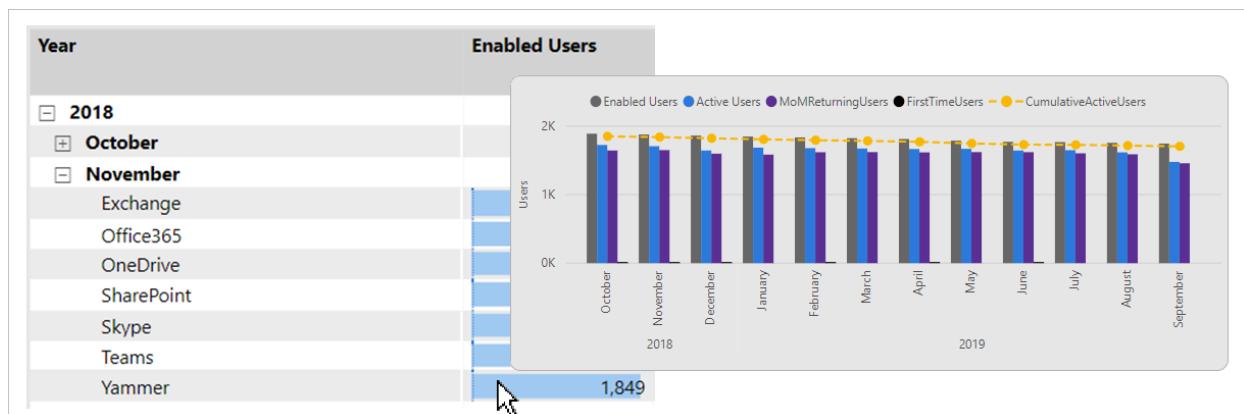
Here are some suggested design scenarios:

- [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.

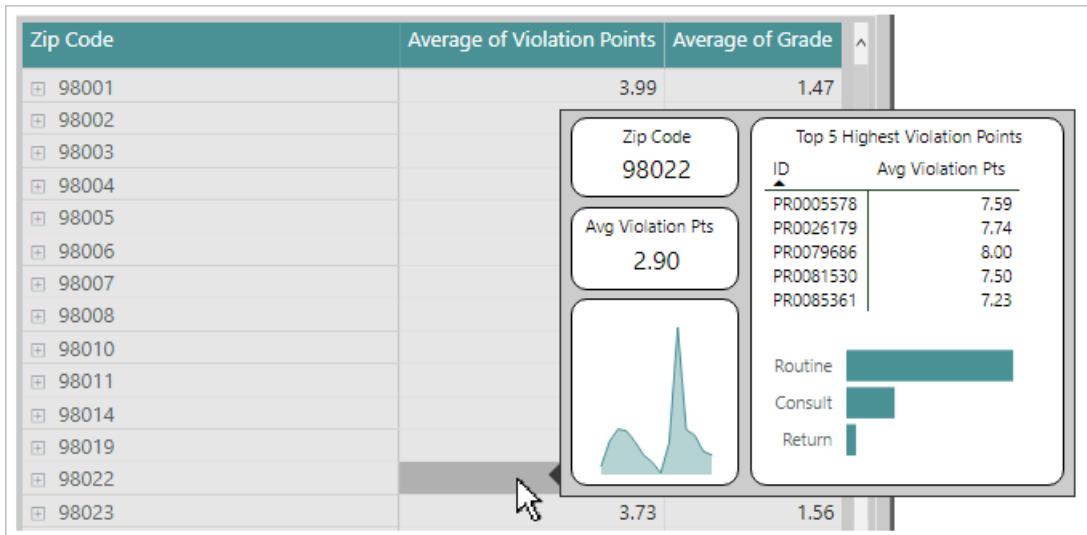


A page tooltip is revealed. It presents a different data visual (line and clustered column chart) and applies a contrasting time filter. Notice that the filter context for the data point is November 2018. Yet the page tooltip displays trend over *a full year of months*.

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.



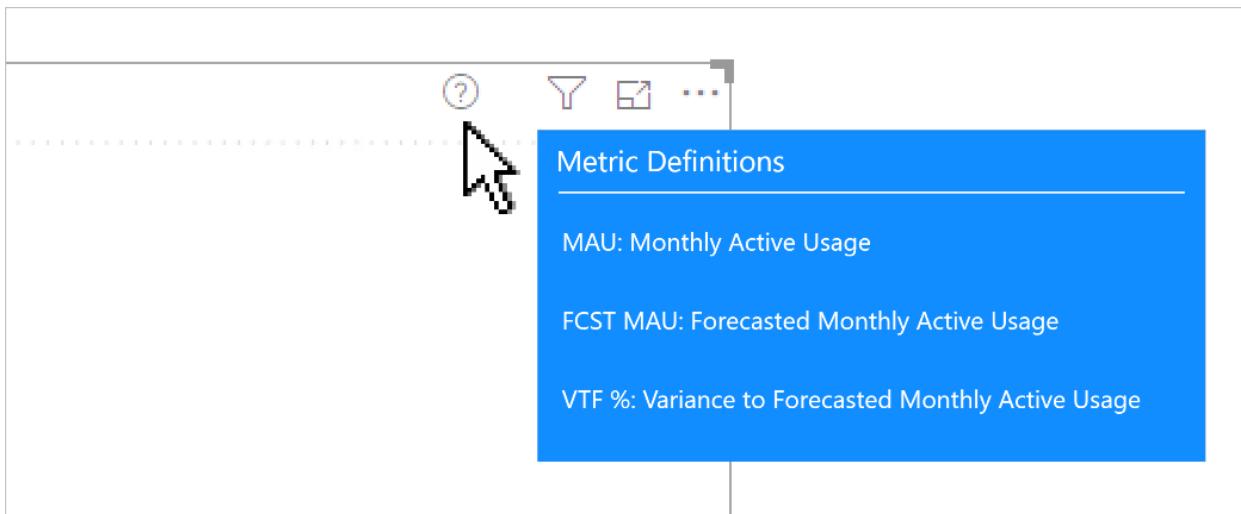
A page tooltip is revealed. It presents specific attributes and statistics for zip code 98022.

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.

Next steps

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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Use report page drillthrough

3/24/2022 • 2 minutes to read • [Edit Online](#)

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.

Sales Summary

Year	France	Germany	United Kingdom	Total
■ CY2018	\$1,428,020.38		\$1,406,491.96	\$2,834,512.33
■ CY2019	\$3,179,517.56	\$1,983,988.04	\$2,872,516.87	\$8,036,022.46
CY2019 Apr	\$53,834.40	\$192,211.66	\$116,094.10	\$362,140.17
CY2019 Aug	\$468,277.51	\$143,793.75	\$151,047.79	\$763,119.05
CY2019 Dec	\$119,236.30	\$136,457.24	\$425,979.11	\$681,672.65
CY2019 Feb	\$703,714.99	\$254,388.44	\$213,490.61	\$1,171,594.05
CY2019 Jan	\$55,465.41	\$180,040.57	\$152,484.49	\$387,990.47
CY2019 Jul	\$40,814.59	\$112,638.82	\$80,574.32	\$234,000.00
CY2019 Jun	\$153,406.10	\$161,191.86	\$418,461.00	\$733,000.00
CY2019 Mar	\$198,316.78	\$125,723.49	\$496,251.52	\$820,254.00
CY2019 May	\$638,169.15	\$185,310.66	\$198,629.90	\$1,022,109.00
CY2019 Nov	\$605,404.06	\$193,329.28	\$198,648.08	\$997,351.00
CY2019 Oct	\$54,602.78	\$163,616.96	\$110,014.25	\$328,254.00
CY2019 Sep	\$88,275.47	\$135,285.31	\$310,841.69	\$534,402.46
Total	\$4,607,537.94	\$1,983,988.04	\$4,279,008.83	\$10,870,534.80

See Records

Show data

Drillthrough ▾

Group

Copy ▾

Order Details

Order Details

Sales Order	Lines	Quantity	Sales	Freight
SO61175	325	57	\$25,882.74	\$647.07
SO61176	1	1	\$1,466.01	\$36.65
SO61190	820	159	\$37,202.78	\$930.07
SO61194	6	7	\$3,040.66	\$76.02
SO61197	6	10	\$7,351.62	\$183.79
SO61198	1	1	\$323.99	\$8.10
SO61214	21	7	\$4,706.06	\$117.65
SO61216	465	106	\$28,344.64	\$708.62
SO61229	1,035	151	\$24,336.00	\$608.40
SO61231	10	6	\$181.18	\$4.53
SO61238	630	143	\$22,608.86	\$565.22
SO61242	3	4	\$2,049.28	\$51.23
SO61246	6	4	\$1,409.38	\$35.23
SO61249	210	42	\$26,203.84	\$655.10
SO61252	3	2	\$372.59	\$9.31
SO61254	15	7	\$8,043.03	\$201.08
SO61258	210	101	\$10,879.40	\$271.99
SO61263	528	133	\$29,625.69	\$740.64
Total	4,295	941	\$234,027.74	\$5,850.70

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.

The screenshot shows a Power BI report interface. On the left is a table titled 'Zip Code' with columns for 'Zip Code', 'Average of Violation Points', and 'Average of Grade'. A cursor is hovering over the row for Zip Code 98004. A context menu is open, with a pink arrow pointing from the 'See Records' option to a detailed analysis page below. The analysis page is titled 'Zip Code Analysis' for Zip Code 98004. It displays three summary values: '3.20' (Zip Avg), '7.14' (Average of Inspection Score), and '1.43' (Average of Grade). Below these are two visualizations: a line chart titled 'Average of Grade by Year' showing a peak around 2012, and a map titled 'Average of Violation Points by Latitude and Longitude' showing several data points across a geographic area.

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 by adding links to Power BI reports. Links can define [URL parameters](#).

Next steps

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

- [Use drillthrough in Power BI Desktop](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to manage axes in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, working with visuals that have axes. These visuals include bar charts, columns charts, line charts, and many others.

Watch the video demonstrating eight tips to effectively manage axes in your Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top eight tips to effectively manage axes in Power BI reports include:

1. Visualize nominal categories
2. Visualize interval categories
3. Adjust X-axis labels
4. Adjust Y-axis labels
5. Manage X-axis hierarchies
6. Manage Y-axis hierarchies
7. Avoid the X-axis scrollbar
8. Remove axes to create sparklines

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 8 Tips To Effectively Manage Axes in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to control chart gridlines in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, working with chart visuals that have gridlines.

Watch the video demonstrating the top three tips to control gridlines in your Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top three tips to control chart gridlines in Power BI reports include:

1. Sort by value
2. Sort by time/numerical sequences
3. Sort by categorical sequences

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 3 Tips to Control Chart Gridlines in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to optimize the use of labels in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, working with visuals that have labels.

Watch the video demonstrating the top four tips to optimize the use of labels in your Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top four tips to optimize the use of labels in Power BI reports include:

1. Adjust label position
2. Adjust label color for contrast
3. Format labels for easy tracking
4. Avoid overwhelming labels

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 4 Tips to Optimize the Use of Labels in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to format and implement legends in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, when formatting and implementing legends.

Watch the video demonstrating the top six tips to format and implement legends in your Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top six tips to format and implement legends in Power BI reports include:

1. Sort legend variables
2. Emulate a legend
3. Size and positioning legends
4. Solve truncated legends
5. Match style format options
6. Craft comprehensive visuals with context

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 6 Tips for Legend Formatting & Implementation in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to optimize visual colors in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, configuring visual colors.

Watch the video demonstrating the top eight tips to optimize visual colors in your Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top eight tips to optimize visual colors in Power BI reports include:

1. Apply contrast within visuals
2. Implement optimal color palettes
3. Highlight important data with color
4. Consider color vision deficiencies
5. Use color with purpose
6. Choose appropriate color scales
7. Use color to differentiate key elements
8. Distinguish color and data

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 8 Tips To Optimize Visual Colors in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to sort and distribute data plots in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, when using data plot visuals.

Watch the video demonstrating the top nine tips to sort and distribute data plots in Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top nine tips to sort and distribute data plots in Power BI reports include:

1. Sort nominal categories
2. Sort interval categories
3. Sort ordinal categories
4. Visualize distant plots
5. Visualize crowded categorical plots
6. Visualize crowded time plots
7. Distribute multiple dimensions
8. Categorize data plots
9. Differentiate value plots

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 9 Tips to Sort and Distribute data plots in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Tips to improve analysis with shapes, images, and icons in Power BI reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

This article targets you as a report author designing Power BI reports, enhancing visuals with shapes, images, or icons.

Watch the video demonstrating the top four tips to improve analysis with shapes, images, and icons in Power BI reports.

NOTE

This video might use earlier versions of Power BI Desktop or the Power BI service.

Tips

In summary, the top four tips to improve analysis with shapes, images, and icons in Power BI reports include:

1. Design reports for accessibility
2. Use plot area images for context
3. Use images to convey information
4. Add icons to enrich report designs

Next steps

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

- [Tips for creating stunning reports](#)
- biDezine video: [Top 4 Tips to Improve Analysis with Pictograms in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

When to use paginated reports in Power BI

3/24/2022 • 5 minutes to read • [Edit Online](#)

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](#).

NOTE

Publishing Power BI paginated reports requires a Power BI Premium subscription. Reports will render only when they're in a workspace on a capacity that has [the Paginated Reports workload enabled](#).

Power BI paginated reports are optimized for **printing**, or **PDF generation**. They also provide you with the ability to produce highly formatted, pixel-perfect 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, pixel-perfect 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 develop a Power BI dataset. 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.

Next steps

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

- [What are paginated reports in Power BI Premium?](#)

- Migrate SQL Server Reporting Services reports to Power BI
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Data retrieval guidance for paginated reports

3/24/2022 • 10 minutes to read • [Edit Online](#)

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 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, are not suited to reports that need to retrieve very large datasets (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 dataset data sources, or any SQL Server Analysis Services or Azure Analysis Services model—tabular or multidimensional.

We suggest 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 can first develop a Power BI Desktop data model. This way, you can connect to over 100 [Power BI](#)

[data sources](#). Once published to the Power BI service, you can then develop a paginated report that connects to the Power BI dataset.

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 dataset.

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.

Next steps

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

- [Supported data sources for Power BI paginated reports](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Image use guidance for paginated reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

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. 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".

Next steps

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

- [What are paginated reports in Power BI Premium?](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Use cascading parameters in paginated reports

3/24/2022 • 6 minutes to read • [Edit Online](#)

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 \(Report Builder and SSRS\)](#).

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.

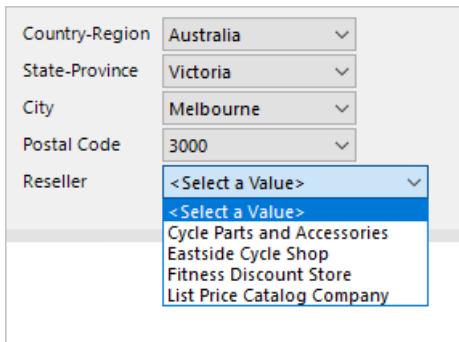
Filter large sets of items

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:

```
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:

```
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:

```
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:

```

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

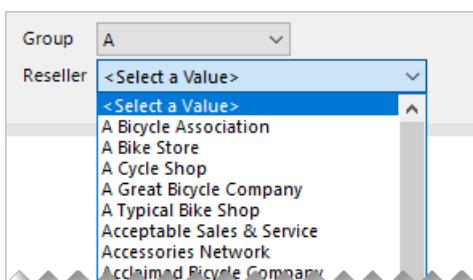
All query parameters (prefixed with the @ symbol) shown in these examples could be embedded within SELECT statements, or passed to stored procedures.

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:

```

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:

```

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](#).

```

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.

```

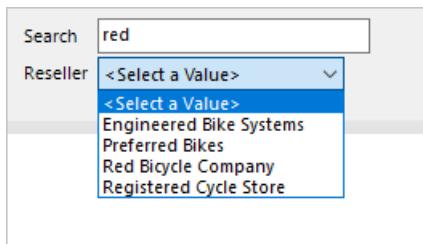
ALTER TABLE [Reseller]
ADD [ReportGroup2] AS CASE
    WHEN [ResellerName] LIKE '[A-C]%' THEN 'A-C'
    WHEN [ResellerName] LIKE '[D-H]%' THEN 'D-H'
    WHEN [ResellerName] LIKE '[I-M]%' THEN 'I-M'
    WHEN [ResellerName] LIKE '[N-S]%' THEN 'N-S'
    WHEN [ResellerName] LIKE '[T-Z]%' THEN 'T-Z'
    ELSE '[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:

```

SELECT
    [ResellerCode],
    [ResellerName]
FROM
    [Reseller]
WHERE
    [ResellerName] LIKE '%' + @Search + '%'
ORDER BY
    [ResellerName]

```

- 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.

```

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.

```

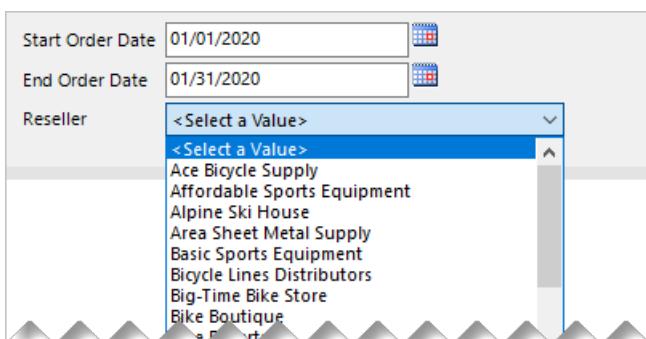
WHERE
    [ResellerName] LIKE SUBSTITUTE(@Search, '%', '*')

```

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:

- 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:

```
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

Next steps

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 \(Report Builder and SSRS\)](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Avoid blank pages when printing paginated reports

3/24/2022 • 2 minutes to read • [Edit Online](#)

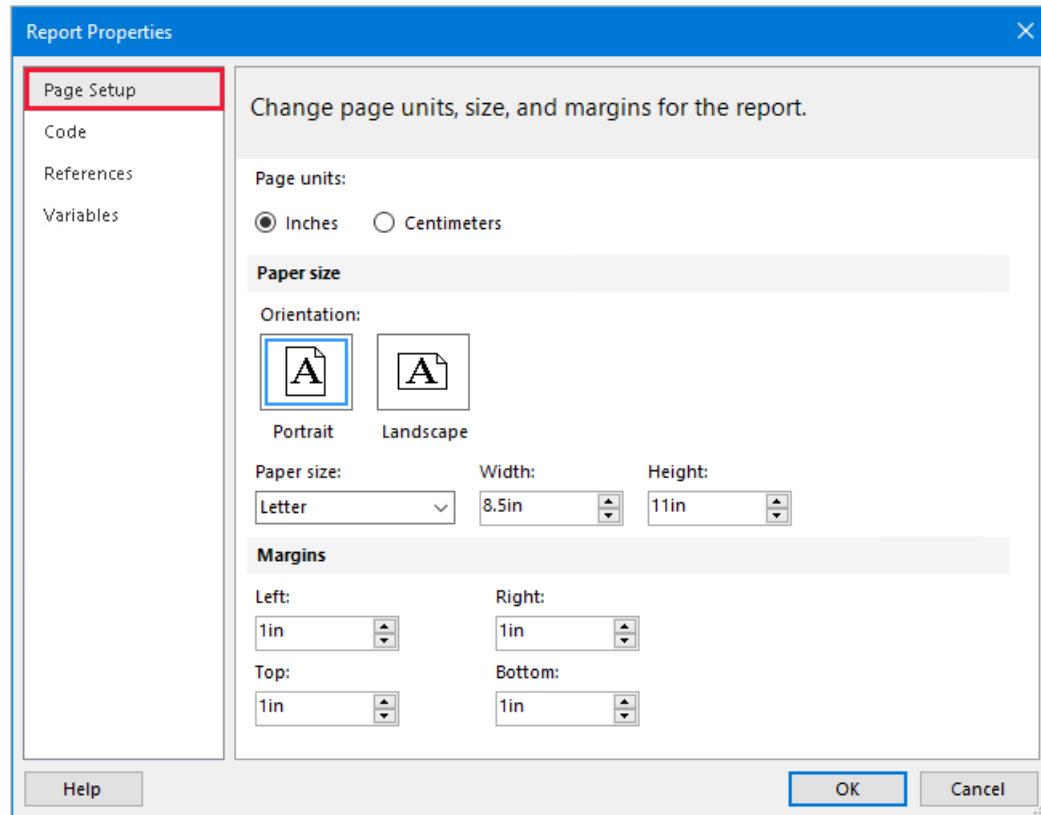
This article targets you as a report author designing Power BI [paginated reports](#). It provides recommendations to help you avoid blank pages when your report is exported to a hard-page format—like PDF or Microsoft Word—or, is printed.

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:

PROPERTY	RECOMMENDATION
Page units	Select the relevant units—_inches or centimeters.
Orientation	Select the correct option—portrait or landscape.

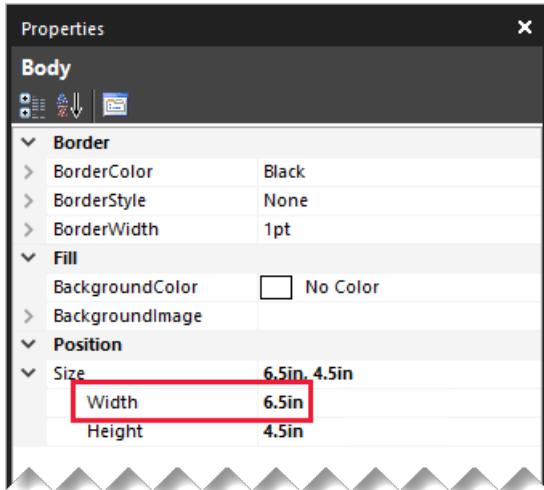
PROPERTY	RECOMMENDATION
Paper size	Select a paper size, or assign custom width and height values.
Margins	Set appropriate values for the left, right, top, and bottom margins.

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.

The screenshot shows a report titled "Salesperson Sales". Inside the report, there is a table with three columns: "Salesperson", "Sales", and "Profit Pct". The "Sales" column contains the expression "[Sales]" and the "Profit Pct" column contains the expression "[Profit_Pct]". Below the table, there is a formula section with the expression "[Aggregate(Sale] [Aggregate(Prof". A red double-headed arrow and an X mark are overlaid on the report area, pointing towards the bottom right corner where the formula is located.

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.

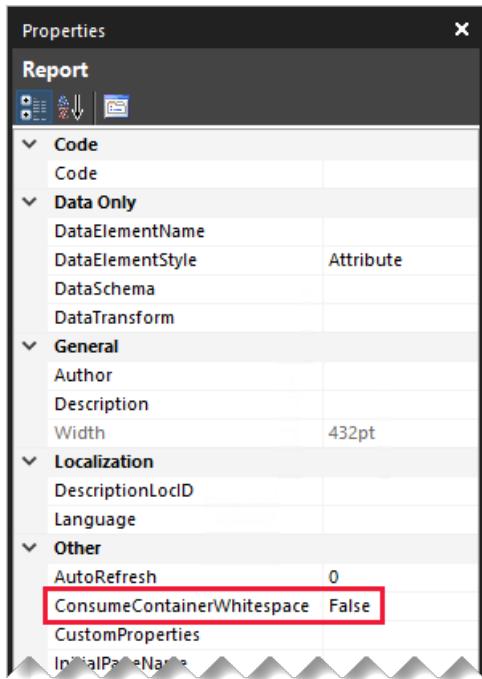
The screenshot shows the "Tablix Properties" dialog box. In the left sidebar, "General" is selected. The main area displays the following settings:

- Name:** Tablix1
- ToolTip:** (empty)
- Dataset name:** dsMain
- Page break options:**
 - Add a page break before
 - Add a page break after
 - Keep together on one page if possible

The "Add a page break after" checkbox is highlighted with a red border.

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](#).

Next steps

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

- [What are paginated reports in Power BI Premium?](#)
- [Pagination in Power BI paginated reports](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Migrate SQL Server Reporting Services reports to Power BI

3/24/2022 • 8 minutes to read • [Edit Online](#)

This article targets SQL Server Reporting Services (SSRS) report authors and Power BI administrators. It provides you with guidance to help you migrate your [Report Definition Language \(RDL\)](#) reports to Power BI.

NOTE

It's only possible to migrate RDL reports. 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 SSRS servers, or disruption to your report users. It's important to understand that no data or reports need to be removed. 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, you should verify that your environment meets certain prerequisites. We'll describe these prerequisites, and also introduce you to a helpful migration tool.

Preparing for migration

As you prepare to migrate your reports to Power BI, first verify that your organization has a [Power BI Premium](#) subscription. This subscription is required to host and run your Power BI paginated reports.

Supported versions

You can migrate SSRS instances running on-premises, or on Virtual Machines hosted by cloud providers, like Azure.

The following list describes the SQL Server versions supported for migration to Power BI:

- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017
- SQL Server 2019

Migration from Power BI Report Server is possible, too.

Migration tool

We recommend 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 (on a Premium capacity)

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, the tool may be improved by Microsoft. 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 SSRS instances. This process involves scanning the network to identify all SQL Server instances in your organization.

You can use the [Microsoft Assessment and Planning Toolkit](#). Also known as the "MAP Toolkit", it discovers and reports on your SQL Server instances, versions, and installed features. It's a powerful inventory, assessment, and reporting tool that can simplify your migration planning process.

Assess

Having discovered your SSRS instances, the goal of the *Assess* phase is to understand any SSRS reports—or server items—that can't be migrated.

Only RDL reports can be migrated from your SSRS servers to Power BI. Each migrated RDL report will become a Power BI paginated report.

The following SSRS item types, however, can't be migrated to Power BI:

- Shared data sources ¹
- Shared datasets ¹
- Resources, like image files
- KPIs (SSRS 2016, or later—Enterprise Edition only)
- Mobile reports (SSRS 2016, or later—Enterprise Edition only)
- Report models (deprecated)
- Report parts (deprecated)

¹ The [RDL Migration Tool](#) automatically converts shared data sources and shared datasets—providing they're using supported data sources.

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](#). Even if your RDL reports can migrate, we recommend you consider modernizing them as Power BI reports, when it makes sense.

If your RDL reports need to retrieve data from *on-premises data sources*, they cannot use single sign-on (SSO). Currently, all data retrieval from these sources will be done by using the security context of the *gateway data source user account*. It's not possible for SQL Server Analysis Services (SSAS) to enforce row-level security (RLS) on a per-user basis.

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](#).

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 SSRS items that won't migrate.

1. Ensure the [Paginated Reports workload](#) is enabled for your Power BI Premium capacity, and that it has sufficient memory.
2. Verify support for your report [data sources](#), and set up a [Power BI Gateway](#) to allow connectivity with any on-premises data sources.
3. Become familiar with Power BI security, and plan [how you'll reproduce your SSRS folders and permissions with Power BI new workspaces](#).
4. Become familiar with Power BI sharing, and plan how you'll distribute content by publishing [Power BI apps](#).
5. Consider using [shared Power BI datasets](#) in place of your SSRS shared data sources.
6. Use [Power BI Desktop](#) to develop mobile-optimized reports, possibly using the [Power KPI custom visual](#) in place of your SSRS mobile reports and KPIs.
7. 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\mblythe*, the built-in field will return something like *m.blythe@adventureworks.com*. You will 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.
8. 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).
9. 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 \(SQL Server complex data types\)](#).
10. Ensure your report authors have [Power BI Report Builder](#) installed, and that later releases can be easily distributed throughout your organization.

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 SSRS instance and the Power BI workspace can manually migrate reports to Power BI. Here are the steps to follow:

1. Open the SSRS 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 Azure AD 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 will fail. You'll be notified of the reasons. You'll then need to revise your report design, and try saving again.

Automated migration

There are two options for automated migration. You can use:

- The RDL Migration Tool
- The publicly available APIs for SSRS and Power BI

The [RDL Migration Tool](#) has already been described in this article.

You can also use the publicly available SSRS 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.

Configure data sources

Once reports have been migrated to Power BI, you'll 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 recommend 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 rendering times in SSRS and Power BI. Check that Power BI reports render in an acceptable time.
3. If Power BI reports fail to render because of insufficient memory, allocate [additional resources to the Power BI Premium capacity](#).
4. For long-rendering reports, consider having Power BI deliver them to your report users as [email subscriptions with report attachments](#).
5. For Power BI reports based on Power BI datasets, 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.

For more information about these issues, including specific steps to understand and mitigate them, see the following articles:

- [Optimizing Premium capacities](#)
- [Monitor Premium capacities within the app](#)

Next steps

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

- [What are paginated reports in Power BI Premium?](#)
- [Data retrieval guidance for paginated reports](#)
- [When to use paginated reports in Power BI](#)
- [Paginated reports in Power BI: FAQ](#)
- [Online course: Paginated Reports in a Day](#)
- [Power BI Premium FAQ](#)
- [RDL Migration Tool](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Tenant settings guidance

3/24/2022 • 6 minutes to read • [Edit Online](#)

This article targets Power BI administrators who are responsible for setting up and configuring the Power BI environment in their organization.

We provide guidance for specific tenant settings that help improve the Power BI experience, or could expose your organization to risk. We recommend you always configure your tenant to align with your organization's policies and processes.

Tenant settings are managed in the [Admin portal](#), and can be configured by a [Power BI service administrator](#). Many tenant settings can restrict capabilities and features to a limited set of users. So, we recommend you first become familiar with the settings to plan the security groups you'll need. You might find that you can apply the same security group to multiple settings.

Improve Power BI experience

Publish "Get Help" information

We encourage you to set up internal Power BI-related sites using [Microsoft Teams](#), or other collaboration platform. These sites can be used to store training documentation, host discussions, make requests for licenses, or respond to help.

If you do so, we recommend you then enable the **Publish "Get Help" information** setting *for the entire organization*. It's found in the **Help and support settings** group. You can set URLs for your:

- Training documentation
- Discussion forum
- Licensing requests
- Help desk

These URLs will become available as links in the Power BI help menu.

NOTE

Supplying the **Licensing requests** URL prevents individual users from buying a Power BI Pro license. Instead, they'll be directed to your internal site with information on how to acquire a license. The setting **Allow users to try Power BI Pro** is enabled by default and separates the purchase and trial experiences. To learn more about how these settings work together, see [Allow users to try Power BI Pro](#).

Help and support settings

▲ Publish "Get Help" information

Users in the organization can go to internal help and support resources from the Power BI help menu.



Training documentation:

Enter URL

Discussion Forum:

Enter URL

Licensing requests:

Enter URL

Help Desk:

Enter URL

Apply to:

- The entire organization
- Specific security groups
- Except specific security groups

For more information, see [Help and support settings](#).

Manage risk

The settings to manage risks can help you establish governance policies in your Power BI tenant. Keep in mind, however, that governance settings are not a security measure. For example, disabling the **Export data** setting removes the feature from the Power BI user interface and helps in this way Power BI users to work in compliance with your organization's governance policies, but it does not prevent determined users from exporting data using other options. From a security viewpoint, a Power BI user with read access to a dataset has the permission to query this dataset and can persist the results regardless of the features available in the Power BI user interface.

Receive email notification service outages or incidents

You can be notified by email if your tenant is impacted by a service outage or incident. This way, you can proactively respond to incidents.

We recommend you enable the **Receive email notification service outages or incidents** setting. It's found in the **Help and support settings** group. Assign one or more *mail-enabled* security groups.

- ▲ Receive email notifications for service outages or incidents

Mail-enabled security groups will receive email notifications if this tenant is impacted by a service outage or incident.



Enter security groups

Information protection

Information protection allows enforcing protection settings—such as encryption or watermarks—when exporting data from the Power BI service.

There are two tenant settings related to information protection. By default, both settings are disabled for the entire organization.

We recommend you enable these settings when you need to handle and protect sensitive data. For more information, see [Data protection in Power BI](#).

Create workspaces

You can restrict users from creating workspaces. This way, you can govern what is created within your organization.

NOTE

Currently there's a transition period between the old workspace experience and the new. This tenant setting applies only to the new experience.

The **Create workspaces** setting is enabled by default for the entire organization. It's found in the **Workspace settings** group.

We recommend you assign one or more security groups. These groups can be granted *or denied* permission to create workspaces.

Be sure to include instructions in your documentation letting users (who don't have workspace creation rights) know how they can request a new workspace.

Workspace settings

▲ Create workspaces (new workspace experience)

Users in the organization can create app workspaces to collaborate on dashboards, reports, and other content.



The permission to create workspaces in the new workspaces experience preview is currently controlled by the permission to create groups in Office 365. By clicking Apply, the values below will control which users can create workspaces in the new workspaces experience preview. [Learn more](#).

Apply to:

- The entire organization
- Specific security groups
- Except specific security groups

Share content with external users

Users can share reports and dashboards with people outside your organization.

The **Share content with external users** setting is enabled by default for the entire organization. It's found in the **Export and sharing settings** group.

We recommend you assign one or more security groups. These groups can be granted *or denied* permission to share content with external users.

Export and sharing settings

▲ Share content with external users

Enabled for the entire organization

Users in the organization can share dashboards and reports with users outside the organization.



Apply to:

- The entire organization
- Specific security groups
- Except specific security groups

Apply

Cancel

Publish to web

The [publish to web](#) feature allows publishing public reports on the web. If used inappropriately, there's risk that confidential information could be made available live on the web.

The **Publish to web** setting is enabled by default for the entire organization, but restricting the ability for non-admin users to create embed codes. It's found in the **Export and sharing settings** group.

If enabled, we recommend you assign one or more security groups. These groups can be granted *or denied* permission to publish reports.

Further, there's an option to choose how your embed codes work. By default, it's set to **Only allow existing codes**. It means users will be asked to contact a Power BI admin to create an embed code.

▲ Publish to web ⓘ

People in your org can publish public reports on the web. Publicly published reports don't require authentication to view them.

Go to [Embed Codes](#) in the admin portal to review and manage public embed codes. If any of the codes contain private or confidential content remove them.

Review embed codes regularly to make sure no confidential information is live on the web. [Learn more about Publish to web](#)



Choose how embed codes work

- Only allow existing codes
- Allow existing and new codes

Apply to:

- The entire organization
 - Specific security groups
- Except specific security groups

We also recommend you review [publish to web embed codes](#) regularly. Remove codes if they result in the publication of private or confidential information.

Export data

You can restrict users from exporting data from dashboard tiles or report visuals.

The **Export data** setting is enabled by default for the entire organization. It's found in the **Export and sharing settings** group.

We recommend you assign one or more security groups. These groups can be granted *or denied* permission to publish reports.

IMPORTANT

Disabling this setting also restricts the use of the [Analyze in Excel](#) and Power BI service [live connection](#) features.

▲ Export data
Enabled for the entire organization

Users in the organization can export data from a tile or visualization. This also controls the Analyze in Excel and Power BI Service Live Connect features.

 Enabled

Apply to:

The entire organization
 Specific security groups
 Except specific security groups

Apply **Cancel**

NOTE

If users allow users to export data, you can add a layer of protection by enforcing [data protection](#). When configured, unauthorized users will be blocked from exporting content with sensitivity labels.

Allow external guest users to edit and manage content in the organization

It's possible that external guest users can edit and manage Power BI content. For more information, see [Distribute Power BI content to external guest users with Azure AD B2B](#).

The **Allow external guest users to edit and manage content in the organization** setting is disabled by default for the entire organization. It's found in the **Export and sharing settings** group.

If you need to authorize external users to edit and manage content, we recommend you assign one or more security groups. These groups can be granted *or denied* permission to publish reports.

▲ Allow external guest users to edit and manage content in the organization

The specified guest users in the organization can edit and manage content in workspaces in the organization. They receive the ability to browse content and request access to content. [Learn more](#).



Apply to:

- The entire organization
 Specific security groups
 Except specific security groups

Apply

Cancel

Developer settings

There are two tenant settings related to [embedding Power BI content](#). They are:

- Embed content in apps (enabled by default)
- Allow service principals to user Power BI APIs (disabled by default)
- Block ResourceKey Authentication (disabled by default)*

If you have no intention of using the developer APIs to embed content, we recommend you disable them. Or, at least configure specific security groups that would be doing this work.

** This setting applies to streaming and PUSH datasets. If disabled, users will not be allowed send data to streaming and PUSH datasets using the API with a resource key.*

Developer settings

- ▶ Embed content in apps
Enabled for the entire organization
- ▶ Allow service principals to use Power BI APIs
Enabled for the entire organization
- ▶ Block ResourceKey Authentication
Disabled for the entire organization

Next steps

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

- [What is Power BI administration?](#)
- [Administering Power BI in the admin portal](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

On-premises data gateway sizing

3/24/2022 • 4 minutes to read • [Edit Online](#)

This article targets Power BI 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).

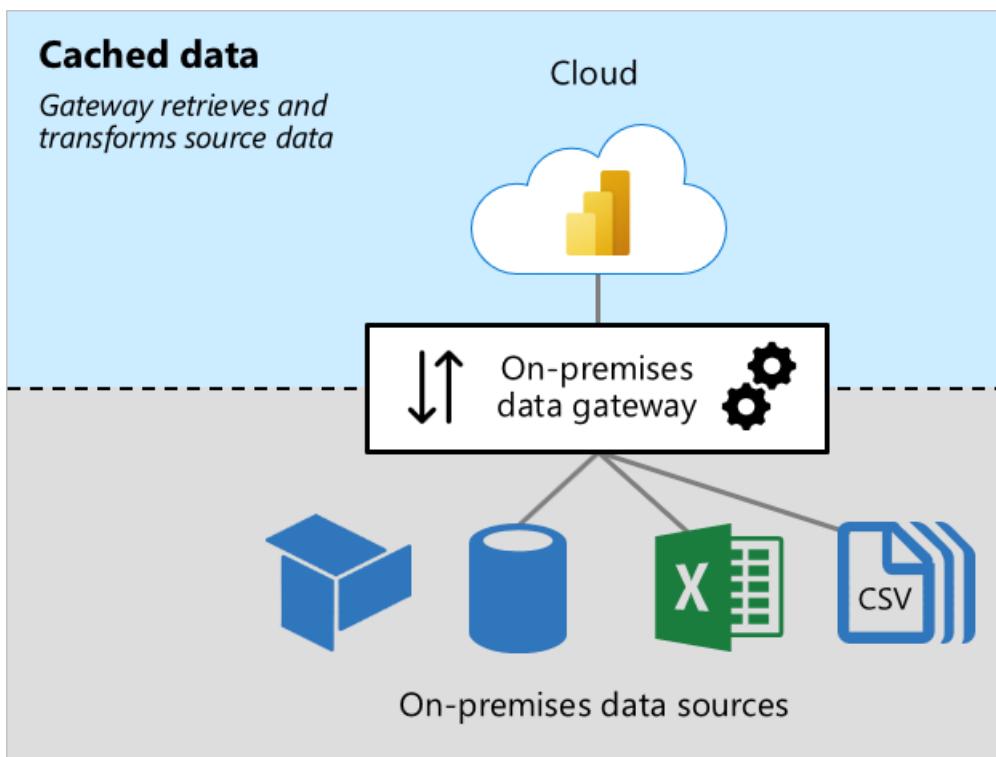
Gateway workloads

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 datasets. 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.

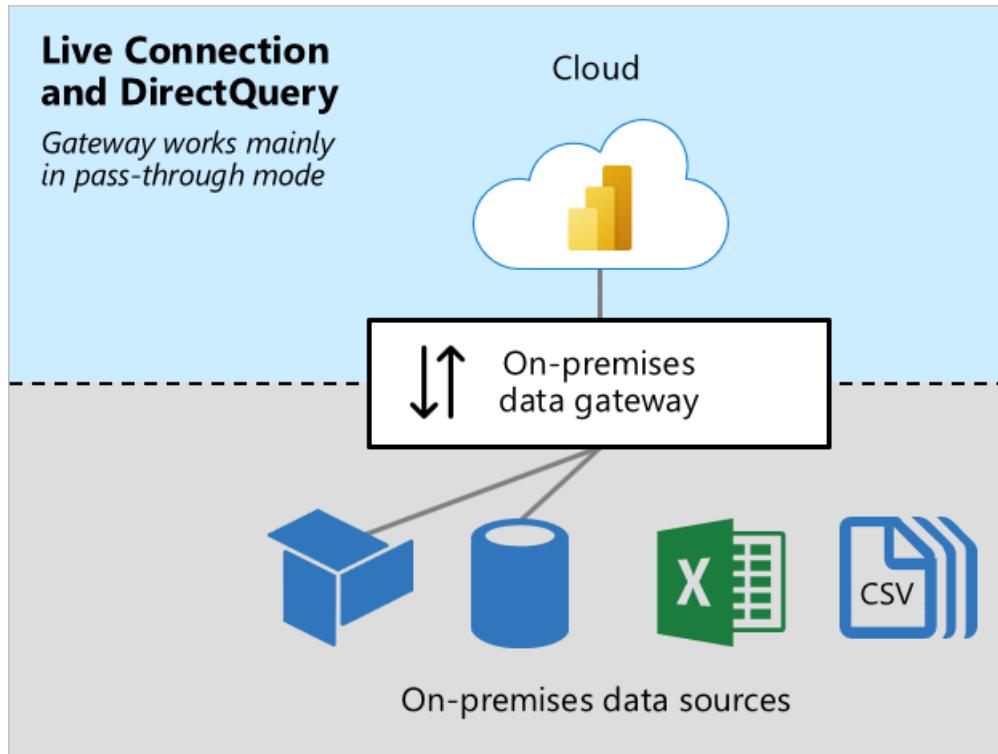
- For more information about Live Connection, see [Datasets in the Power BI service \(Externally-hosted\)](#)

models).

- For more information about DirectQuery, see [Dataset 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 dataset 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 datasets 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 [What is Power BI Premium? \(Capacity nodes\)](#).

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](#).
- Configure [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](#).

Dataset design and settings

Dataset design, and their settings, can impact on gateway workloads. To reduce gateway workload, you can consider the following actions.

For Import datasets:

- Configure less frequent data refresh
- Configure [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 datasets:

- 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 datasets:

- Especially when dynamic RLS is enforced, restrict dashboard cache update frequency

Next steps

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

- [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 Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Monitor report performance in Power BI

3/24/2022 • 3 minutes to read • [Edit Online](#)

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

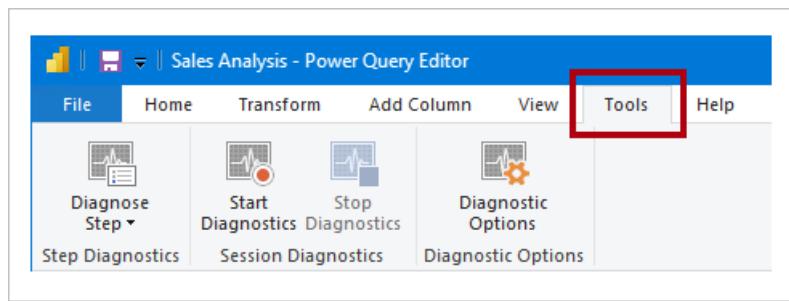
The Performance Analyzer cannot be used to monitor Premium Per User (PPU) activities or capacity.

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.

NOTE

Query Diagnostics is currently a preview feature, and so you must enable it in *Options and Settings*. Once enabled, its commands are available in the Power Query Editor window, on the **Tools** ribbon tab.



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 not supported. Changes to the data model 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 Gen2 capacity with the help of the [Gen2 metrics app](#).

To monitor the Power BI Premium original version capacities, use the [Power BI Premium Metrics app](#).

Next steps

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

- [Query Diagnostics](#)
- [Performance Analyzer](#)

- Troubleshoot report performance in Power BI
- Power BI Premium Metrics app
- Questions? Try asking the Power BI Community
- Suggestions? Contribute ideas to improve Power BI

Troubleshoot report performance in Power BI

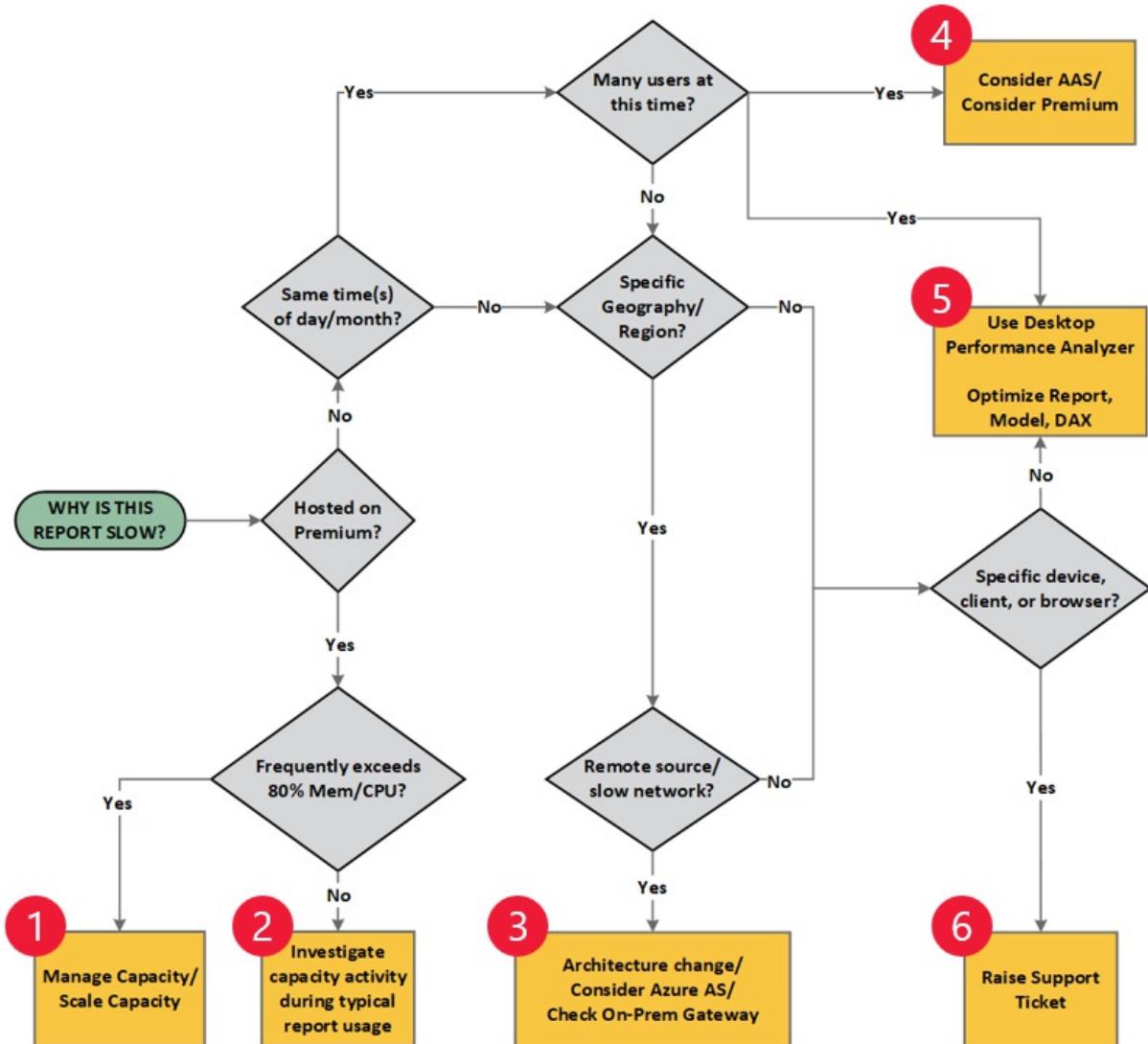
3/24/2022 • 2 minutes to read • [Edit Online](#)

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, slow reports can also be identified by monitoring the [Power BI Premium Metrics app](#). This app helps you to monitor the health and capacity of your Power BI Premium subscription.

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:

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
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 [Power BI Premium Metrics app](#) to determine if the report-hosting capacity frequently exceeds capacity resources. It's the case for CPU when it frequently exceeds 80%. For memory, it's when the [active memory metric](#) exceeds 50. 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 dataset 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).

Next steps

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

- [Power BI guidance](#)
- [Monitoring report performance](#)
- [Performance Analyzer](#)
- [Power BI adoption roadmap](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Deployment pipelines best practices

3/24/2022 • 9 minutes to read • [Edit Online](#)

This article provides guidance for BI creators who are managing their content throughout its lifecycle. It focuses on leveraging deployment pipelines as a BI content lifecycle management tool.

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 the deployment pipelines test stage, to test your environment.
- **Production** - Utilize the deployment pipelines production stage when making your content available for consumption.

Content preparation

Prepare your content for on-going management throughout its lifecycle. Make sure you review the information in this section, before you do any of the following:

- Release your content to production
- Start using a deployment pipeline for a specific workspace
- Publish your work

Treat each workspace as a complete package of analytics

Ideally, a workspace should contain a complete view of one aspect (such as department, business unit, project, or vertical) in your organization. This makes it easier to manage permissions for different users, and allows content releases for the entire workspace to be controlled according to a planned schedule.

If you're using [centralized datasets](#) that are used across the organization, we recommend that you create two types of workspaces:

- **Modeling and data workspaces** - These workspaces will contain all the centralized datasets
- **Reporting workspaces** - These workspaces will contain all dependent reports and dashboards

Plan your permission model

A deployment pipeline is a Power BI object, with its own [permissions](#). In addition, the pipeline contains workspaces, that have their own permissions.

To implement a secure and easy workflow, plan who gets access to each part of the pipeline. Some of the considerations to take into account are:

- Who should have access to the pipeline?
- 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 will oversee deployment to the production stage?

- Which workspace are you assigning?
- 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 better not to overload it with queries generated by BI creators for their development or test datasets. Build separate databases for development and testing. This helps protect production data, and doesn't overload the development database with the entire volume of production data, which can slow down things.

NOTE

If your organization is using [shared centralized datasets](#), you can skip this recommendation.

Use parameters in your model

As you can't edit datasets data sources in Power BI service, we recommend using [parameters](#) to store connection details such as instance names and database names, instead of using a static connection string. This allows you to manage the connections through the Power BI service web portal, or [using APIs](#), at a later stage.

In deployment pipelines, you can configure parameter rules to set specific values for the development, test, and production stages.

If you don't use parameters for your connection string, you can define data source rules to specify a connection string for a given dataset. However, in deployment pipelines, this isn't supported for all data sources. To verify that you can configure rules for your data source, see [deployment rules limitations](#).

Parameters have additional uses, such as making changes to queries, filters, and the text displayed in the report.

Development

This section provides guidance for working with the deployment pipelines development stage.

Use Power BI Desktop to edit your reports and datasets

Consider Power BI Desktop as your local development environment. Power BI Desktop allows you to try, explore, and review updates to your reports and datasets. Once the work is done, you can upload your new version to the development stage. Due to the following reasons, it's recommended to edit PBIX files in the Desktop (and not in Power BI service):

- It is easier to collaborate with fellow creators on the same PBIX file, if all changes are being done on the same tool.
- Making online changes, downloading the PBIX file, and then uploading it again, creates reports and datasets duplication.
- You can use version control to keep your PBIX files up to date.

Version control for PBIX files

If you want to manage the version history of your reports and datasets, use [Power BI's autosync with OneDrive](#). This will keep your files updated with the latest version. It will also enable you to retrieve older versions if needed.

NOTE

Use auto-sync with OneDrive (or any other repository) only with the PBIX files in the deployment pipeline's development stage. Do not sync PBIX files into the deployment pipeline's test and production stages. This will cause problems with deploying content across the pipeline.

Separate modeling development from report and dashboard development

For enterprise scale deployments, it's recommended to separate dataset development, and the development of reports and dashboards. To promote changes to only a report or a dataset, use the deployment pipelines selective deploy option.

This approach should start from Power BI Desktop, by creating a separate PBIX file for datasets and reports. For example, you can create a dataset PBIX file and uploaded it to the development stage. Later, your report authors can create a new PBIX only for the report, and [connect it to the published dataset](#) using a live connection. This technic allows different creators to separately work on modeling and visualizations, and deploy them to production independently.

With [shared datasets](#), you can also use this method across workspaces.

Manage your models using XMLA read/write capabilities

Separating modeling development from report and dashboard development, allows you to use advanced capabilities such as source control, merging diff changes, and automated processes. These changes should be done in the development stage, so that finalized content can be deployed to the test and production stages. This allows changes to go through a unified process with other dependent items, before they're deployed to the production stage.

You can separate modeling development from visualizations, by managing a [shared dataset](#) in an external workspace, using XMLA r/w capabilities. The shared dataset can connect to multiple reports in various workspaces that are managed in multiple pipelines.

Test

This section provides guidance for working with the deployment pipelines test stage.

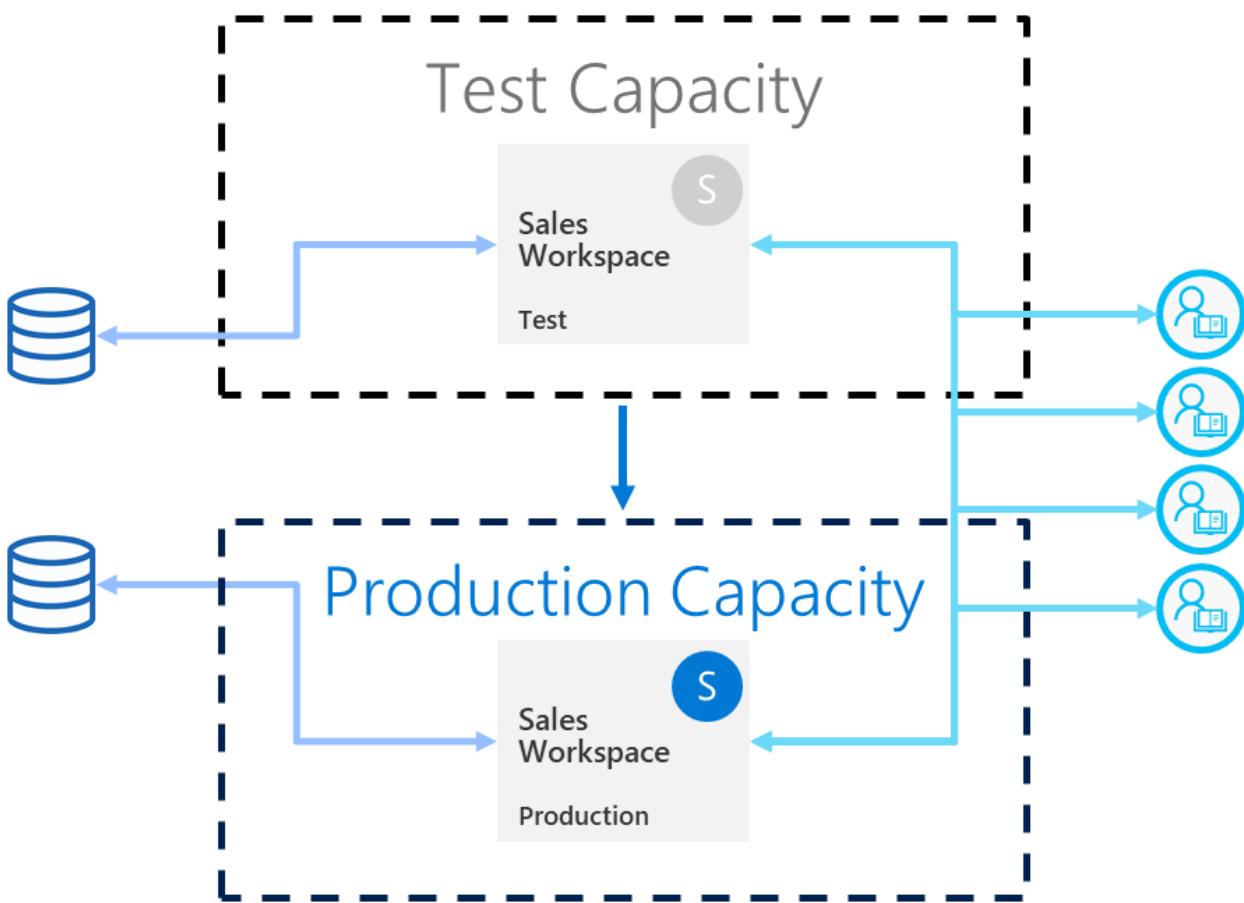
Simulate your production environment

Other than verifying that new reports or dashboards look alright, it's also important to see how they perform from an end user's perspective. The deployment pipelines test stage, allows you to simulate a real production environment for testing purposes.

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, this can make production unstable during load testing. To avoid unstable production, use another capacity similar in resources to the production capacity, for testing. To avoid extra costs, you can use [Azure A capacities](#) to 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 recommended 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.

Controlling the amount of data you import from your data source, is useful if you're using a production data source in the test stage. To do this, add a parameter to your data source query in Power BI Desktop. Use parameter rules to control the amount of imported data, or edit the parameter's value. You can also use this approach if you don't want to overload your capacity.

Measure performance

When you simulate a production stage, [check the report load and the interactions](#), and find out if the changes you made impact them.

You also need to [monitor the load on the capacity](#), so that you can catch extreme loads before they reach production.

NOTE

It's recommended to monitor capacity loads again, after deploying updates to the production stage.

Check related items

Related times can be affected by changes to datasets or reports. During testing, verify that your changes don't impact or break the performance of existing items, which can be dependent on the updated ones.

You can easily find the related items using the workspace [lineage view](#).

Test your app

If you are distributing content to your end users through an app, review the app's new version, before it's in production. As each deployment pipeline stage has its own workspace, you can easily publish and update apps

for development and test stages. This will allow you to test the app from an end user's point of view.

IMPORTANT

The deployment process does not include updating the app content or settings. To apply changes to content or settings, you need to manually update the app in the required pipeline stage.

Production

This section provides guidance to the deployment pipelines production stage.

Manage who can deploy to production

As 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. This can be managed using production [workspace permissions](#).

To deploy content between stages, users need to have either member or admin permissions for both stages. Make sure that only the people you want deploying to production, will have production workspace permissions. Other users can have production workspace contributor or viewer roles. They will be able to see content from within the pipeline but won't be able to deploy.

In addition, you should limit access to the pipeline by only enabling pipeline 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. Once deployment rules are applied, deployments can run while you have the assurance that end users will see the relevant info without disturbance.

Make sure that you set production deployment rules for data sources and parameters defined in the dataset.

Update the production app

Deployment in a pipeline updates the workspace content, but it doesn't update the associated app automatically. If you're using an app for content distribution, don't forget to update the App after deploying to production, so that end users will immediately be able to use the latest version.

Quick fixes to content

In case there are bugs in production that require a quick fix, don't be tempted to either upload a new PBIX version directly to the production stage, or make an online change in Power BI service. Deploying backwards to test and development stages isn't possible when there's already content in those stages. Furthermore, deploying a fix without testing it first is bad practice. Therefore, the correct way to treat this problem, is to implement the fix in the development stage, and push it to the rest of the deployment pipeline stages. This allows checking that the fix works, before deploying it to production. Deploying across the pipeline takes only a few minutes.

Next steps

[Introduction to deployment pipelines](#)

[Get started with deployment pipelines](#)

[Assign a workspace to a pipeline stage](#)

[Understand the deployment pipelines process](#)

[Automate your deployment pipeline using APIs and DevOps](#)

Deployment pipelines troubleshooting

Access the Power BI activity log

3/24/2022 • 5 minutes to read • [Edit Online](#)

This article targets Power BI administrators who need to access the Power BI activity log. Because there's no user interface to search the activity log yet, you'll need to use the Power BI REST API and management cmdlets.

NOTE

This article doesn't introduce or describe the Power BI activity log. For more information, see [Track user activities in Power BI](#).

PowerShell sample

A PowerShell sample is available to help you learn how to filter and retrieve Power BI activity log events. All code fragments and scenarios are annotated with how-to instructions and common gaps or issues to watch out for.

Two scenarios are covered:

- Retrieve a list of users for specific app.
- Retrieve a list of users for direct report sharing.

NOTE

You'll need to have some familiarity with the [Power BI Admin API](#) and [Power BI PowerShell modules](#). PowerShell modules must be installed prior to executing these script blocks. For more information, see [Track user activities in Power BI](#).

There can be a lag of up to 30 minutes to retrieve Power BI events.

To use the sample, you must meet the following requirements:

- Install the [Power BI PowerShell modules](#).
- The user of the PowerShell script must sign in using the [Connect-PowerBIServiceAccount cmdlet](#) and enter Power BI admin credentials when prompted. Admin privileges are required to use the Activity Log API.

IMPORTANT

Users without admin privileges won't be able to execute code fragments in the sample script.

```
# Written by Sergei Gundorov; v1 development started on 07/08/2020
#
# Intent: 1. Address common friction points and usage issues and questions related to the
#           events generated by Power BI Service that are stored in the activity log.
#           2. Provide boiler plate code for capturing all 30 days of available data.
#           3. Power BI admin privileges are required to use the Activity Log API.
#
# Use:     Sign in to the Power BI service with admin privileges and execute specific segment one at a time.

# IMPORTANT: Use Connect-PowerBIServiceAccount to connect to the service before running individual code
# segments.

# IMPORTANT: $day value may need to be adjusted depending on where you're located in the world relative to
# UTC.
#           The Power BI activity log records events using UTC time; so add or subtract days according to
```

```

your global location.

# SCENARIO: Sample code fragment to retrieve a limited number of attributes for specific events for specific user report viewing activity.
# You need to get user's Azure Active Directory (AAD) object ID. You can use this Azure AD cmdlet:
https://docs.microsoft.com/powershell/module/azuread/get-azureaduser?view=azureadps-2.0

# Dates need to be entered using ISO 8601 format; adjust dates to span no more than 24 hours.
$a=Get-PowerBIActivityEvent -StartTime '2020-06-23T19:00:00.000' -EndTime '2020-06-23T20:59:59.999'
-ActivityType 'ViewReport' -User [USER AAD ObjectId GUID] | ConvertFrom-Json

# You can use any attribute value to filter results further. For example, a specific event request Id can be used to analyze just one specific event.
$a | Select RequestId, ReportName, WorkspaceName |where {($_.RequestId -eq '[RequestId GUID of the event]')}

# SCENARIO: Retrieve a list of users for specific app.
# The user list can be partially derived (based on last 30 days of available activity) by combining data for two events: CreateApp and UpdateApp.
# Both events will contain OrgAppPermission property that contains app user access list.
# Actual app installation can be tracked using InstallApp activity.
# Run each code segment separately for each event.

# Iterate through 30 days of activity CreateApp.
$day=Get-date

for($s=0; $s -le 30; $s++)
{
    $periodStart=$day.AddDays(-$s)
    $base=$periodStart.ToString("yyyy-MM-dd")

    write-host $base

    $a=Get-PowerBIActivityEvent -StartTime ($base+'T00:00:00.000') -EndTime ($base+'T23:59:59.999')
    -ActivityType 'CreateApp' -ResultType JsonString | ConvertFrom-Json
    $c=$a.Count

    for($i=0 ; $i -lt $c; $i++)
    {
        $r=$a[$i]
        Write-Host "App Name `t: $($r.ItemName)"
        ` "WS Name `t: $($r.WorkSpaceName)"
        ` "WS ID `t`t: $($r.WorkspaceId)"
        ` "Created `t: $($r.CreationTime)"
        ` "Users `t`t: $($r.OrgAppPermission) `n"
    }
}

# Iterate through 30 days of activity UpdateApp.
$day=Get-date

for($s=0; $s -le 30; $s++)
{
    $periodStart=$day.AddDays(-$s)
    $base=$periodStart.ToString("yyyy-MM-dd")

    write-host $base

    $a=Get-PowerBIActivityEvent -StartTime ($base+'T00:00:00.000') -EndTime ($base+'T23:59:59.999')
    -ActivityType 'UpdateApp' -ResultType JsonString | ConvertFrom-Json
    $c=$a.Count

    for($i=0 ; $i -lt $c; $i++)
    {
        $r=$a[$i]
        Write-Host "App Name `t: $($r.ItemName)"
        ` "WS Name `t: $($r.WorkSpaceName)"
        ` "WS ID `t`t: $($r.WorkspaceId)"
        ` "Updated `t: $($r.CreationTime)"
        ` "Users `t`t: $($r.OrgAppPermission) `n"
    }
}

```

```

    }

# Iterate through 30 days of activity InstallApp.
$day=Get-date

for($s=0; $s -le 30; $s++)
{
    $periodStart=$day.AddDays(-$s)
    $base=$periodStart.ToString("yyyy-MM-dd")

    write-host $base

    $a=Get-PowerBIActivityEvent -StartTime ($base+'T00:00:00.000') -EndTime ($base+'T23:59:59.999')
    -ActivityType 'InstallApp' -ResultType JsonString | ConvertFrom-Json
    $c=$a.Count

    for($i=0 ; $i -lt $c; $i++)
    {
        $r=$a[$i]
        Write-Host "App Name `t: $($r.ItemName)"
        ` "Installed `t: $($r.CreationTime)"
        ` "User `t`t: $($r.UserId) `n"
    }
}

# SCENARIO: Retrieve a list of users for direct report sharing.
# This logic and flow can be used for tracing direct dashboard sharing by substituting activity type.
# Default output is formatted to return the list of users as a string. There is commented out code block to
get multi-line user list.
# IMPORTANT: Removal of a user or group from direct sharing access list event is not tracked. For this
reason, the list may be not accurate.
# IMPORTANT: If the user list contains a GUID instead of a UPN the report was shared to a group.
#           Group name and email can be obtained using Azure AD cmdlets using captured ObjectId GUID.

# Iterate through 30 days of activity ShareReport.
$day=Get-date

for($s=0; $s -le 30; $s++)
{
    $periodStart=$day.AddDays(-$s)
    $base=$periodStart.ToString("yyyy-MM-dd")

    #write-host $base

    $a=Get-PowerBIActivityEvent -StartTime ($base+'T00:00:00.000') -EndTime ($base+'T23:59:59.999')
    -ActivityType 'ShareReport' -ResultType JsonString | ConvertFrom-Json
    $c=$a.Count

    for($i=0 ; $i -lt $c; $i++)
    {
        $r=$a[$i]

        Write-Host "Rpt Name `t: $($r.ItemName)"
        ` "Rpt Id `t: $($r.ArtifactId)"
        ` "WS Name `t: $($r.WorkSpaceName)"
        ` "WS ID `t`t: $($r.WorkspaceId)"
        ` "Capacity `t: $($r.CapacityId)"
        ` "SharedOn `t: $($r.CreationTime.Replace('T',' ').Replace('Z',''))"
        ` "User `t`t: $($r.UserId)"
        # NOTE: $_.RecipientEmail + $_.RecipientName or +$_._ObjectId is the case for group
sharing
        #           can never happen both at the same time in the same JSON record
        ` "Shared with`t: $($r.SharingInformation)| % {$_._RecipientEmail + $_._ObjectId + '[' +
$_._ResharePermission +']}'})"
    }
}

#OPTIONAL: Formatted output for SharingInformation attribute
#$sc= $r.SharingInformation.Count

```

```
#Write-Host "Shared with`t:"  
#for($j=0;$j -lt $sc;$j++)  
#{  
#    Write-Host "`t`t`t $($r.SharingInformation[$j].RecipientEmail)" -NoNewline  
#    Write-Host $r.SharingInformation[$j].ObjectId -NoNewline  
#    Write-Host ('[' + $r.SharingInformation[$j].ResharePermission +']')  
#}  
  
Write-host ""  
}  
}
```

Next steps

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

- [Track user activities in Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Power BI migration overview

3/24/2022 • 9 minutes to read • [Edit Online](#)

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:

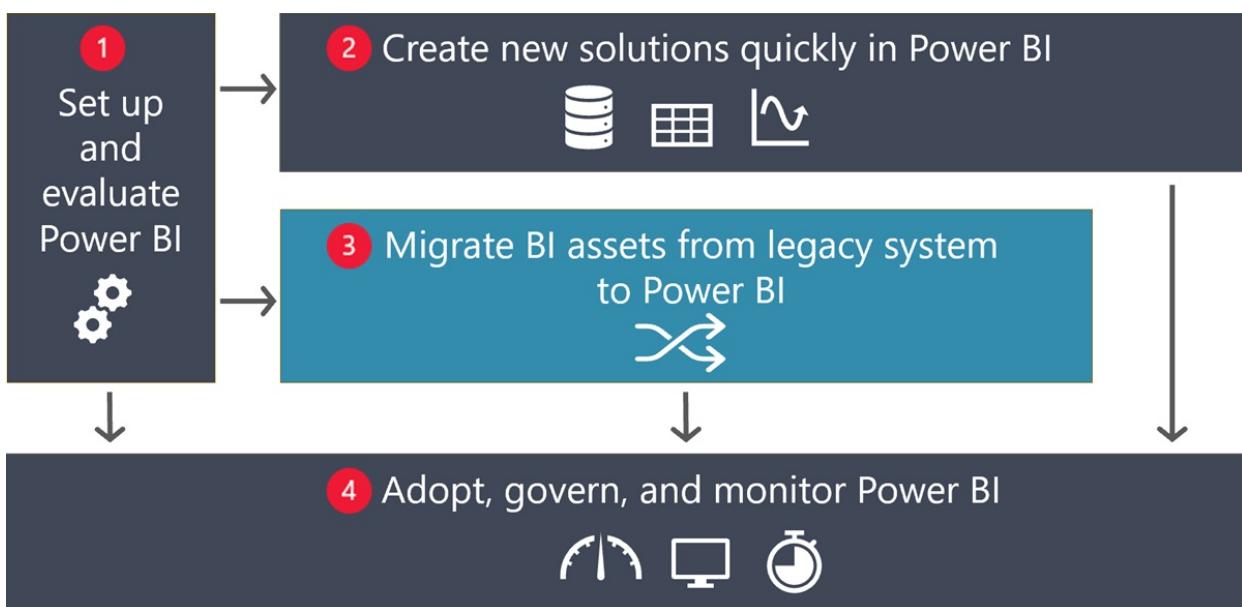
1. Power BI migration overview (this article)
2. [Prepare to migrate to Power BI](#)
3. [Gather requirements to migrate to Power BI \(Stage 1\)](#)
4. [Plan deployment to migrate to Power BI \(Stage 2\)](#)
5. [Conduct proof of concept to migrate to Power BI \(Stage 3\)](#)
6. [Create content to migrate to Power BI \(Stage 4\)](#)
7. [Deploy to Power BI \(Stage 5\)](#)
8. [Learn from customer Power BI migrations](#)

NOTE

It's also recommended that you thoroughly read the [Power BI 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 may apply for national 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.



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.

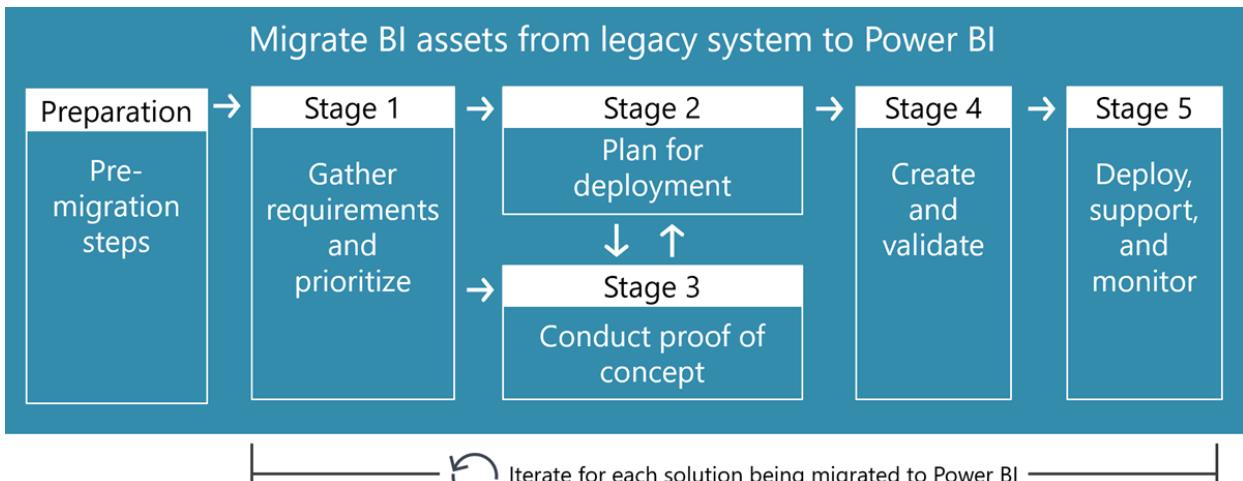
IMPORTANT

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 may 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 may 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 may be 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) may occur in parallel with this stage. Even while creating the solution (in Stage 4), additional information may 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 may 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 may 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 may 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 artifacts 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 may 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 may 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.

Next steps

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:

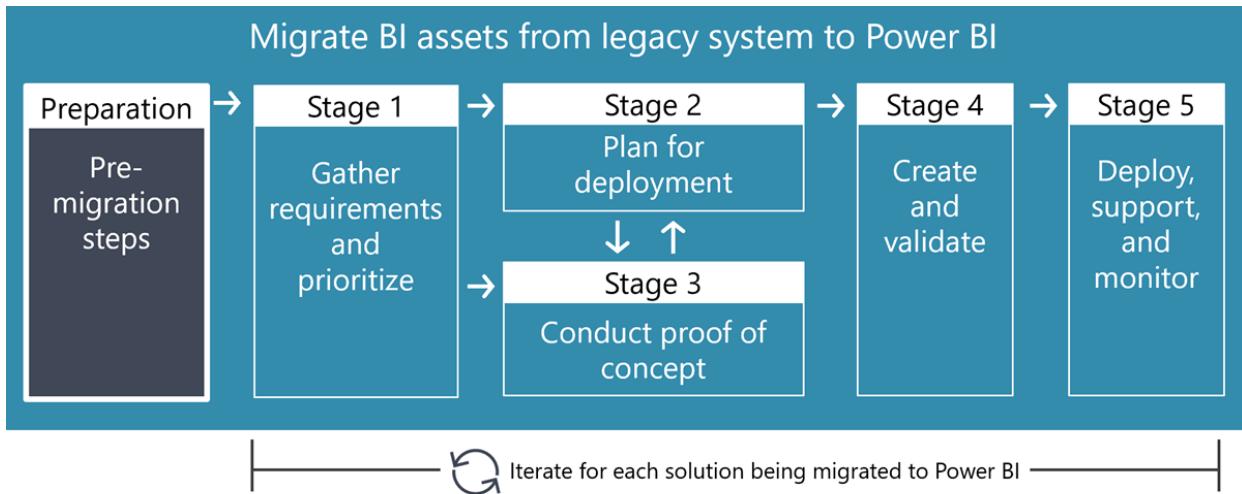
- [Power BI adoption roadmap](#)
- [Power BI implementation planning](#)
- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- [Migrate SSRS reports to Power BI](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Prepare to migrate to Power BI

3/24/2022 • 7 minutes to read • [Edit Online](#)

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 may 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 [Power BI adoption](#) and where Power BI fits into the overall BI strategy for the organization.
- How the Power BI 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:

- [Power BI 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 [dataset 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 datasets](#) 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 artifacts and business-managed BI artifacts.
- 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 may 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 Azure Active Directory.
- Define [Power BI administrators](#).
- Procure and assign initial [user licenses](#).
- Configure and review [Power BI tenant settings](#).
- Setup [workspace roles](#) and assign access to Azure Active Directory security groups and users.
- Configure an initial [data gateway](#) cluster—with a plan to update regularly.
- Procure initial [Premium capacity license](#) (if applicable).
- Configure [Premium capacity workloads](#)—with a plan to manage on an ongoing basis.

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 may benefit significantly from cost savings, whereas migrating a different legacy BI solution may 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 may 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 may 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.

Next steps

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:

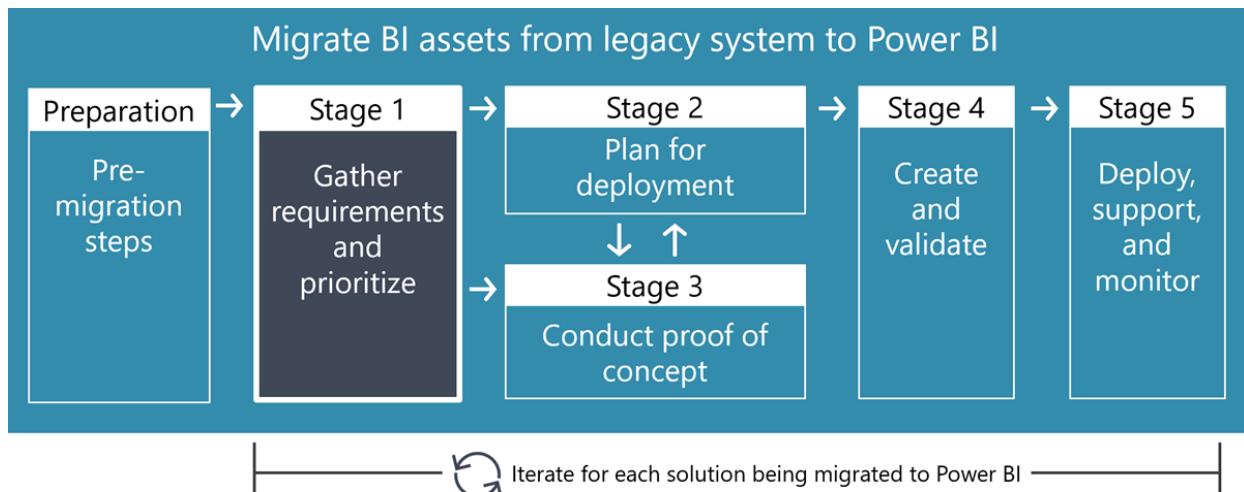
- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Gather requirements to migrate to Power BI

3/24/2022 • 8 minutes to read • [Edit Online](#)

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 [Power BI 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 artifacts, 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 be 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 may 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 may 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 may be on-demand, interactive execution, embedded within a custom application, or delivery on a schedule using an e-mail subscription. There may 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 printing, exporting, or pixel-perfect 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 may possibly need up-front because they believe it may be 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 datasets, 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 dataset 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 datasets, 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 datasets](#), 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 datasets. Dataflows can also significantly reduce the load on source systems because the data is retrieved less often—multiple datasets 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 may consider include:

- **Consolidation of reports:** Similar reports may 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 datasets 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 dataset 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 may include specific requirements. When prioritizing the initial set of BI artifacts ready for migration, reports and data should be considered collectively as well as independently of each other.

Identify high priority reports, which may 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 may include data that:

- Contains critical data elements.
- Is common organizational data that serves many use cases.
- May be used to create a shared dataset for reuse by reports and many report authors.
- Involves a reasonable level of complexity (to improve chances of success when in the initial migration iterations).

Next steps

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:

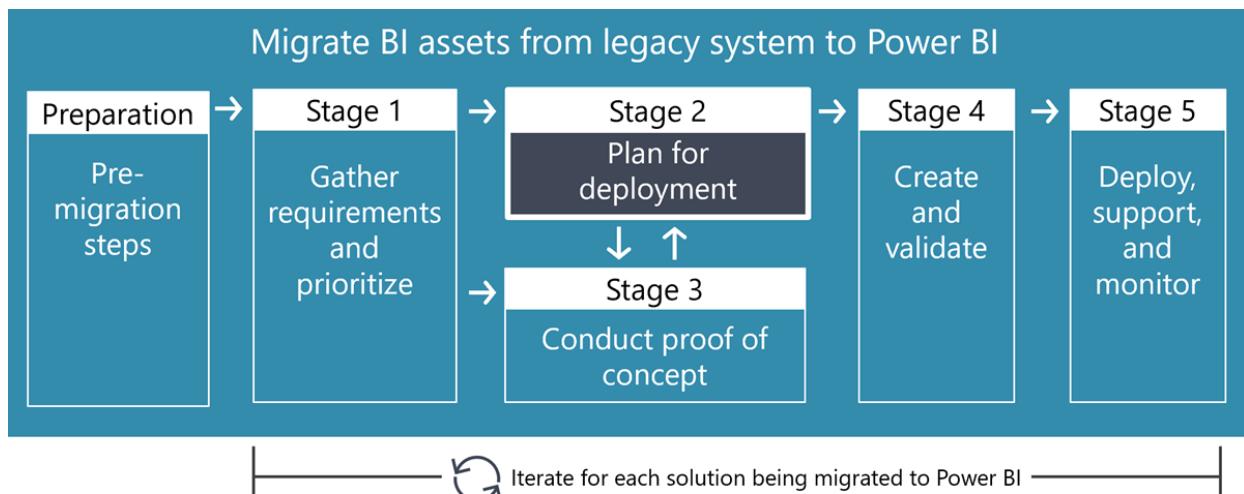
- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Plan deployment to migrate to Power BI

3/24/2022 • 6 minutes to read • [Edit Online](#)

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](#)) may occur in parallel with deployment planning. Even while creating the solution (described in [Stage 4](#)), additional information may 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.

For more information about architectural considerations, see [Section 3](#) of the [Planning a Power BI enterprise deployment whitepaper](#).

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 datasets. 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, [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 Azure Active Directory and Microsoft 365 groups 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 may need to work with your Azure Active Directory administrator and your Power BI administrator to configure [guest user access](#).

TIP

Consider creating a workspace for a specific business activity or project. You may 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 if other content may 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 dataset? This capability can be enabled by assigning dataset [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

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 datasets.
- Support for more frequent data refreshes.
- Support for using the full feature set of dataflows.
- Enterprise features, including deployment pipelines and the XMLA endpoint.
- Support for paginated reports (when the workload is enabled).

Determine data acquisition method

The data required by a report may influence several decisions. Questions to ask include:

- Can an existing Power BI [shared dataset](#) be used, or is the creation of a new Power BI dataset appropriate for this solution?
- Does an existing shared dataset 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 datasets?
- 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 Business 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 dataset authors 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).

Next steps

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:

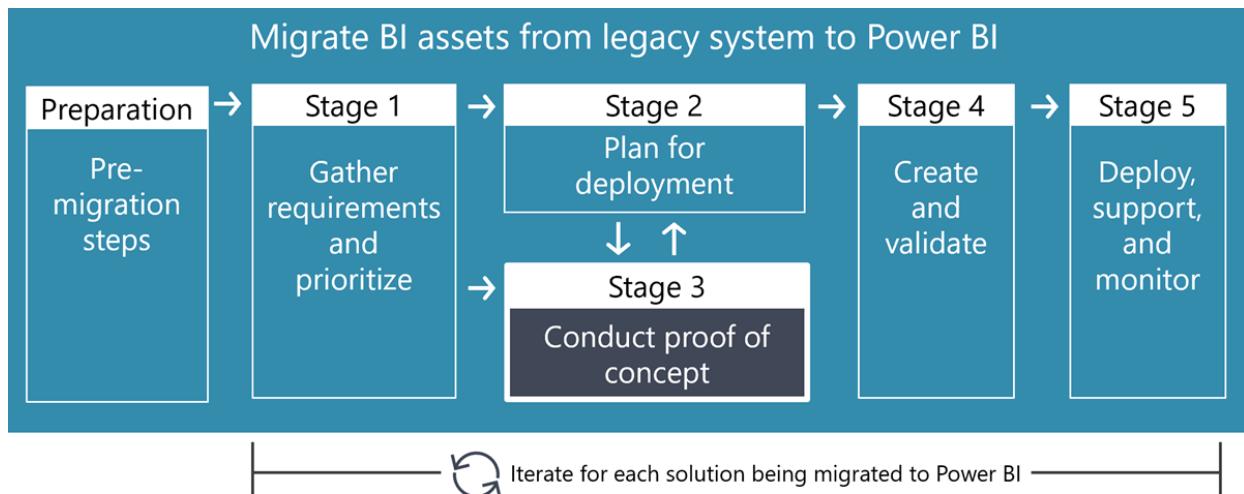
- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Conduct proof of concept to migrate to Power BI

3/24/2022 • 5 minutes to read • [Edit Online](#)

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 may 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 dataset 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 may 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 dataset using 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 may 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 dataset 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 may 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 may 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 datasets](#) 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 may 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.

Next steps

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:

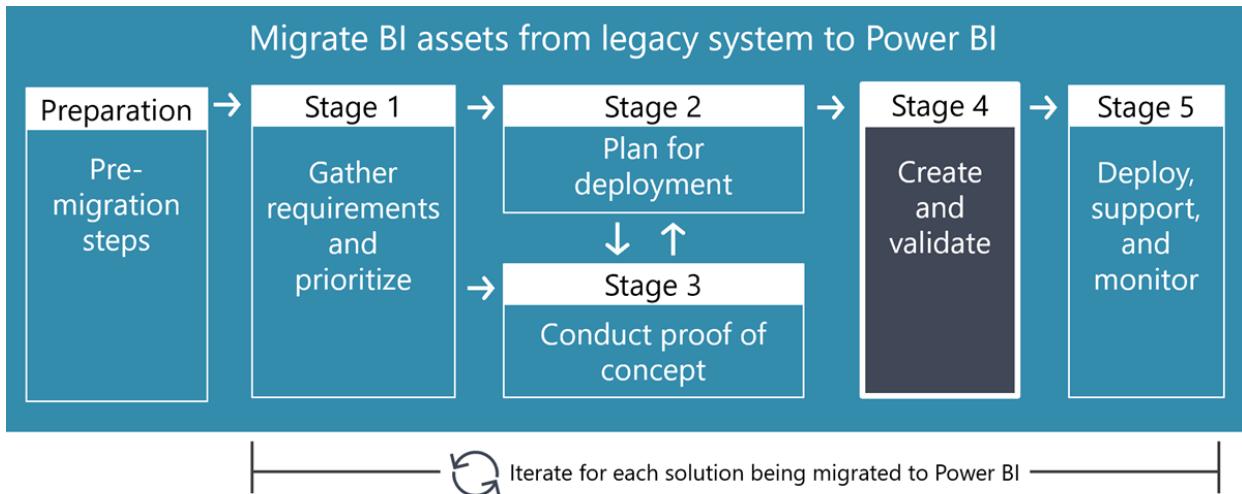
- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Create content to migrate to Power BI

3/24/2022 • 6 minutes to read • [Edit Online](#)

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 may carry on with producing the production-ready Power BI solution. Or, someone different may 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 dataset

You may choose to create a new Import dataset when an existing Power BI dataset 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 dataset include:

- [Acquire data](#) from one or more data sources (which may be a Power BI dataflow).
- [Shape, combine, and prepare](#) data.
- Create the [dataset 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 may 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 datasets, reports, and dashboards

In an Import dataset, 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 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 dataset 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 may 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 dataset 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](#).

Document the solution

There are two main types of documentation that are useful for a Power BI solution:

- Dataset 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 may exist for your Center of Excellence or an internal Power BI community site.
- **Within an app:** URLs may be configured 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 **Fields** 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 dataset documentation

Dataset documentation is targeted at users who will be managing the dataset in the future. It's useful to include:

- Design decisions made and reasons why.
- Who owns, maintains, and certifies datasets.
- Data refresh requirements.
- Custom business rules defined in datasets.
- Specific dataset security or data privacy requirements.
- Future maintenance needs.
- Known open issues or deferred backlog items.

You may also elect to create a change log that summarizes the most important changes that have happened to the dataset 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.

Next steps

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:

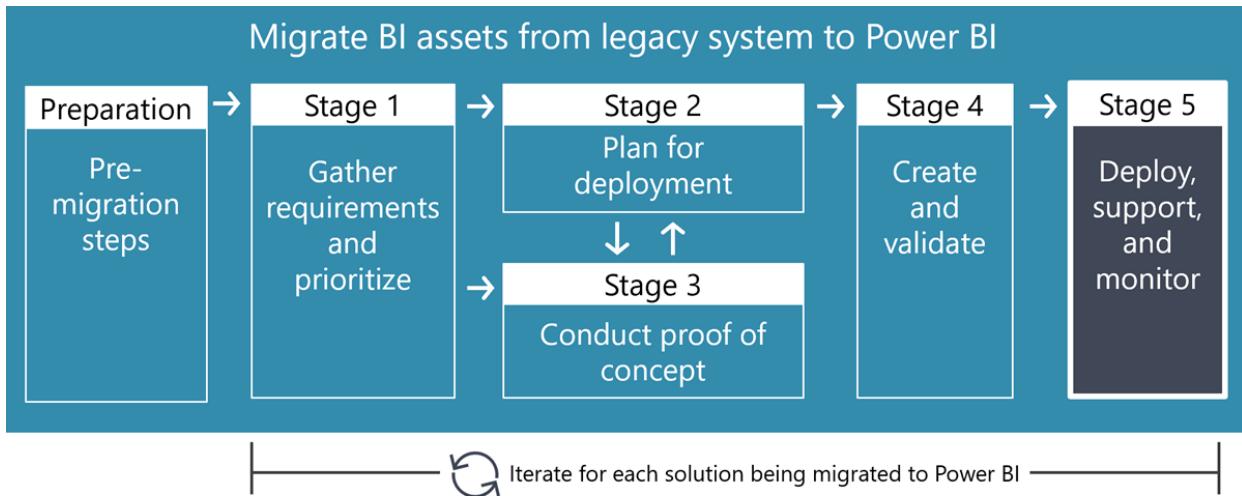
- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Deploy to Power BI

3/24/2022 • 8 minutes to read • [Edit Online](#)

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 graphic, 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 may be done manually or with [PowerShell scripts](#).

Deploy to test workspace

Key activities during a deployment to the test workspace typically include:

- **Connection strings and parameters:** Adjust dataset connection strings if the data source differs between

development and test. [Parameterization](#) can be used to effectively manage connection strings.

- **Workspace content:** Publish datasets 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 dataset refresh](#) for any Import datasets 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.

NOTE

For more information about options for deployment to development, test, and production, see Section 9 of the [Planning a Power BI enterprise deployment whitepaper](#).

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 may 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 may opt to perform a staged deployment. The first deployment to production may 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 may 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 may be required.
- **Gateway drivers and connectors:** A new proprietary data source may require installation of a new driver or custom connector on each server in the gateway cluster.
- **Create a new Premium capacity:** You may be able to use an existing [Premium capacity](#). Or, there may 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 may [feature content](#) in the Power BI service home page.
- **Set up sensitivity labels:** All [sensitivity labels](#) are integrated with Microsoft 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 may 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 dataset connection strings when the data source differs between test and production. [Parameterization](#) can be used effectively for this purpose.
- **Data refresh:** [Schedule the dataset refresh](#) for any imported datasets.
- **Workspace content:** Publish datasets 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.
- **Dataset settings:** Update and verify settings for each dataset, 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 Power BI adoption roadmap.

You may 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 [Power BI adoption roadmap: User support](#).

Next steps

In the [final article in this series](#), learn from customers when migrating to Power BI.

Other helpful resources include:

- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Learn from customer Power BI migrations

3/24/2022 • 11 minutes to read • [Edit Online](#)

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 may 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 may 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

may contain a single analyst, while a larger group can 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 dataset 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 dataset. It resulted in many datasets 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

Since 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 Power BI 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.

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.

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.

Next steps

Other helpful resources include:

- [Microsoft's BI transformation](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- [Dashboard in a Day](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Power BI adoption roadmap

3/24/2022 • 7 minutes to read • [Edit Online](#)

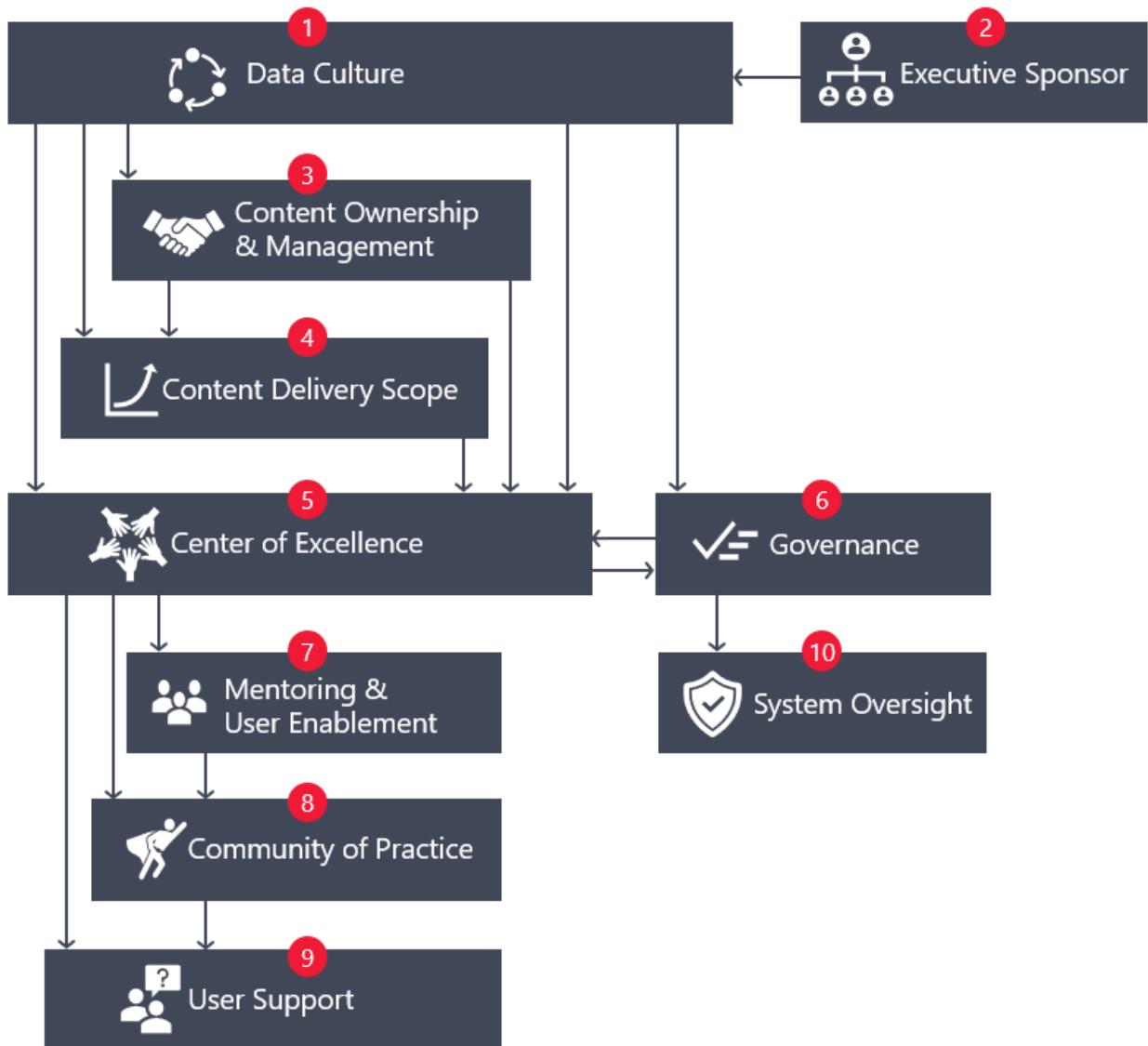
The goal of this series of articles is to provide a roadmap. The roadmap presents a series of strategic and tactical considerations and action items that directly lead to successful Power BI adoption, and help build a data culture in your organization.

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 a lot of considerations across the spectrum of people, processes, and technology.

NOTE

While reading this series of articles, it's recommended you also take into consideration [Power BI implementation planning](#) guidance. After you're familiar with the concepts in the Power BI adoption roadmap, consider reviewing the [usage scenarios](#). Understanding the diverse ways how Power BI is used can influence your implementation strategies and decisions.

This series of articles correlates with the following Power BI adoption roadmap diagram:



The areas in the above diagram include:

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 Power BI, and it is 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 Power BI.
3	Content ownership and management: There are three primary strategies for how business intelligence (BI) 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.

Area	Description
4	Content delivery scope: There are four primary strategies for content delivery including personal BI, team BI, departmental BI, and enterprise BI. These strategies have a significant influence on adoption, governance, and the COE operating model.
5	Center of Excellence: A Power BI 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.
6	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 Power BI, 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.
7	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.
8	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.
9	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.
10	System oversight: System oversight includes the day-to-day administration responsibilities to support the internal processes, tools, and people.

The relationships in the diagram shown above can be summarized in the following bullet list:

- 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.
- All data culture and adoption-related decisions and actions are accomplished more easily with guidance and leadership from an **executive sponsor**.

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.

Power BI adoption

Successful Power BI adoption 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 the [Power BI adoption maturity levels](#) article 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 is not the only factor. Adoption is not just about *using* the technology on a regular basis; it's about using it *effectively*. Effectiveness is much harder to define and measure.

Whenever possible, adoption efforts should be aligned across analytics platforms, BI services, and other Power Platform products. These products include Power Apps and Power Automate.

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](#)
- [Content ownership and management](#)
- [Content delivery scope](#)
- [Center of Excellence](#)
- [Governance](#)
- [Mentoring and enablement](#)
- [Community of practice](#)
- [User support](#)
- [System oversight](#)
- [Conclusion and additional resources](#)

IMPORTANT

You may be wondering how this Power BI adoption roadmap is different from the [Power BI adoption framework](#). The adoption framework was created primarily to support Microsoft partners. It is a lightweight set of resources to help partners deploy Power BI solutions for their customers.

This Power BI adoption series is more current. It is intended to guide any person or organization that is using—or considering using—Power BI. If you're seeking to improve your existing Power BI implementation, or planning a new Power BI implementation, this adoption roadmap is a great place to start. You will find a lot of valuable information in the [Power BI adoption framework](#), so we encourage you to review it.

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 Power BI.
- Increasing their organization's maturity level related to Power BI delivery.
- Understanding and overcoming adoption-related challenges faced when scaling Power BI.
- Increasing their organization's return on investment (ROI) in data and analytics.

Primarily, this series of articles will be helpful to those who work in an organization with one or more of the following characteristics:

- Power BI is deployed with some successes.
- Power BI has pockets of viral adoption, but is not purposefully governed across the entire organization.
- Power BI is 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 Power BI is under consideration or is planned.

Secondarily, this series of articles will be helpful for:

- Organizations that are in the early stages of a Power BI 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 Power BI technology platform, with an emphasis on the Power BI service.

To fully benefit from the information provided in these articles, it is an advantage to have at least an understanding of [Power BI fundamental concepts](#).

Next steps

In the [next article in this series](#), learn about the Power BI 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](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

Experienced Power BI partners are available to help your organization succeed with adoption of Power BI. To engage a Power BI partner, visit the [Power BI partner portal](#).

Acknowledgments

This series of articles was written by Melissa Coates, Data Platform MVP, and owner of [Coates Data Strategies](#), with significant contributions from Matthew Roche. Reviewers include Cory Moore, James Ward, Timothy Bindas, Greg Moir, Chuy Varela, Daniel Rubiolo, Sanjay Raut, and Peter Myers.

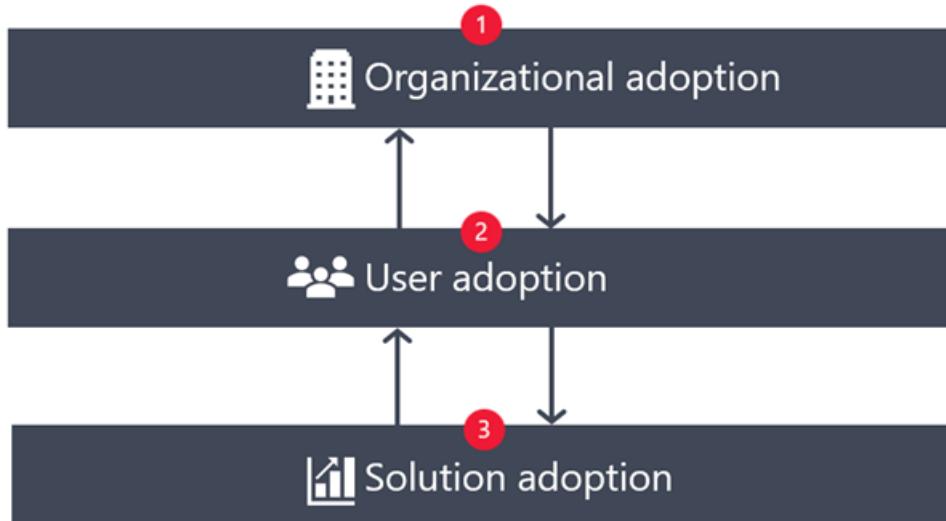
Power BI adoption roadmap maturity levels

3/24/2022 • 12 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

There are three inter-related perspectives to consider when adopting a technology like Power BI.



The three types of adoption shown in the above diagram include:

TYPE	DESCRIPTION
1	Organizational adoption: Organizational adoption refers to the effectiveness of Power BI governance. It also refers to data management practices that support and enable business intelligence efforts.
2	User adoption: User adoption is the extent to which consumers and creators continually increase their knowledge. It is concerned with whether they are actively using Power BI, and whether they are using it in the most effective way.
3	Solution adoption: Solution adoption refers to the impact and business value achieved for individual requirements and artifacts.

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, an app, or a dataset—impacts and guides users on how to use Power BI 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 Power BI by educated and informed users, contributes to stronger and more successful individual solutions.

The remainder of this article introduces the three types of Power BI adoption in more detail.

Organizational adoption maturity levels

Organizational adoption measures the state of Power BI governance and data management practices. There are several organizational adoption goals:

- Effectively support the community.
- Enable and empower users.
- Oversee information delivery via enterprise BI and self-service BI with continuous improvement cycles.

It is 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 Power BI 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 the DMM is a paid resource).

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 Power BI, 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 cognizant 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 is 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 Power BI exist in one or more areas of the organization.
- Achieving quick wins has been a priority, and it has delivered some successes.
- 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 Power BI is acknowledged, but there is 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. Data management is based on defined processes, though these processes may not apply uniformly throughout the organization.

Common characteristics of maturity level 200 include:

- Certain Power BI content is now critical in importance and/or it is broadly used by the organization.

- There are attempts to document and define repeatable practices, however efforts are siloed, reactive, and deliver varying levels of success.
- There is an over-reliance on individuals having good judgment and adopting healthy habits that they learned on their own.
- Power BI adoptions 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 Power BI governance strategy is underway.
- There is recognition that a Power BI 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 Power BI.
- Progress is made on the standardization of repeatable practices, though less-than-optimal aspects may still exist due to early uncontrolled growth.
- The Power BI COE is established, and it has clear goals and scopes of responsibilities.
- The internal community gains traction with the participation of a growing number of users.
- Power BI champions emerge in the community.
- Initial investments in training, documentation, and resources are made.
- An initial governance model is in place.
- Power BI has an active and engaged executive sponsor.
- Roles and responsibilities for all Power BI 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:

- Business intelligence efforts deliver significant value.
- Power BI is commonly used for delivering critical content throughout the organization.
- There is 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 Power BI community of users.
- Standardized processes are in place for the oversight and monitoring of Power BI usage and practices.
- The Power BI COE includes representation from all key business units.
- A Power BI champions network supports the internal community: champions actively work with their colleagues and 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 Power BI solutions is prevalent in the organization, and Power BI is widely accepted throughout the organization.
- Power BI skillsets are highly valued in the organization, and they are recognized by leadership.
- The internal Power BI community is self-sustaining, with support from the COE. The community is not 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 above characteristics 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 Power BI 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 Power BI adoption series present maturity levels on a per-topic basis.

Individuals—and the organization itself—continually learn, change, and improve. So, that means there's no formal end to adoption-related efforts. However, it is 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 using Power BI effectively. Usage statistics alone do not indicate user adoption. User adoption is also concerned with individual user behaviors and practices. The aim is to ensure users engage with Power BI in the correct way and to its 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, Power BI in some way.
- An individual may have access to Power BI but is not 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 Power BI to deliver analytical value and support decision-making.

- An individual shows interest and starts to use Power BI.

User adoption stage 3 – Momentum

Common characteristics of stage 3 user adoption include:

- An individual actively gains Power BI skills by attending formal training, self-directed learning, or experimentation.
- An individual gains basic competency with the aspects of Power BI relevant to their role.

User adoption stage 4 – Proficiency

Common characteristics of stage 4 user adoption include:

- An individual actively uses Power BI on a regular basis.
- An individual understands how to use Power BI in the way in which it was 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 Power BI 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 individual Power BI solutions. It is 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 or a single Power BI app.

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 BI, 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.
- All exploration—and initial feedback—could occur within Power BI Desktop or Excel. Use of the Power BI service is limited.

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 Power BI service.
- All necessary supporting components are in place, such as gateways to support scheduled refresh.
- Users are aware of the solution and show interest in using it. Potentially, it may be a limited preview release, and may 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 is valuable and experience tangible benefits.
- The solution is promoted to a production [workspace](#).
- 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), and help future creators (such as documenting any future maintenance or planned enhancements).
- Ownership and subject matter experts for the content is clear.
- Report branding and theming are in place, and they are inline 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 is carefully controlled due to the impact of changes.
- A subset of users regularly provide feedback to ensure the solution continues to meet 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](#) since it's critical in nature.
- Formal user acceptance testing for new changes may occur, particularly for IT-managed content.

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn more about the organizational data culture and its impact on adoption efforts.

Power BI adoption roadmap: Data culture

3/24/2022 • 10 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

Building a data culture is closely related to adopting Power BI, 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 the organization. It encourages a culture that regularly employs informed data decision-making:

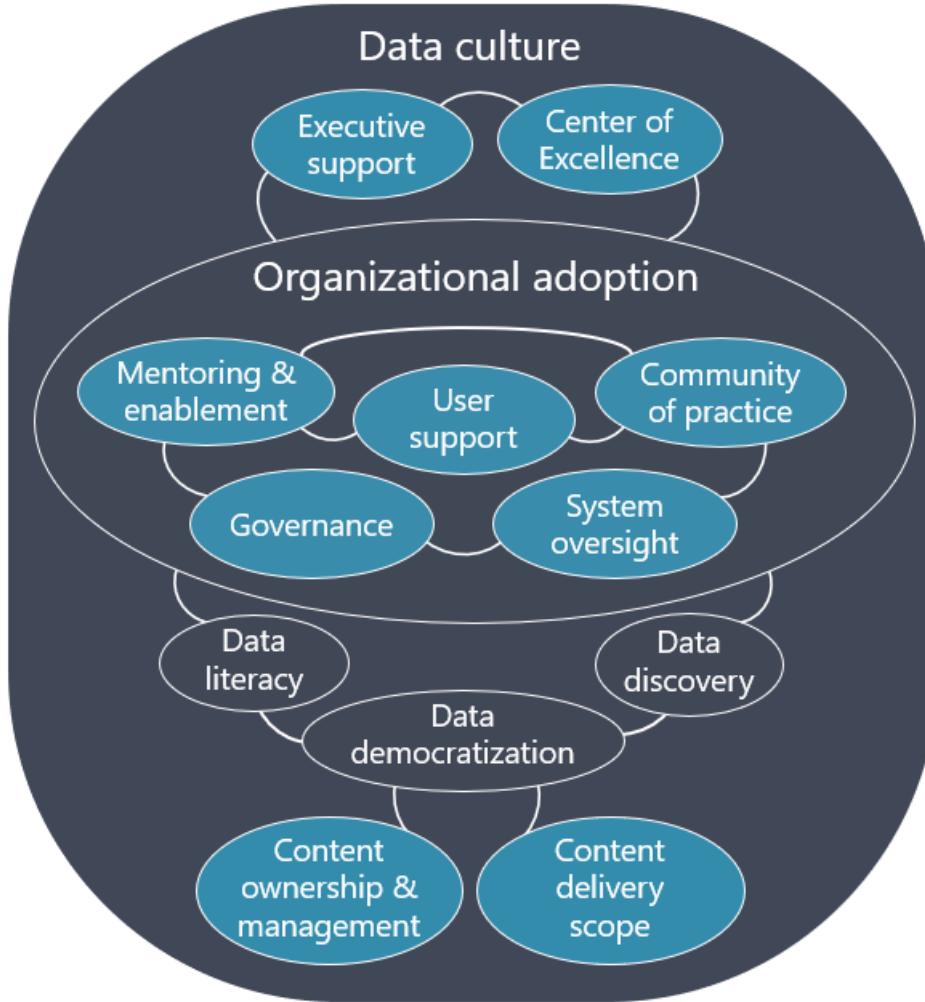
- By more stakeholders throughout more areas of the organization.
- Based on analytics, not opinion.
- In an effective, efficient way that is based on best practices endorsed by the [Center of Excellence](#).
- Based on trusted data.
- That reduces reliance on undocumented tribal knowledge.
- That reduces 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. It means that the specific ways that data culture objectives are achieved can vary across organizational boundaries. What's important is that they should all align with the organizational data culture objectives. This can be thought of as *aligned autonomy*.

The following diagram conveys the inter-related aspects that influence your data culture:



The above circular diagram represents the somewhat ambiguous relationships among the following items:

- **Data culture** as the outer circle: all topics contained 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**) as 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.

All items in the above diagram are discussed throughout the remainder of this series of articles.

Data culture vision

The concept of data culture is imprecise and can be difficult to define and measure. Even though it's challenging to articulate in a way that is meaningful, actionable, and measurable, it's important that you 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 are not specifically mandated. Rather, the state of the data culture is the result of

following the governance rules as they are enforced (or the lack thereof). Leaders at all levels need to actively demonstrate what's important through their actions, including how they praise, recognize, and reward staff members who take initiative. It's what makes the concept of data culture imprecise and somewhat abstract.

TIP

If you can take for granted that your efforts to develop a data solution (such as a dataset 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 may be:

- A reactive change, such as responding to new agile competition.
- A proactive change, such as initiating a new line of business, or expanding into new markets to seize a "green field" opportunity. It can be relatively easy to be data-driven from the beginning 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 any of these situations, there is often a specific area where the data culture takes root. The specific area could be a scope of effort that is smaller than the entire organization, even if it is still significant. Once 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 adoption series of articles covers a lot of topics that contribute to 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 artifacts.

Data discovery is the ability to effectively search for, and locate, relevant data sources and artifacts across the organization. Primarily, data discovery is concerned with improving awareness that data exists, particularly when data is siloed in departmental systems. Once aware of its existence, a user can go through the standard process to request access to the information. Today, technology helps a lot with data discovery, advancing well past asking colleagues where to find data artifacts.

TIP

It's important to have a clear and simple process so users can request access to data. Knowing that a data artifact exists—but unable to access it within the guidelines and processes established by the domain owner—can be a source of frustration for users. It can force them to use inefficient workarounds rather than pursue 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 investments in existing data resources.
- Promoting the use and enrichment of existing Power BI artifacts.
- Helping people understand who owns and manages data artifacts.

- Establishing connections between consumers, creators, and owners.

In Power BI, the [datasets hub](#) and [use of endorsements](#) help promote data discovery of [shared datasets](#). It also encourages the [reuse and augmentation](#) of datasets by self-service creators. Further, data catalog solutions such as [Azure Purview](#), which can scan and catalog an entire Power BI tenant, are extremely valuable for data discovery purposes. They can record metadata tags and descriptions to provide deeper context and meaning.

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 them to make decisions with the data.

NOTE

The concept of data democratization does not imply a lack of security, nor does it imply a lack of justification based on job role. As part of a healthy data culture, data democratization helps reduce *shadow IT* by providing data artifacts that are secured, governed, well-managed, and that 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. Here are some examples of Power BI governance decisions that can impact on data democratization:

- 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?
- What is the desired level of self-service BI user enablement? How does this vary based on business unit or job role?
- What is the desired balance between enterprise BI and self-service BI?
- Are there data sources that are strongly preferred? What is 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 own and manage self-service BI content?
- Who can consume content? Is this decision different for external partners, customers, and suppliers?

WARNING

If access to data and/or the ability to perform analytics is limited to a select number of individuals in the organization, that's typically a warning sign since the ability to work with data is a key characteristic of a data culture.

Data literacy

Data literacy refers to the ability to interpret, create, and communicate data 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 produce high-quality solutions, however 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 Power BI 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. Several things you can focus on that are related to data literacy include:

- Interpreting charts and graphs.
- Assessing the validity of data.
- Performing root cause analysis.

- Discerning correlation and causation.
- Understanding how context and outliers affect results that 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 can be solved or mitigated 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, and from there on to the solution details.

Considerations and key actions

Here are some considerations and key actions you can take to strengthen your data culture:

- Give serious consideration to the type of data culture you want to cultivate. Ideally, it's from a position of user empowerment than a position of command and control.
- Talk to stakeholders in different business units to understand what analytics practices are currently working well and what isn't working well for data-driven decision-making. Conduct a series of workshops to understand current state and formulate the desired future state.
- Talk to stakeholders in IT, BI, and/or the [COE](#) to understand what [governance](#) constraints need consideration. These talks can present an opportunity to educate teams on topics like security and infrastructure, and what Power BI actually is (and that it's significantly more powerful than a query tool).
- Verify the level of [executive sponsorship](#) and support you have in place to advance data culture goals.
- Make purposeful decisions about your BI strategy. It includes deciding what the ideal balance of business-led self-service BI, managed self-service BI, and enterprise BI should be for the key business units in the organization (discussed in the [content ownership and management](#) article). Also consider how it relates to the extent of published content for personal BI, team BI, departmental BI, and enterprise BI (discussed in the [content delivery scope](#) article). Determine how these decisions affect your action plan.
- Begin creating an action plan for immediate, short-term, and long-term action items. Identify business groups and problems that represent a "quick win" and can make a visible difference.

Maturity levels

The following maturity levels will help you assess the current state of your data culture:

LEVEL	STATE OF DATA CULTURE
100: Initial	<p>There are business intelligence (BI) initiatives taking place—with some successes—in various areas of the organization. These activities are occurring in a somewhat chaotic manner, with few formal processes, and without a strategic plan.</p> <p>A significant backlog of requests may exist for the enterprise BI team.</p> <p>There is a lack of oversight and visibility into self-service BI operations. The successes or failures of BI solutions are not well-understood.</p>

LEVEL	STATE OF DATA CULTURE
200: Repeatable	<p>Multiple teams have had measurable successes with self-service BI solutions.</p> <p>Investments are made to identify the ideal balance of enterprise BI and self-service BI.</p>
300: Defined	<p>Specific goals are established for advancing the data culture, and these goals are implemented incrementally.</p> <p>Effective self-service BI practices are incrementally—and purposely—replicated throughout more areas of the organization.</p>
400: Capable	<p>The data culture goals to employ informed decision-making are aligned with organizational objectives, actively supported by the executive sponsor, and directly impact on the adoption strategies put into place.</p> <p>A healthy and productive partnership exists between the COE, business units, and IT.</p> <p>It's common that individuals who take initiative in building valuable BI solutions are recognized and rewarded.</p>
500: Efficient	<p>Regular review of key performance indicators validate that data culture results are aligned with intentions.</p> <p>The business value of BI solutions is regularly evaluated and measured; feedback loops encourage ongoing improvements.</p> <p>Continual improvement of organizational adoption, user adoption, and solution adoption are top priorities. Learnings are incrementally adapted throughout the organization on an ongoing basis.</p>

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn more about the importance of an executive sponsor.

Power BI adoption roadmap: Executive sponsorship

3/24/2022 • 4 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

When planning to advance the data culture and the state of organizational adoption for Power BI, it's crucial to have executive support. An executive sponsor is imperative because adopting Power BI 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 activities such as:

- Formulating a strategic vision and priorities for BI and analytics.
- Leading by example by actively using Power BI in a way that's consistent with data culture and adoption goals.
- Allocating staffing and prioritizing resources.
- Approving funding (for example, Power BI licenses).
- Communicating announcements that are of critical importance.
- Decision-making, particularly for strategic-level governance decisions.
- Dispute resolution (for escalated issues that cannot be resolved by operational or tactical personnel).
- Supporting organizational changes (for example, creating or expanding the [Center of Excellence](#)).

IMPORTANT

The ideal executive sponsor has sufficient credibility, influence, and authority throughout the organization.

Identifying an executive sponsor

There are multiple ways to identify an executive sponsor.

Top-down pattern

An executive sponsor may be selected by a more senior executive. For example, the Chief Executive Officer (CEO) may hire a Chief Data Officer (CDO) or Chief Analytics Officer (CAO) to advance the organization's data culture objectives. The CDO or CAO then becomes the ideal candidate to serve as the executive sponsor for Power BI (or analytics in general).

Here's another example: The CEO may 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 a Power BI executive sponsor at the C-level is an excellent 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 BI. For example, a business unit within the organization, such as Finance, has organically achieved great success with respect to their use of data and analytics. Essentially, they successfully formed their own data culture on a small scale. A leader who hasn't reached the executive level (such as the Finance Director) may 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's because the return on investment and strategic imperative of a data culture or digital transformation is not an urgent priority for C-level executives.

With a bottom-up approach, the sponsor may 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 may start a healthy competition across other business units in the adoption of BI.

Considerations and key actions

Here is a list of considerations and key actions you can take to establish or strengthen executive support for Power BI:

- Ensure you have clarity on who your executive sponsor is.
- Identify someone in a sufficient position of influence and authority (across organizational boundaries) who understands the value and impact of business intelligence. It is important that the individual has a vested interest in the success of analytics in the organization.
- Involve your executive sponsor in all strategic-level governance decisions involving business intelligence and data management.
- Involve your executive sponsor in all data culture initiatives to ensure alignment and consensus on goals and priorities.
- Formalize the arrangement with documented responsibilities for the executive sponsor role, so there is no uncertainty about expectations.
- Consider naming a backup executive sponsor. The backup can attend meetings in the sponsor's absence and make time-sensitive decisions when necessary.
- Identify influential advocates within the leadership team for each business unit. In addition to your executive sponsor, it is easier to accomplish your objectives if you have the cooperation and involvement of business unit leaders.

Maturity levels

The following maturity levels will help you assess your current state of executive support:

LEVEL	STATE OF POWER BI EXECUTIVE SUPPORT

LEVEL	STATE OF POWER BI EXECUTIVE SUPPORT
100: Initial	There is awareness from at least one executive about the strategic importance of how Power BI can play a part in advancing the organization's data culture goals. However, neither a Power BI sponsor nor an executive-level decision-maker is identified.
200: Repeatable	Executive support exists for Power BI through informal channels and relationships.
300: Defined	A formal arrangement exists for C-level Power BI sponsorship and support, with well-understood expectations for the role.
400: Capable	Senior-level support is in place to grow and sustain the investment in Power BI by someone with sufficient authority across organizational boundaries.
500: Efficient	Executive support is a driver for advancing the data culture vision. The executive sponsor is involved with ongoing organizational adoption improvements, and measurable goals are in place.

Next steps

In the [next article](#) in the Power BI 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.

Power BI adoption roadmap: Content ownership and management

3/24/2022 • 17 minutes to read • [Edit Online](#)

NOTE

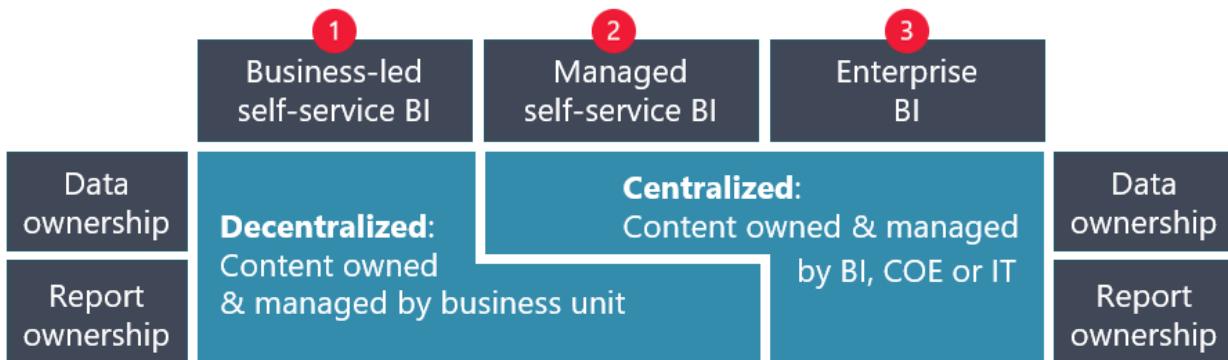
This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

NOTE

The Power BI implementation planning [usage scenarios](#) explore many concepts discussed in this article. The usage scenario articles include detailed diagrams that you may find helpful to support your planning and decision making.

There are three primary strategies for how business intelligence (BI) content is owned and managed: business-led self-service BI, managed self-service BI, and enterprise BI. For the purposes of this series of articles, the term *content* refers to any type of data artifact. It's synonymous with *solution*.

The organization's data culture is the driver for why, how, and by whom each of these three content ownership strategies are implemented.



The areas in the above diagram include:

AREA	DESCRIPTION
1	Business-led self-service BI: 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 <i>decentralized</i> or <i>bottom-up BI</i> strategy.
2	Managed self-service BI: 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 <i>discipline at the core and flexibility at the edge</i> .
3	Enterprise BI: 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 BI 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).
- 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 is 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.

In general:

- Business-led self-service BI content is subject to the least stringent governance and oversight controls. It often includes [personal BI](#) and [team BI](#) solutions.
- Managed self-service BI content is subject to moderately stringent governance and oversight controls. It frequently includes [team BI](#) and [departmental BI](#) solutions.
- [Enterprise BI](#) 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 many ways and can be easily misunderstood. The following table presents possible ways you may conceptually define these roles:

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 is used for, who may access it, and how the data is presented to others. Collaborates with domain owner as needed and supports colleagues in their use of data.

ROLE	DESCRIPTION
Technical owner	Responsible for creating, maintaining, publishing, and securing access to data and reporting artifacts.
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 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 artifacts. 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 Power BI service, content owners can set the [contact list property](#) for many types of artifacts. 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 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 artifacts to communicate ownership and/or support responsibilities.
- Specify 2-4 [workspace administrators](#) and conduct an [audit](#) of workspace admins on a regular basis (perhaps twice a year). Workspace admins might be directly responsible for managing workspace content, or it may 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 Power BI service. A standard template, as described in the [mentoring and user enablement](#) article, can encourage and simplify the consistent use of branding.
- Make use of best practices reviews with the COE, which are discussed in the [COE](#) article.

The remainder of this article covers considerations related to the three content ownership and management strategies.

Business-led self-service BI

With business-led self-service 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 BI is often a good strategy for [personal BI](#) and [team BI](#) solutions.

IMPORTANT

The concept of business-led self-service BI is not the same as shadow IT. In both scenarios, 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 complies with all established data governance guidelines and policies.

Business-led self-service BI 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 people capable of—and fully committed to—supporting solutions through the entire lifecycle. It covers all types of Power BI artifacts, including the data (dataflows and datasets), the visuals (reports and dashboards), and apps.
- The flexibility to respond to changing business conditions and react quickly outweighs the need for stricter governance and oversight.

Guidelines for being successful with business-led self-service BI:

- Teach your creators to use the same techniques that IT would use, like [shared datasets](#) and [dataflows](#). Having fewer duplicated datasets reduces maintenance, improves consistency, and reduces 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 (discussed 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 an app or workspace roles may 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.

Managed self-service BI

Managed self-service BI is a blended approach. 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 content at a pace faster than the centralized team can accommodate.
- Different people are responsible for handling data preparation, data modeling, and report creation.

Guidelines for being successful with self-service BI:

- Teach users to [separate model and report development](#). They can use [live connections](#) to create reports based on existing datasets. When the dataset 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 dataset creators. Refine the dataflow as much as possible, using friendly column names and correct data types to reduce the downstream effort required by dataset 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 datasets. The use of dataflows also reduces the number of data refreshes on source systems and allows fewer users requiring direct access to source systems.
- When self-service creators need to augment an existing dataset with departmental data, educate them to use [DirectQuery connections to Power BI datasets and Azure Analysis Services](#). 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 datasets 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 Power BI service.
- 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 dataset permissions allow creators to create new reports with row-level security (RLS) in effect, when applicable.
- Use the [Power BI REST APIs](#) to compile an inventory of Power BI artifacts. Analyze the ratio of datasets to reports to evaluate the extent of dataset reuse.

Enterprise BI

[Enterprise BI](#) is a centralized approach in which all content is owned and managed by a centralized team. This team is usually IT, enterprise BI, or the COE.

Enterprise BI is most suitable when:

- Centralizing content management with a single team aligns with the organization's data culture.
- The organization has BI expertise to manage all the BI artifacts 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 people.
- The data is highly sensitive or subject to regulatory requirements.

Guidelines for being successful with enterprise BI:

- Implement a rigorous process for use of the [certified endorsement](#) for datasets, reports, and apps. Not all enterprise BI 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 from the Power BI service.
- 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 dataset, and it doesn't include branding. The template file can be shared as a link within a Power BI app, or from the community site.

Ownership transfers

Occasionally, the ownership of a particular solution may 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 people, 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.
- 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 may be addressed during COE office hours; a more complex transfer may warrant a small project managed by the COE.

NOTE

There's potential that the new owner will need to do some refactoring 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, it's an ideal time to apply those enhancements. The branding of reports and dashboards may 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

Here is a list of considerations and key actions you can take to strengthen your approach to content ownership and management:

- 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 Power BI implementation planning [usage scenarios](#) to understand how Power BI can be used in diverse ways.
- 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.
- If your enterprise BI team currently has challenges related to scheduling and priorities, do an assessment to determine if a managed self-service BI strategy can be put in place to empower more content creators throughout the organization. Managed self-service BI can be extremely effective on a global scale.
- Clarify terms used in your organization for owner, data steward, and subject matter expert. Make sure roles and responsibilities are documented, including backup personnel.
- Ensure that all your content owners—from both the business and IT—are part of your [community of practice](#).
- Determine how you will use the contacts feature in Power BI. Communicate with content creators about how it should be used, and why it's important.
- If ownership transfers occur regularly, create a process for how it will work.
- Determine your strategy for using [external tools](#) for advanced authoring capabilities and increased productivity.

Maturity levels

The following maturity levels will help you assess the current state of your content ownership and management:

LEVEL	STATE OF POWER BI CONTENT OWNERSHIP AND MANAGEMENT
100: Initial	<p>Content is owned and managed by self-service creators throughout the organization in an uncontrolled way, without a specific strategy.</p> <p>A high ratio of datasets to reports exists, indicating an opportunity to improve data reusability and reduce the number of duplicate datasets.</p> <p>Discrepancies between different reports is common, causing distrust of content produced by others.</p>
200: Repeatable	<p>A plan is in place for which content ownership and management strategy to use and in which circumstances.</p> <p>Initial steps are taken to improve the consistency and trustworthiness levels for self-service BI efforts.</p> <p>Guidance for the user community is available that includes expectations for self-service versus enterprise content.</p>

LEVEL	STATE OF POWER BI CONTENT OWNERSHIP AND MANAGEMENT
300: Defined	<p>Managed self-service BI 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.</p> <p>Report branding is consistently used to indicate who produced the content.</p> <p>A mentoring program exists to educate self-service content creators on how to apply best practices and make good decisions.</p>
400: Capable	<p>Criteria is defined to align governance requirements for self-service versus enterprise content.</p> <p>There is a plan in place for how to request and handle ownership transfers.</p> <p>Managed self-service BI—and techniques for the reuse of data—are commonly used and well-understood.</p>
500: Efficient	<p>Proactive measures are in place to communicate with a user when any concerning user activities are detected in the activity log.</p> <p>External tools are used by highly proficient content creators to improve productivity and efficiency.</p>

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn more about the scope of content delivery.

Power BI adoption roadmap: Content delivery scope

3/24/2022 • 14 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

The four delivery scopes described in this article include [personal BI](#), [team BI](#), [departmental BI](#), and [enterprise BI](#). To be clear, focusing on the scope of a delivered BI solution does refer to the number of people who may view the solution, though the impact is much more than that. The scope strongly influences best practices for content distribution, sharing, 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.

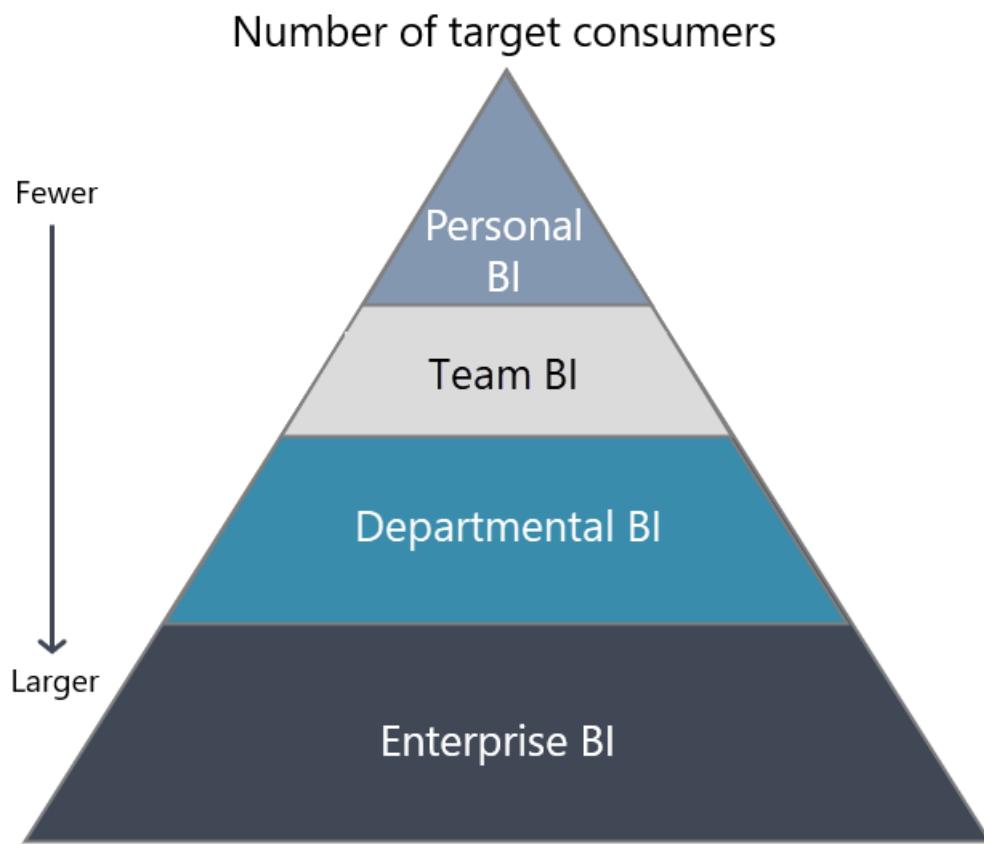
The related [content ownership and management](#) article makes similar points. Whereas the focus of that article was on the content creator, the focus of this article is on the target content usage. Both inter-related aspects need to be considered to arrive at governance decisions and the Center of 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

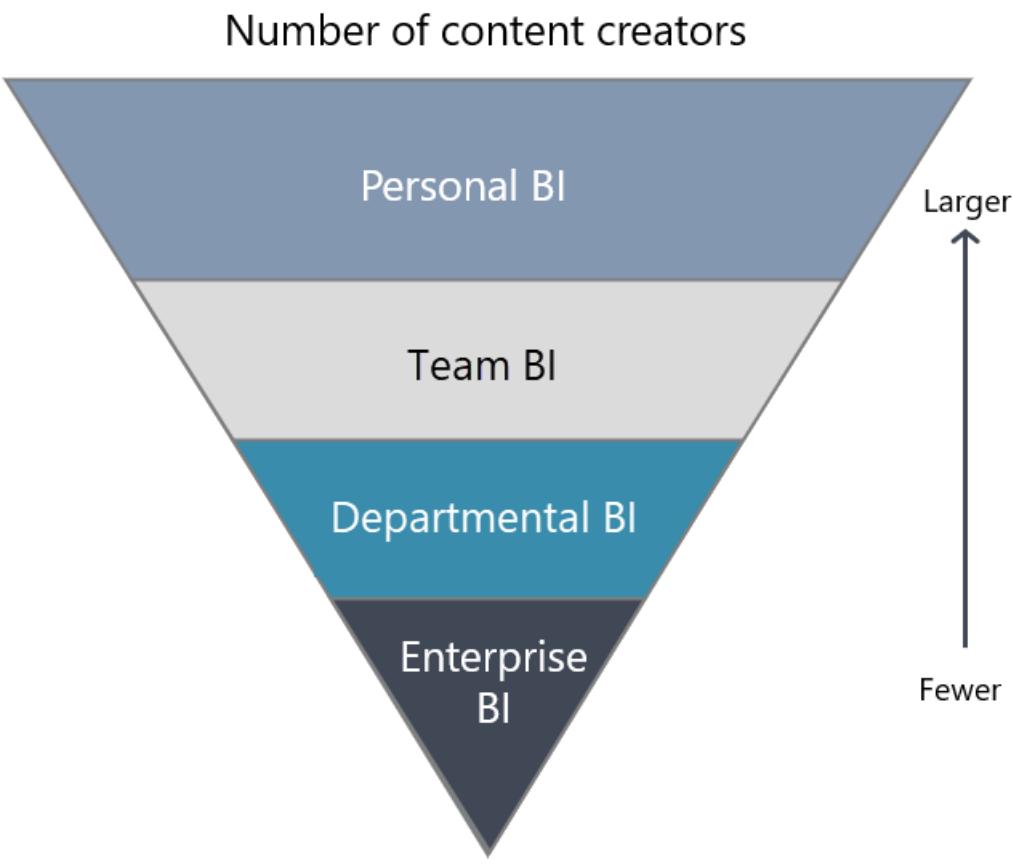
The following diagram focuses on the number of *target consumers* who will consume the content.



The four scopes of content delivery shown in the above diagram include:

- **Personal BI**: Personal BI solutions are, as the name implies, intended for use by the creator. So, sharing content with others isn't an objective. Therefore, personal BI has the fewest number of target consumers.
- **Team BI**: Collaborates and shares content with a relatively small number of colleagues who work closely together.
- **Departmental BI**: Delivers content to a large number of consumers, who can belong to a department or business unit.
- **Enterprise BI**: 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 BI:** Represents the largest number of creators because any user can work with data using business-led self-service BI methods. Although managed self-service BI methods can be used, it's less common with personal BI.
- **Team BI:** Colleagues within a team collaborate and share with each other using business-led self-service BI patterns. It has the next largest number of creators in the organization. Managed self-service BI patterns may also begin to emerge as skill levels advance.
- **Departmental BI:** Involves a smaller population of creators. They are likely to be considered power users who are using sophisticated tools to create sophisticated solutions. Managed self-service BI practices are very common and highly encouraged.
- **Enterprise BI:** Involves the smallest number of content creators because it typically includes only professional 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 BI, managed self-service BI, and enterprise BI. The most common alignment between ownership and delivery scope is:

- **Business-led self-service BI ownership:** Commonly deployed as personal and team BI solutions.
- **Managed self-service BI ownership:** Can be deployed as personal, team, or departmental BI solutions.
- **Enterprise BI ownership:** Deployed as enterprise BI-sscoped 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 Power BI](#), which is a specific implementation where a user or group is granted read-only access to a single artifact. In this article, the term *sharing* is meant in a general way to describe sharing content with colleagues. When the sharing feature is intended, this article will make a clear reference to that feature.

Personal BI

[Personal BI](#) 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 of content with others isn't the objective. Personal content can reside in Power BI Desktop or in a personal workspace in the Power BI service. Usage of the personal workspace is permitted with the free Power BI license.

Characteristics of personal BI:

- 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 may be an exploratory proof of concept that may, or may not, evolve into a project.

Guidelines for being successful with personal BI:

- Consider personal 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 may 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 BI, and managed self-service BI in the [content ownership and management](#) article. They are highly relevant techniques that help content creators create efficient and personal BI solutions.
- Analyze data from the [activity log](#) to discover situations where personal BI 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

See the [adoption maturity levels](#) article for information about how users progress through the stages of user adoption. See the [system oversight](#) article for information about usage tracking via the activity log.

Team BI

[Team BI](#) 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. Due to this work style, team members will typically each have a [Power BI Pro](#) or [Power BI Premium Per User \(PPU\)](#) license.

Content is often shared among the team more informally as compared to departmental or enterprise BI. For instance, the workspace is often sufficient for consuming content within a small team. It doesn't require for

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.

Characteristics of team BI:

- 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 reports may occur by 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.

Guidelines for being successful with team BI:

- 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 an app. It's tempting to start with one workspace per team, but that may not be flexible enough to satisfy all needs.
- See the techniques described for business-led self-service BI and managed self-service BI in the [content ownership and management](#) article. They are highly relevant techniques that help content creators create efficient and effective team BI solutions.

Departmental BI

Content is delivered to members of a department or business unit. Content distribution to a larger number of consumers is a priority for [departmental BI](#).

Usually there's a much larger number of consumers who are content viewers (versus a much smaller number of content creators). Therefore, a combination of [Power BI Pro](#) licenses, [Premium Per User](#) licenses, and/or [Premium capacity](#) licenses may be used.

Characteristics of departmental BI delivery:

- A few content creators typically publish content for colleagues to consume.
- Formal delivery of reports and 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 application lifecycle management (ALM) begin to emerge to ensure release stability and a consistent experience for consumers.

Guidelines for being successful with departmental BI delivery:

- Ensure the COE is prepared to support the efforts of self-service creators. Creators who publish content used throughout their department or business unit may emerge as candidates to become champions, or they may 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 [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.
- As the importance and criticality level grows, expectations for trustworthiness grows 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](#) may be permitted for departmental BI 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 Power BI service, content owners can set the [contact list property](#) for many types of artifacts. 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 will 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 to align with departmental colors and styling. It can also indicate who produced the content. For more information, see the [Content ownership and management](#) article. A small image or text label in the report footer is valuable when the report is exported from the Power BI service. A standard Power BI Desktop template file can encourage and simplify the consistent use of branding. For more information, see the [Mentoring and user enablement](#) article.
- See the techniques described for business-led self-service BI and managed self-service BI in the [content ownership and management](#) article. They are highly relevant techniques that help content creators create efficient and effective departmental BI solutions.

Enterprise BI

[Enterprise BI](#) content is typically managed by a centralized team and is subject to additional governance requirements. Content is delivered broadly across organizational boundaries.

Enterprise BI usually has a significantly larger number of consumers versus content creators. Therefore, a combination of [Power BI Pro](#) licenses, [Premium Per User](#) licenses, and/or [Premium capacity](#) licenses may be used.

Characteristics of enterprise BI delivery:

- A centralized team of BI experts manages the content end-to-end and publishes it for others to consume.
- Formal delivery of reports and 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 datasets and dataflows may 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 BI 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 may exist, possibly with other Azure services and Power Platform products.

Guidelines for being successful with enterprise BI delivery:

- Governance and oversight techniques described in the [governance](#) article are relevant for managing an enterprise BI solution. Techniques primarily include change management and application lifecycle management.
- Plan for how to effectively use [Premium Per User](#) or [Premium 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 BI content. 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 BI reports and apps. Datasets and dataflows can be certified, too, when there's the expectation that self-service creators will build solutions based on them. Not all enterprise BI 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 to align with departmental colors and styling. It can also indicate who produced the content. For more information, see the [Content ownership and management](#) article. A small image or text label in the report footer is valuable when the report is exported from the Power BI service. A standard Power BI Desktop template file can encourage and simplify the consistent use of branding. For more information, see the [Mentoring and user enablement](#) article.
- 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 BI in the [content ownership and management](#) article. They are highly relevant techniques that help content creators create efficient and effective enterprise BI solutions.
- See the techniques described in the [system oversight](#) article for auditing, governing, and the oversight of enterprise BI content.

Considerations and key actions

Considerations and key actions you can take to strengthen your approach to content delivery:

- Ensure that guidelines, documentation, and other resources align with the strategic goals defined for Power BI adoption.
- Clarify the scopes for content delivery in your organization, who they apply to, and how they align with governance decisions. Ensure it's consistent with how [content ownership and management](#) is handled.
- Consider 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, an app owned by a divisional sales team is distributed to the entire organization.
- Create helpful documentation for your community so content creators understand when it's appropriate to use [workspaces](#), [apps](#), or [sharing \(direct access or link sharing\)](#).
- Ensure you have a specific strategy in place to handle user licensing considerations for [Power BI Pro](#), [Premium Per User](#), and [Premium capacity](#). Create a process for how workspaces may be assigned each license type, and the prerequisites required for the type of content that may be assigned to Premium.

Maturity levels

The following maturity levels will help you assess the current state of your content delivery:

LEVEL	STATE OF POWER BI CONTENT DELIVERY
100: Initial	Content is published by self-service creators in an uncontrolled way, without a specific strategy.

LEVEL	STATE OF POWER BI CONTENT DELIVERY
200: Repeatable	Pockets of good practices exist which depend on the knowledge, skills, and habits of the content creator.
300: Defined	Clear guidelines are defined and communicated to describe what can and cannot occur within each delivery scope. These guidelines are followed by some—but not all—groups across the organization.
400: Capable	<p>Criteria are defined to align governance requirements for self-service versus enterprise content.</p> <p>Guidelines are followed by most, or all, groups across the organization.</p> <p>Change management requirements are in place to approve critical changes to content that is distributed to a larger audience.</p> <p>Changes are announced so creators are aware of the downstream effects on their content.</p>
500: Efficient	<p>Proactively communicate with a user when any concerning activities are detected in the activity log.</p> <p>Analysis is conducted regularly to evaluate the business value that's achieved for deployed solutions.</p>

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn more about the Center of Excellence.

Power BI adoption roadmap: Center of Excellence

3/24/2022 • 12 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

A Power BI 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 *business intelligence (BI) competency center*, *capability center*, or a *center of expertise*. Some organizations use the term *squad*. Many organizations perform the COE responsibilities within their BI team or analytics 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 BI or analytics team to take on many of the COE responsibilities; some responsibilities may also reside within IT. For simplicity, in this series of articles, COE means a *specific group of people*, although you may implement it differently. It's also very common to implement the COE with a scope broader than Power BI alone: for instance, a Power Platform COE or an analytics COE.

Goals for a COE

Goals for a COE include:

- Evangelizing a data-driven culture.
- Promoting the adoption of Power BI.
- 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 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 Power BI is 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 others. To varying degrees, most COEs handle hands-on work too.

Common COE responsibilities include:

- Mentoring the internal Power BI community. For more information, see the [Community of practice](#) article.
- Producing, curating, and promoting training materials. For more information, see the [Mentoring and user enablement](#) article.
- Creating documentation and resources to encourage consistent use of standards and best practices. For more information, see the [Mentoring and user enablement](#) article.
- Applying, communicating, and assisting with governance guidelines. For more information, see the [Governance](#) article.
- Handling and assisting with system oversight and administration. For more information, see the [System oversight](#) article.
- Responding to user support issues escalated from the help desk. For more information, see the [User support](#) article.
- Developing solutions and/or proofs of concept.
- Establishing and maintaining the BI platform and data architecture.

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, Power BI.
- Are interested in seeing Power BI 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 Power BI 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 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 serve also as a coach or mentor.

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 details of additional roles and responsibilities, see the Governance article.
Coach	Coaches and educates others on 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 shared datasets and dataflows to support self-service content creators.
Report creator	Creates and publishes reports and dashboards.
Data engineer	Plans Power BI deployment and architecture, including integration with Azure services and other data platforms. Publishes data assets which are utilized broadly across the organization.
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. Therefore, the roles found for COE members can vary too.

Structuring a COE

The selected COE structure can vary among organizations. The four most common ways you can structure the COE are: centralized, unified, federated, or decentralized. Additional variations exist too.

NOTE

The following terms may 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 is comprised of 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 may 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. It may 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 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.

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 may 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 may have more resources available to them, which may 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 may 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 may 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 metrics of Power BI. For a Power BI 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.

Considerations and key actions

Considerations and key actions you can take to establish or improve your Power BI COE:

- Define the scope of responsibilities for the COE. Once the scope is known, identify the skills and competencies required to fulfill those responsibilities.
- Analyze whether the COE has the required systems and infrastructure in place to meet its goals and scope of responsibilities.
- Determine what 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.
- Identify the internal customers, and any external customers, to be served by the COE. Identify how the COE will generally engage with those customers (push model, pull model, or both).
- If you're starting out with a centralized or decentralized COE, consider how you will scale the COE over time with the unified or federated approach. Plan for any actions to take now that will facilitate future growth.
- Verify the funding plan 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 [communications strategy](#) to educate the Power BI community about the services the COE offers, and how to engage with the COE.

Maturity levels

The following maturity levels will help you assess the current state of your COE:

LEVEL	STATE OF POWER BI CENTER OF EXCELLENCE
100: Initial	<p>One or more COEs exist, or the activities are performed within the BI team or IT. There's no clarity on the specific goals nor expectations for responsibilities.</p> <p>Requests for assistance from the COE are handled in an unplanned manner.</p>
200: Repeatable	<p>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 BI while reducing the risks.</p> <p>Goals, scope, staffing, structure, and funding are established for the COE.</p>
300: Defined	<p>The COE operates with active involvement from all business units in a unified or federated mode.</p>
400: Capable	<p>The goals of the COE align with organizational goals, and they are reassessed regularly.</p> <p>The COE is well-known throughout the organization, and consistently proves its value to the internal user community.</p>
500: Efficient	<p>Regular reviews of key performance indicators evaluate COE effectiveness.</p> <p>Agility and implementing continual improvements from lessons learned are top priorities for the COE.</p>

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn about implementing governance guidelines, policies, and processes.

Also, consider reading about [Microsoft's journey and experience with driving a data culture](#). This article describes the importance of *discipline at the core* and *flexibility at the edge*. It also shares Microsoft's views and experiences about the importance of establishing a COE.

Power BI adoption roadmap: Governance

3/24/2022 • 24 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI 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 Power BI, 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."

Although it is called *data governance* it's really a misnomer. The primary focus for governance isn't on the data itself, but on governing *what people do with the data*. Put another way: the true focus is on governing people's behavior to ensure organizational data is well-managed.

When focused on self-service business intelligence, the primary goals of governance are to:

- Empower the internal user community to be productive and efficient.
- Comply with the organization's industry, governmental, and contractual regulations.
- 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. With a platform like Power BI, you'll be most successful 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 BI platform such as Power BI include:

- Empowering people throughout the organization to use data and make decisions, within the defined boundaries.
- Ensuring that the data usage is appropriate for the needs of the business.

- Ensuring that data ownership and stewardship responsibilities are clear. For more information, see the [Content ownership and management](#) article.
- Improving the user experience by providing clear and transparent guidance (with minimal friction) on what actions are permitted, why, and how.
- 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 [System oversight](#) 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.

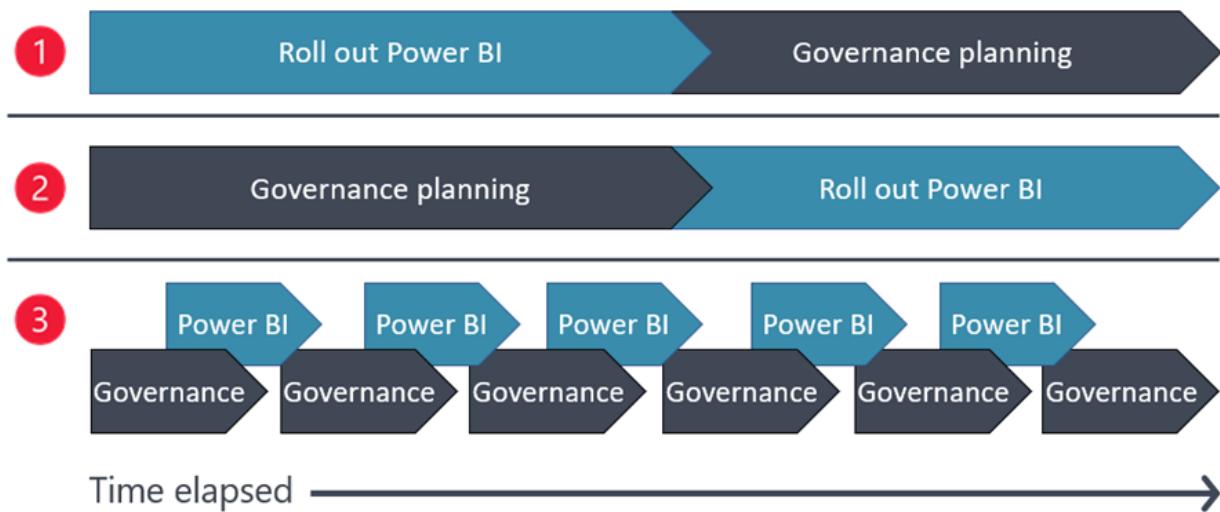
Governance success factors

Governance is not well-received when it's enacted with top-down mandates that are focused more on control than empowerment. Governing Power BI 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, and then takes action to formalize and scale out those methods based on lessons learned.
- 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 people to do the right thing in the right way with little friction.
- The answer to new requests for data is not "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 and creators can—and cannot—do are not made solely by an administrator or in isolation.

Introducing governance to your organization

There are three primary timing methods organizations take when introducing Power BI governance to an organization.



The methods in the above diagram include:

METHOD	STRATEGY FOLLOWED
1	Roll out Power BI first, then introduce governance: Power BI is made widely available to users in the organization as a new self-service 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 Power BI: Extensive governance planning occurs prior to permitting users to begin using Power BI. This method prioritizes control and stability.
3	Iterative governance planning with rollouts of Power BI in stages: Just enough governance planning occurs initially. Then Power BI 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 Power BI is already used for self-service scenarios, and you are 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 Power BI.

Choose Method 3 when you want to provide the greatest degree of flexibility and agility. This balanced approach is the best choice for most organizations and most scenarios.

Method 1: Roll out Power BI first

Method 1 prioritizes agility and speed. It allows users to quickly get started creating solutions. This method occurs when Power BI has been made widely available to users in the organization as a new self-service 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 Power BI is used prevalently throughout the organization.
- Resistance from self-service users who are asked to change what they've been doing.
- In the absence of a strategic plan, self-service users are required to figure out things on their own.

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 Power BI. This situation is most likely to occur when the implementation of Power BI is led by IT. It is also likely to occur when the organization operates in a highly-regulated industry, or when an existing data governance board exists that imposes significant prerequisites and requirements.

Pros:

- More fully prepared to meet regulatory requirements.
- More fully prepared to support the user community.

Cons:

- Favors enterprise BI more than self-service BI.
- 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 Power BI development projects that deliver value.

Pros:

- Puts equal priority on governance and user productivity.
- Emphasizes a *learning as you go* mentality.
- Encourages rolling out to groups in stages.

Cons:

- Requires a high level of communication to be successful with agile governance practices.
- This level of agility 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 Power BI without a governance approach or strategic direction (as described above by Method 1), there could be numerous challenges requiring attention. Depending on the approach you've taken and your current state, some of the following challenges may 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 self-service content.
- Inconsistent reports produced without data validation.
- Valuable data not used or difficult to access.
- Fragmented, siloed, and duplicated datasets.
- 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 may affect your organization, so you can start thinking about what the right solution is for your circumstances.

Governance planning

For organizations that have implemented Power BI without a governance approach or strategic direction (as described above by Method 1), 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 is 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 BI and enterprise BI.

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 with your assessment.

Some potential governance planning activities and outputs that you may find valuable are described next.

Strategy

Key activities:

- Assess current state of data culture, adoption, and BI practices.
- Conduct a series of information gathering sessions to define the desired future state, strategic vision, priorities, and objectives for data culture, adoption, and BI practices. Be sure to include adoption goals for Power BI as suggested in the [Power BI adoption framework series](#). They are a useful approach if you don't already have a structured method for information gathering.
- 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 support is sufficient to sustain and grow the program.
- Clarify where Power BI fits in to the overall data 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. These are 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 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 may 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 industry guidance found in the [Roadmap conclusion article](#) for some helpful resources.

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 BI content?** The [Content ownership and management](#) article introduced three types of strategies: business-led self-service BI, managed self-service BI, and enterprise BI. Who owns and manages the content has a significant impact on governance requirements.
- **What is the scope for delivery of the BI content?** The [Content delivery scope](#) article introduced four scopes for delivery of content: personal BI, team BI, departmental BI, and enterprise BI. 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 may 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 Power BI content.

Key Power BI 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 may choose to prioritize when introducing governance for Power BI:

- Recommendations and requirements for [content ownership and ownership](#).
- 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.
- Allowed activities with regulated data and highly sensitive data.
- Allowed use of unverified data sources that are unknown to IT and/or recommendations for manually maintained data sources.
- How to manage [workspaces](#) effectively.
- Who is allowed to be a [Power BI administrator](#).
- [Security](#), privacy, and data protection requirements, and allowed actions for data artifacts 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 may [certify](#) data artifacts, as well as requirements which 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 to ensure consistency.
- Recommendations and requirements for use of [external tools](#).

If you don't make governance decisions and communicate them well, people 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 cannot do. You may call it something different, but the goal remains the same: when decisions—such as those discussed in the previous section—are made, they are 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 may choose to prioritize:

POLICY	DESCRIPTION
Data ownership policy	Specifies when an owner is required for a data artifact, 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 a data artifact. Requirements may 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 align with data protection policies that are discussed in the System oversight article. It should specify activities such as: allowed sharing with external users (with or without NDA), encryption requirements, and ability to download the data artifact. Sometimes, it is also called a <i>data handling policy</i> or a <i>data usage policy</i> .

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 ensure that governance guidelines and policies are followed. Auditing and monitoring activities are also important. For information about these activities, see the [System oversight](#) article.

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 which are siloed.
- Potential for conflicting information.
- Difficult to scale.

TIP

Finding the right balance of standardization and customization for supporting self-service 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 significantly between organizations. In larger organizations there may 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 may 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.



The levels in the above diagram include:

LEVEL	DESCRIPTION
1	Operational - Business units: Level 1 is the foundation of a well-governed system, which includes people within the business units performing their work. Self-service BI creators have a lot of responsibility related to authoring, publishing, sharing, security, and data quality. Self-service 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 BI, the data governance office, as well as other ancillary teams. Ancillary teams can include IT, security, HR, and legal. Groups, like a change control board, are included here as well.
3	Tactical - Audit and compliance: Level 3 includes internal audit, risk management, and compliance teams, which provide guidance to levels 1 and 2. They also provide enforcement when necessary.
4	Strategic - Executive sponsor and steering committee: The top 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 people 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 people 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 below. In some cases, the same person may serve multiple roles. For instance, the Chief Data Officer (CDO) may also be the executive sponsor.

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 <i>and</i> 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 management office	Manages individual governance projects and the ongoing data governance program.
Power BI executive sponsor	Promotes adoption and the successful use of Power BI. Actively ensures that Power BI decisions are consistently aligned with business objectives, guiding principles, and policies across organizational boundaries.
Center of Excellence	Mentors the community of creators and consumers to promote the effective use of Power BI for decision-making. Provides cross-departmental coordination of Power BI activities to improve practices, increase consistency, and reduce inefficiencies. For more information, see the Center of Excellence article.

ROLE	DESCRIPTION
Power BI champions	A subset of content creators found within the business units who help advance the adoption of Power BI. They contribute to data culture growth by advocating the use of best practices and actively assisting colleagues.
Power BI administrators	Day-to-day-system oversight responsibilities to support the internal processes, tools, and people. Handles monitoring, auditing, and management.
Information technology	Provides occasional assistance to Power BI administrators for services related to Power BI, such as Azure Active Directory, 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

Considerations and key actions you can take to establish or strengthen your governance initiatives:

- Confirm that the high-level goals and guiding principles of the data culture goals are clearly documented and communicated, to ensure that alignment exists for any new governance guidelines or policies.
- Ensure that you have a deep understanding of how Power BI is currently used for self-service BI and enterprise BI. Document opportunities for improvement. Also, document strengths and good practices that would be helpful to formalize.
- 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 implementing the 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.
- Determine if there is an inventory of all critical data assets, or create one if necessary. You cannot govern what you don't know about.
- Confirm that you have support and sufficient attention from your [executive sponsor](#), as well as from business unit leaders.
- Evaluate the existing processes and policies you can immediately adopt for Power BI for consistency. They could include guidance from an existing data governance board or an existing change management board.
- Determine the cadence for how often data policies are reevaluated.

- Determine how conflicts, issues, and requests for exceptions to documented policies will be handled.
- Prepare an action plan that includes:
 - 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.

Maturity levels

The following maturity levels will help you assess the current state of your governance initiatives:

LEVEL	STATE OF POWER BI GOVERNANCE
100: Initial	<p>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.</p> <p>There's a significant reliance on undocumented tribal knowledge.</p>
200: Repeatable	<p>Certain areas of the organization have made a purposeful effort to standardize, improve, and document their data management and governance practices.</p>
300: Defined	<p>An approved plan with governance focus, objectives, and priorities is in place and broadly communicated.</p> <p>Roles and responsibilities are documented and understood.</p> <p>Governance guidelines and policies are implemented for the top few priorities (pain points or opportunities) and are actively followed by the COE and the community of users.</p>
400: Capable	<p>Learnings from existing practices are continually enacted and scaled throughout the organization.</p> <p>It's clear where Power BI fits in to the overall BI strategy for the organization.</p>
500: Efficient	<p>All Power BI governance priorities align with business objectives.</p> <p>Measurable governance goals are clear and tracked regularly for iterative, continual progress. Transparency and communication are a priority.</p>

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn more about mentoring and user enablement.

Power BI adoption roadmap: Mentoring and user enablement

3/24/2022 • 21 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI 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 the [Power BI adoption maturity levels](#) article.

Skills mentoring

Mentoring and helping users in the Power BI 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. Since office hours are group-based, Power BI champions and other members of the community can also pitch in to 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. The primary goal is usually to get questions answered 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, people may or may 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 may 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 people with specific skills that the COE didn't previously know about.
- The COE can learn what people 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. Set clear expectations for what's in scope for office hours, and if there's any commitment for follow up.

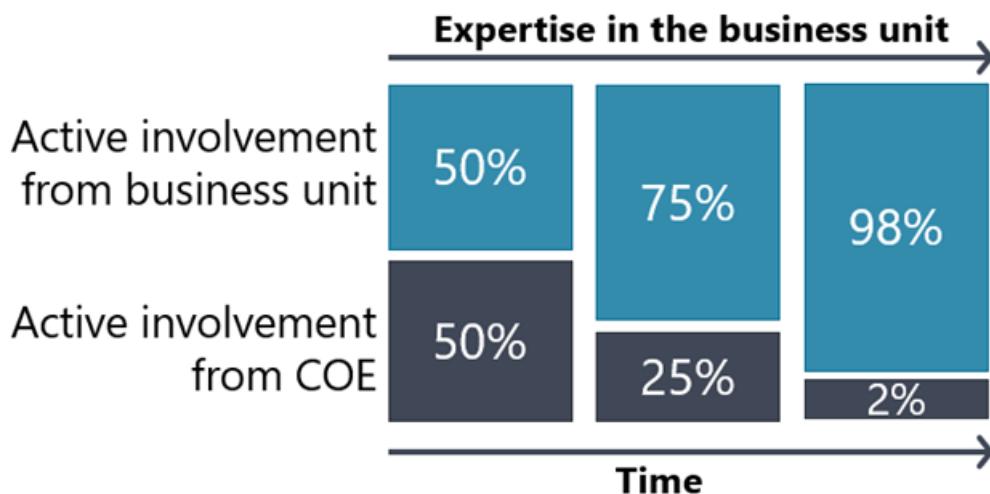
Co-development projects

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 BI solution that the business stakeholders could not 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 for 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 BI solutions in the future.

IMPORTANT

Essentially, a co-development project helps less experienced users learn the right way to do things. It reduces 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 may 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*.

During a review, an expert from the COE evaluates self-service Power BI content developed by a member of the community and identifies areas of risk or opportunities for improvement. The following bullet list presents some examples of when a best practices review could be beneficial:

- The sales team has an [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 Premium capacity](#). A review of the workspace content is required to ensure sound development practices were followed. This type of review is common when the capacity is shared among multiple business units. (A review may not be required when the capacity is assigned to only one business unit.)
- The operations team is creating a new 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 dataset design, though the review can encompass all types of artifacts (dataflows, datasets, reports, or apps).

Before content is deployed to the Power BI service, a best practices review may verify that:

- [Data sources](#) used are appropriate and [query folding](#) is invoked whenever possible.
- [Connectivity mode](#) and [storage mode](#) choices (for example, import, live connection, DirectQuery 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](#)).
- [Data preparation](#) steps are clean, orderly, and [efficient](#).
- [Datasets](#) are well-designed, clean, and understandable (a [star schema](#) design is highly recommended).
- [Relationships](#) are configured correctly.
- [DAX calculations](#) use efficient coding practices (particularly if the data model is large).
- The dataset 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 to the Power BI service, the best practices review is not necessarily complete yet. Completing the remainder of the review may also include items such as:

- The target [workspace](#) is suitable for the content.
- [Workspace security roles](#) are appropriate for the content.
- Other permissions ([app permissions](#), [build permission](#), 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.
- Artifact [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 [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 may 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 is an extraordinarily powerful tool, providing data preparation and data modeling capabilities in addition to data visualization. The complexity of the tool inherently means that there's a significant learning curve to develop mastery. 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 data artifacts 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.
- Templates.
- 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 people might naturally evolve to become your Power BI 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 important 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 may 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 contain 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 users in a Power BI community is training. It is important that the right training resources are readily available and easily discoverable. While some users are so enthusiastic about Power BI 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 community users 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 are 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 people learn new skills while helping them avoid bad habits. It can be a balancing act. For instance, you don't want to overwhelm people 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 dataset. By teaching this concept at the earliest logical time, you can save a less experienced creator thinking they always need one dataset for every report (and encourage the good habit of reusing existing datasets 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 is 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 may target for training include:

- Content consumers.
- Report creators.
- Data creators (datasets and dataflows).
- Content owners, subject matter experts, and workspace administrators.
- Satellite COE members and the champions network.
- Power BI 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. If you have a diverse population of Power BI 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 grant badges, like *blue belt* or *black belt*, as people progress through the training programs.

Give some consideration to how you want to handle users at various stages of [user adoption](#). Training to onboard new users (sometimes referred to as *training day zero*) and for less experienced users is very different to training for 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 may 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 Power BI 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](#).
- [Power BI courses and "in a day" training materials](#).
- [LinkedIn Learning](#).
- [Virtual workshops and training](#).

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 may choose to provide your user community with a curated set of recommended links to trusted online sources. There is a wide array of videos, blogs, and articles produced by the worldwide community. The community comprises 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 Power BI service is customizable. Once 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 Power BI is managed in your organization. For more information, see the [Content ownership and management](#) article.

Certain aspects of Power BI 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 Premium capacity.
- How to request a new workspace.
- How to request a workspace be added to Premium 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.

Other aspects of Power BI 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 a dataset.
- How to request a dataset enhancement.

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 Power BI administrator role (who might have this role solely for the purpose of viewing settings).

Over time, you may choose to allow some documentation to be maintained by the community if you have willing volunteers. In this case, you may 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 may be appropriate. When the documentation exists, it allows colleagues to reference it when needed. It contributes to user enablement and a self-sustaining community.

TIP

When creating custom documentation or training materials, reference existing Microsoft sites using links when possible. Since Power BI is in a continual state of evolution, it will reduce the level of documentation maintenance needed over time.

Templates

A [Power BI template](#) is a `.pbtt` 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 the community, 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.
- Datasets 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 dataset 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.

Considerations and key actions

Considerations and key actions you can take to establish, or improve, mentoring and user enablement:

- Establish regular office hours, ideally at least once per week (depending on demand from users as well as staffing and scheduling constraints).
- Decide how you will communicate and advertise office hours to the user community.
- Decide what the expectations will be for office hours, including allowed topics or types of issues users can bring, how the queue of requests will work, if any information should be submitted ahead of time, and whether any follow up afterwards can be expected.
- Beyond office hours, consider what other types of mentoring services the COE could offer, such as co-development projects or best practices reviews.
- Create a centralized portal to serve as the hub for Power BI training, 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.
- Compile a curated list of reputable 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 what custom in-house training resources will be useful and worth the time investment.
- Create and promote a top 3-5 list of documentation and resources that will be most useful to the user community.

Maturity levels

The following maturity levels will help you assess the current state of your mentoring and user enablement:

LEVEL	STATE OF POWER BI MENTORING AND USER ENABLEMENT
100: Initial	<p>Some documentation and resources exist, though they are siloed and inconsistent.</p> <p>Few users are aware of, or take advantage of, available resources.</p>
200: Repeatable	<p>A centralized portal exists with a library of training, documentation, and resources.</p> <p>Office hours are held on a regular basis.</p>
300: Defined	<p>The centralized portal is the primary hub for community members to locate information.</p> <p>The COE's skills mentoring program is in place to assist users in the community in various ways.</p>
400: Capable	<p>Resources in the centralized portal are commonly referenced by champions and community members when supporting and learning from each other.</p> <p>Business units regularly engage with the COE and take advantage of its skills mentoring program. Tangible business value is gained from the program.</p>
500: Efficient	Documentation and resources are continually updated and improved by the COE, based on lessons learned.

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn more about the community of practice.

Power BI adoption roadmap: Community of practice

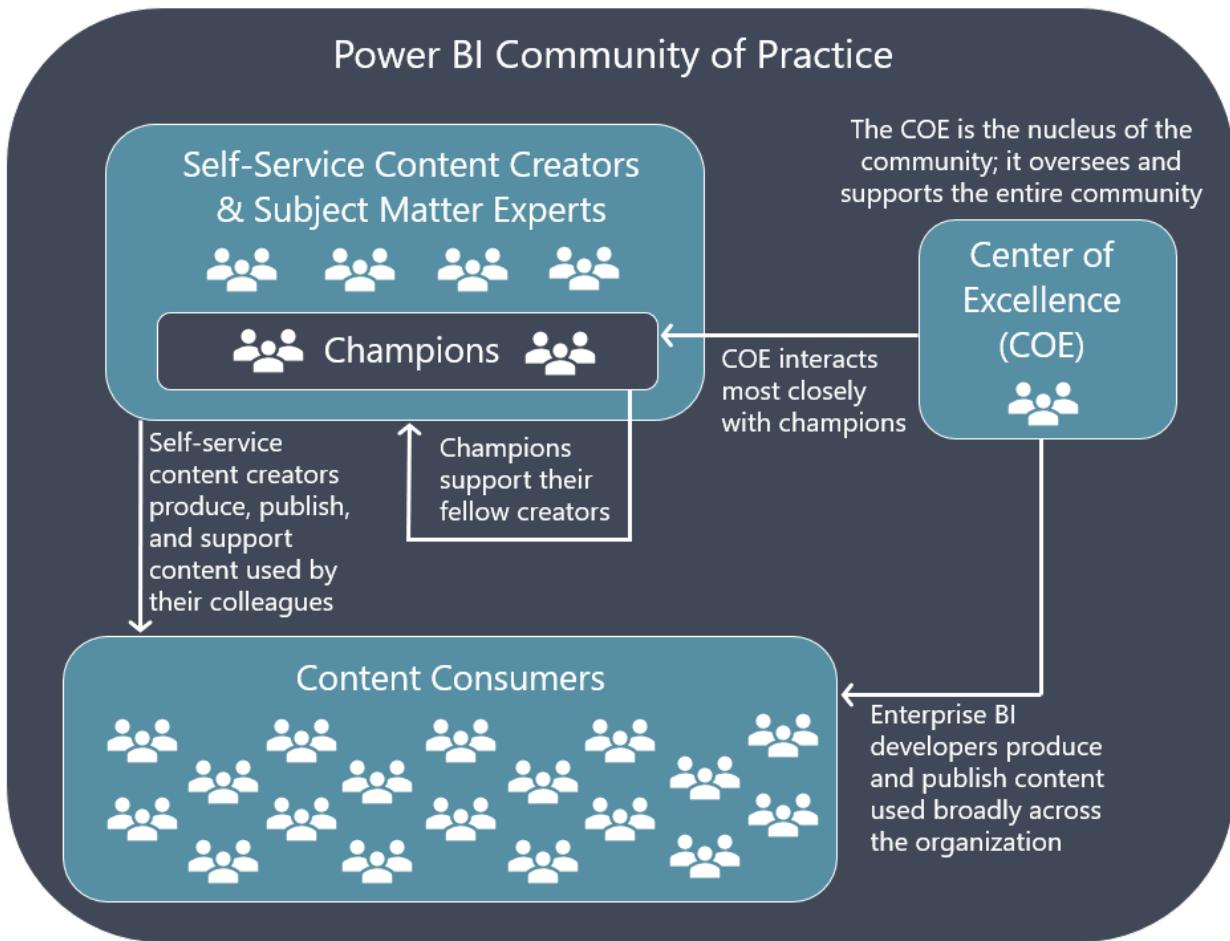
3/24/2022 • 12 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI 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 Power BI 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 Power BI.
- The **Center of Excellence (COE)** forms the nucleus of the community. It 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 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 Power BI 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.

NOTE

All references to the Power BI 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 Power BI. However, internal users are the focus of this article.

For information about related topics including resources, documentation, and training provided for the Power BI 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 Power BI content creator who works in a business unit that engages with the COE. A champion is recognized by their peers as the *go-to* Power BI expert. A champion continually builds and shares their knowledge even if it's not an official part of their job role. Power BI 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 Power BI being used effectively and adopted successfully throughout the organization.
- Possess strong Power BI 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.

IMPORTANT

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 in an internal discussion channel or participating in lunch and learns.

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 may be acting in the role of a champion without even knowing it, and without a formal recognition. The COE should always be on the lookout for champions. COE members should actively monitor the discussion channel to see who is helpful. They 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:

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.
Internal Power BI 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.
Internal Power BI conferences or events	An annual or semi-annual internal conference that delivers a series of sessions focused on Power BI.

IMPORTANT

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 peers.
- **Leadership communication:** Reach out to a manager when someone goes above and beyond so that their leader, who may not be active in the Power BI community, sees the value that their staff member provides.

TIP

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.

Rewarding champions

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 during the previous month. 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 *Power BI test* before a user is granted a Power BI license. This *test* is a misnomer because it doesn't focus on any Power BI skills. Rather, it verifies 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 new employees to get started with Power BI. 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 Power BI license or

are getting involved with the Power BI community. 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 Power BI users, or subsets of users. It can include announcing changes that are planned to key organizational content. For example, changes are to be published for a critical shared dataset that's used heavily throughout the organization. It can also include the announcement of new features from the [Microsoft Power BI blog](#) and [Microsoft Power BI release plan](#) updates. For more information about planning for change, see the [System oversight](#) article. Feature announcements 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 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

Considerations and key actions you can take to initiate, grow, and sustain your champions network include:

- Clarify your specific goals for cultivating a champions network. Make sure these goals align with your overall Power BI strategy, and that your executive sponsor is on board.
- Create a plan to support 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.
- Decide what level of commitment and expected time investment will be required of Power BI champions. (Note that the time investment can vary wildly from person to person, and team to team.) Plan to clearly communicate expectations to people who are interested to get involved. Obtain manager approval when appropriate.
- 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).
- 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](#).
- Determine how members of the champions network will be managed. It could be implemented through membership in 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
- Ensure champions have the resources they need, including:
 - Direct access to COE members.
 - Influence on data policies being implemented (for example, requirements for a dataset certification policy).
 - Influence on the creation of best practices and guidance (for example, recommendations for accessing a specific source system).
- Actively involve certain champions as satellite members of the COE. For more information about federating the COE, see the [Center of Excellence](#) article.
- Create a feedback loop so champions 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.

Considerations and key actions you can take improve knowledge sharing:

- 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.
- Verify who will take responsibility for coordinating all knowledge sharing activities.

Considerations and key actions you can take to introduce incentives:

- Consider what type of incentives you could offer to your champions network.
- Consider what type of incentives you could offer to your broader internal community.

Considerations and key actions you can take improve communications:

- Evaluate which methods of communication fit well in your data culture.
- Determine who will be responsible for different types of communication, how, and when.
- Set up different ways to communicate, including history retention and search.

Maturity levels

The following maturity levels will help you assess the current state of your community of practice:

LEVEL	STATE OF POWER BI COMMUNITY
100: Initial	<p>Some content creators do great work, but their efforts are unrecognized.</p> <p>Efforts to share knowledge are rare and unstructured.</p> <p>Communication is inconsistent.</p>
200: Repeatable	<p>The first set of champions are identified, and champions network goals are established.</p> <p>Knowledge sharing practices gain traction and are now more consistent.</p>
300: Defined	<p>Best practices are actively shared across the organization.</p> <p>Knowledge sharing in multiple forms is a normal and regularly scheduled occurrence.</p> <p>Goals for transparent communication with the user community are defined and executed with regularity.</p>

LEVEL	STATE OF POWER BI COMMUNITY
400: Capable	<p>Champions are established for all business units and actively support colleagues in their self-service efforts.</p> <p>Incentives to recognize and reward knowledge sharing efforts are a common occurrence.</p> <p>Regular and frequent communication occurs based on a predefined communication plan.</p>
500: Efficient	Bidirectional feedback loops exist between the champions network and the COE.

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn about user support.

Power BI adoption roadmap: User support

3/24/2022 • 15 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI 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 Power BI 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:

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
6	Worldwide community support

TYPE	DESCRIPTION
5	Microsoft support (external) includes support for licensed users and administrators. It also includes comprehensive Power BI documentation .
6	Community support (external) includes the worldwide community of Power BI 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 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 BI (content is owned and managed by a centralized business intelligence team or Center of Excellence). In some organizations, all four types of support could be relevant for any type of content.

Each of the four types of internal 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. People who emerge as your [Power BI 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 should not 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 Power BI 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 forum set up specifically for this purpose. For example, someone posts a message that they're having problems getting a DAX calculation to work. They then receive a response from someone in the organization with suggestions or links.

TIP

The goal of an internal Power BI community is to be self-sustaining, which can lead to reduced formal support demands and costs. It can also facilitate managed self-service BI occurring at a broader scale versus a purely centralized BI 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 [Data Analysis eXpressions \(DAX\)](#) and the [Power Query M formula language](#). When someone becomes a recognized expert, they may become overburdened with too many requests for help.
- A greater number of community members may readily answer certain types of questions (for example, report visualizations), whereas a smaller number of members will answer others (for example, complex 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](#), which brings people together based on a common interest, 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 Power BI experts and potential champions who were previously unknown to the COE.

IMPORTANT

It's a best practice to continually identify emerging Power BI 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. It should deliberately encourage and support them, and if 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 aware that some individuals will not be comfortable asking questions in such a public way because it openly acknowledges what they don't know, which might be embarrassing. This reluctance may reduce over time by promoting a friendly, encouraging, and helpful discussion channel.

TIP

You may 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 Power BI 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, operated by the IT department. People who will likely rely

on a more formal support channel include those who are:

- Less experienced with Power BI.
- 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 which cannot 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, like software installation prerequisites. Some organizations task the help desk to handle only very simple break-fix issues, whereas other organizations have the help desk get involved with anything that is repeatable, like new [workspace](#) requests, managing [gateway data sources](#), or requesting new [Premium capacities](#).

IMPORTANT

Your Power BI 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, usually within 1-4 hours if possible. Consider that automation with Power Apps and Power Automate can help make the process efficient. If you delay too long, users will use what they already have, and that may not be the ideal scenario. Promptness is critical for certain help desk requests.

Over time, troubleshooting and problem resolution skills become more effective as help desk personnel expand their knowledgebase and experience with Power BI. The best help desk personnel are those who have a good grasp of what users need to accomplish with Power BI.

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. For instance, there may be an agreement to respond to blocking issues within one hour and resolve them within eight hours. It's generally more difficult to define service level agreements (SLAs) for troubleshooting issues, like data discrepancies.

Extended support

Since the COE has deep insight into how Power BI 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 Power BI users and administrators that shouldn't be overlooked.

Microsoft documentation

Check the [Power BI support site](#) high-priority issues that broadly affect all customers. Global Microsoft 365

administrators have access to additional support issue details within the Microsoft 365 portal.

Monitor the [Microsoft 365 Twitter account](#). Microsoft posts timely information and updates about outages for all Microsoft 365 services.

Refer to the comprehensive [Power BI documentation](#). It's an authoritative resource that can aid troubleshooting and search for information. You can prioritize results from the Power BI documentation site. For example, enter a site-targeted search request into your web search engine, like "power bi dataset site:docs.microsoft.com".

Power BI Pro and Premium Per User end-user support

Users with a Power BI Pro or Premium Per User license are eligible to [log a support ticket with Microsoft](#).

TIP

Make it clear to your internal user community whether you prefer technical issues 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 [global and Power BI 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 setup to [submit and view support requests](#).

Worldwide community support

In addition to the internal user support approaches discussed in this article, and Microsoft support options discussed previously, you can leverage the worldwide Power BI community. This option is useful when a question can be easily understood by someone not close to the problem, and when it doesn't involve sensitive data.

Publicly available community forums

There are several [public Power BI community forums](#) where users can post issues and receive responses from any Power BI user in the world. It 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.

Publicly available discussion areas

It's very common to see people posting Power BI technical questions on platforms like Twitter. A quick look at the [#PowerBI hashtag](#) reveals a vibrant global community of Power BI enthusiasts. You will find discussions, post announcements, and users helping each other. The [#PowerBIRHelp hashtag](#) is sometimes used, though less frequently.

Community documentation

The Power BI global community is vibrant. Every day, there are a great number of Power BI blog posts, articles, webinars, and videos published. When relying on community information for troubleshooting, watch out for:

- How recent the information is.
- Whether the situation and context of the solution found online truly fits your circumstance.
- The credibility of the information being presented.

Considerations and key actions

Considerations and key actions you can take to improve your intra-team support:

- Provide recognition and encouragement to your Power BI champions, as well as incentives, as described in the [Community of practice](#) article.
- Reward and praise meaningful grassroots efforts when you see them happening.
- If informal intra-team efforts aren't adequate, consider formalizing the roles you want to enact in this area, and the expected contributions and responsibilities.

Considerations and key actions you can take to improve your internal community support:

- Encourage people 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.
- 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.
- Make sure your user population knows the internal community support area exists. It could include the prominent display of links, or including a link in regular communications. You can also [customize the help menu links](#) in the Power BI service to direct users to your internal resources.
- Set up automation to ensure that all your Free, Power BI Pro, and Premium Per User licensed users automatically have access to the community discussion channel. It's possible to automate license setup using [group-based licensing](#).

Considerations and key actions you can take to improve your internal help desk support:

- Determine the initial scope of Power BI topics that the help desk will handle.
- Assess the readiness level of your help desk to handle Power BI support.
- Arrange for additional training for help desk staff, based on readiness gaps.
- Determine what the escalation path will be for requests the help desk cannot directly handle.
- Update the help desk knowledgebase for known Power BI topics. Ensure someone is responsible for regular updates to the knowledgebase to reflect new and enhanced features over time.
- Ensure a good issue tracking system is in place. It's often a ticketing system that can manage priority levels.
- Decide if anyone will be on-call for any issues related to Power BI. If appropriate, ensure the expectations for 24/7 support are clear.
- Determine what SLAs will exist, and that expectations for response and resolution are clearly communicated.
- Be prepared to address specific common issues extremely quickly. For example, a request to add a new [gateway data source](#) should be handled very quickly (to avoid user frustration, and to minimize use of personal gateways as a workaround). Slow support response may result in users finding workarounds.

Considerations and key actions you can take to improve your internal COE extended support:

- Clearly define where help desk responsibilities end, and where COE extended support responsibilities begin.
- Ensure that COE members have a direct escalation path to reach global administrators for Microsoft 365 and Azure. It's critical when a widespread issue arises that's beyond the scope of Power BI.
- Create a feedback loop from the COE back to the help desk so that the IT knowledgebase can 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).

Maturity levels

The following maturity levels will help you assess the current state of your Power BI user support:

LEVEL	STATE OF POWER BI USER SUPPORT
100: Initial	<p>Individual business units find effective ways of supporting each other, though tactics are siloed and not consistently applied.</p> <p>An internal discussion channel is available, but it's not monitored closely so the user experience is inconsistent.</p>
200: Repeatable	<p>The importance of intra-team support is encouraged and actively supported by the COE, including direct support of the champions network.</p> <p>The internal discussion channel gains traction as the default place for Power BI Q&A.</p> <p>The help desk handles a small number of the most common Power BI technical support issues.</p>
300: Defined	<p>The internal discussion channel is now popular and largely self-sustaining. COE members actively monitor and manage the discussion channel to ensure questions are answered quickly and correctly.</p> <p>The help desk is fully prepared to handle all known and expected Power BI technical support issues, and the COE provides appropriate extended support when required.</p>
400: Capable	<p>A recognition program is established, and it generates enthusiasm and encourages sharing best practices.</p> <p>SLAs are in place to define help desk support expectations, including extended support, and they are clear to everyone involved.</p>
500: Efficient	<p>Bidirectional feedback loops exist between the help desk and the COE.</p> <p>Key performance indicators measure community engagement and satisfaction.</p> <p>Automation is in place when it adds direct value to the user experience (for example, automatic access to the community), or for specific help desk activities (for example, use of APIs and scripts that increase speed and reduce error).</p>

Next steps

In the [next article](#) in the Power BI adoption roadmap series, learn about system oversight and administration activities.

Power BI adoption roadmap: System oversight

3/24/2022 • 35 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

System oversight—also known as Power BI administration—is the ongoing, day-to-day, administrative activities that:

- Enact governance guidelines and policies to support self-service BI and enterprise BI.
- 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.
- Allow for broader organizational adoption of Power BI with effective governance and data management practices.

IMPORTANT

Your organizational [data culture](#) objectives provide direction for your [governance](#) decisions, which in turn dictate how [Power BI administration](#) activities take place and by whom.

Administration 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.

Power BI administrators

The Power BI administrator role is a defined role in Microsoft 365, which delegates a [subset of Power BI-specific](#) management activities. Global Microsoft 365 administrators are implicitly Power BI administrators.

A key governance decision is who to assign as a Power BI administrator. It's a centralized role which affects your entire Power BI tenant. Ideally, there are [2-4 people in the organization](#) who are capable of managing the Power BI service, and who are in close coordination with the [Center of Excellence \(COE\)](#).

High privilege role

The Power BI administrator role is considered a high privilege role because:

- Settings that are managed by a Power BI administrator have a significant effect on user capabilities and user experience (described in the [Tenant settings](#) section below).
- Power BI administrators can update access permissions for any workspace (excluding personal workspaces and [classic workspaces](#)). The result is that an administrator can allow permission to view or download data artifacts as they see fit (described in the [Tenant settings](#) section below).
- Power BI administrators can view all tenant metadata, including all user activities that occur in the Power BI service (described in the [auditing and monitoring](#) section below).

IMPORTANT

Having too many Power BI administrators is a risk as it increases the probability of unapproved or unintended changes.

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 BI, managed self-service BI, and enterprise BI. For more information, see the [Content ownership and management](#) article.

TIP

The best type of person to assign as a Power BI administrator is one who has enough knowledge about Power BI to understand what self-service users need to accomplish.

There are several [types of Power BI administrators](#). The following table describes the roles that are used most often on a regular basis:

ROLE	SCOPE	DESCRIPTION
Power BI administrator	Power BI tenant	Manages tenant settings and other aspects of the Power BI service. All general references to <i>administrator</i> in this article refer to this type of administrator.
Power BI Premium capacity administrator	One capacity	Manages workspaces, workloads, and monitors the health of a Premium capacity.
Power BI gateway administrator	One gateway	Manages gateway data source configuration, credentials, and users assignments. May also handle gateway software updates (or collaborate with infrastructure team on updates).
Power BI workspace administrator	One workspace	Manages workspace settings and access.

The Power BI ecosystem is very broad and deep. There are many different ways that the Power BI service integrates with other systems and platforms. From time to time, it will be necessary to work with other system administrators and IT professionals, such as:

- Global Microsoft 365 administrator.
- Azure Active Directory administrator.
- Teams administrator.
- OneDrive administrator.
- SharePoint administrator.
- Database administrator.
- Licensing and billing administrator.
- Intune administrator.
- Desktop support team.
- Infrastructure team.
- Networking team.
- Security and compliance team.

The remainder of this article discusses the most common activities that a Power BI administrator does. It focuses on those that are important to carry out effectively when taking a strategic approach to [Power BI organizational](#)

adoption.

Service management

Overseeing the Power BI service is a crucial aspect to ensure that all users have a good experience with Power BI.

Tenant settings

Proper management of [tenant settings](#) in the Power BI service is critical. Tenant settings are the main way to control which Power BI capabilities are enabled and to which groups of users in your organization.

It's essential that tenant settings align with governance guidelines and policies, and also with how the COE makes decisions. If a Power BI administrator independently decides which settings to enable or disable, that's a clear indicator of an opportunity to improve governance processes.

IMPORTANT

Changing the tenant settings should go through a change control process with an approval mechanism. It should document all changes, recording who made the change, when, and why.

Since content creators and consumers can easily read online about available features in Power BI, it can be very frustrating when capabilities don't function as expected. It can lead to dissatisfied users and less effective [organizational adoption, user adoption, and solution adoption](#). Here's a list of common questions asked by confused and frustrated users:

- 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 dataset?

Caution

If you discover situations that aren't ideal, such as too many data exports in the activity log, resist the urge to disable the feature entirely. Prohibiting features leads to user frustration, and worse, workarounds. Before disabling a setting, find out why users are relying on certain techniques. Perhaps a solution needs to be redesigned, or additional user education and training could mitigate the concerns. The bottom line: knowledge sharing is an effective form of governance.

Since there's no reader role to view tenant settings, it can be a challenge in larger organizations. Consider publishing a document to the centralized portal that describes the tenant settings, as described in the [mentoring and user enablement](#) article.

The following activities apply when reviewing and validating each tenant setting:

- Tenant setting:
 - Enabled, or
 - Disabled
- Tenant setting applicable to:
 - The entire organization, or
 - Limited to specific security group(s):
 - Does a suitable security group already exist?, or
 - Does a new security group need to be created?

Admin portal

As discussed in the Power BI adoption [maturity levels](#) article, organizational adoption refers to the effectiveness of Power BI governance and data management practices to support and enable enterprise BI and self-service BI.

Actively managing all areas of the Power BI service (in addition to the tenant settings) in accordance with adoption goals has a direct influence on ensuring that all users have a good experience with Power BI.

Additional responsibilities for managing the Power BI service include:

- [Workspace management and access](#).
- [Premium capacity and Premium Per User settings](#).
- [Embed codes](#).
- [Organizational visuals](#).
- [Azure connections](#).
- [Custom branding](#).
- [Featured content](#).

In addition to these documentation links, see the [Planning a Power BI enterprise deployment whitepaper](#), which describes additional considerations for Power BI administration.

User machines and devices

The management of user machines and devices is usually a responsibility of the IT department. The adoption of Power BI depends directly on content creators and consumers having the applications they need installed and configured correctly.

The following [software installations](#) are available for content creators:

SOFTWARE	AUDIENCE
Power BI Desktop	Content creators who develop data models and interactive reports for deployment to the Power BI service.
Power BI Desktop Optimized for Report Server	Content creators who develop data models and interactive reports for deployment to Power BI Report Server.
Power BI Report Builder	Content creators who develop paginated reports for deployment to the Power BI service or Power BI Report Server.
Power BI Mobile Application	Content creators or 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.
On-Premises Data Gateway (Personal Mode)	Content creators who publish datasets to the Power BI service and manage scheduled data refresh (see additional description in the Gateway architecture and management section of this article).

IMPORTANT

Not all the listed software will be necessary for all content creators. Power BI Desktop is the most common requirement and is the starting point when in doubt.

It's very important that all content creators who collaborate with others use the same version of the software—especially Power BI Desktop, which is updated monthly. Ideally, software updates are available from the Microsoft Store or installed by an automated IT process. This way, users don't have to take any specific action to obtain updates.

Because new capabilities are continually released, software updates should be released promptly. This way, users can take advantage of the new capabilities, and their experience is aligned to documentation. It's also important to be aware of the [update channel](#). It provides new (and updated) features for Office apps, such as Excel and Word, on a regular basis.

Other common items that may need to be installed on user machines include:

- Drivers to support data connectivity, for example, Oracle, HANA, or the Microsoft Access Database Engine.
- The [Analyze in Excel](#) provider.
- [External tools](#), for example, Tabular Editor, DAX Studio, or ALM Toolkit.
- [Custom data source connectors](#).

In addition to software installations, user machines may be managed for:

- Group policy settings. For example, settings can allow the use of [custom visuals](#) so that the Power BI Desktop experience aligns with the Power BI service to ensure a consistent user experience.
- Registry settings. For example, disable the Power BI Desktop [sign-in form](#) or [tune Query Editor performance](#).

TIP

Effective management of software, drivers, and settings can make a big difference to the user experience, and that can translate to increased [user adoption](#) and satisfaction, and reduced [user support](#) costs.

Architecture

Data architecture

Data architecture refers to the principles, practices, and methodologies that govern and define what data is collected, and how it is 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 significantly impact on Power BI adoption, user satisfaction, and individual project success rates.

A few data architecture considerations that affect adoption of Power BI include:

- Where does Power BI fit into the organization's entire data architecture? And, 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 Power BI used end-to-end for data preparation, data modeling, and data presentation? Or, is Power BI used only some of those capabilities?
- Where will users consume the content? Generally, the three main ways to deliver content are: the Power BI service, Power BI Report Server, and embedded in custom applications. The [Planning a Power BI enterprise deployment whitepaper](#) includes a section on Power BI architectural choices, which describes when to consider each of these three main choices. Additionally, [Microsoft Teams](#) is a convenient alternative to the Power BI service, especially 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, and what types of data will we be acquiring?

- What [connectivity mode](#) and [storage mode](#) choices (for example, import, live connection, DirectQuery, or composite model frameworks) are the best fit for the use cases?
- To what extent is data reusability encouraged using [shared datasets](#)?
- To what extent is the reusability of data preparation logic encouraged using [dataflows](#)?

When becoming acquainted with Power BI, many system administrators assume it's a query tool much like SQL Server Reporting Services (SSRS). The breadth of capabilities for Power BI, however, are vast in comparison. So, it's important for administrators to become aware of Power BI capabilities 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. 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 discussed in the [Mentoring and user enablement](#) article, are another useful way to help content creators with important architectural decisions.

Premium capacity management

[Power BI Premium](#) includes additional features and capabilities to deliver BI solutions at scale. Premium subscriptions may be purchased [by capacity or per user](#) with Premium Per User (PPU). This section primarily focuses on Premium capacity, which requires additional oversight.

Power BI Premium can play a significant role in your BI strategy. Some top reasons to invest in Premium include:

- [Unlimited content distribution](#) to large numbers of read-only (content consumption with a free Power BI license is available in Premium capacity only, not PPU).
- [Deployment pipelines](#) to manage the publication of content to development, test, and production workspaces. They are highly recommended for critical content to improve release stability.
- [Paginated reports](#) to deliver highly-formatted, pixel-perfect reports. This report type allows content creators to meet additional types of information delivery requirements.
- [XMLA endpoint](#), which is an industry standard protocol for managing and publishing a dataset, or querying the dataset from any XMLA-compliant tool.
- Increased model size limits, including [large dataset](#) support.
- More frequent [data refreshes](#).
- Storage of data in a specific geographic area ([multi-geo](#) is available by capacity only).

This list is not all-inclusive. For a complete list of Premium features, see [Power BI Premium FAQ](#).

Managing Premium capacity

Overseeing the health of Power BI Premium capacity is an essential ongoing activity for administrators because, by definition, Premium capacity includes a fixed level of system resources. It equates to memory and CPU limits that must be managed to achieve optimal performance.

Caution

Lack of management and exceeding the limits of Premium capacity 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 Premium capacity:

- Create a specific set of criteria for content that will be published to Premium capacity. It's particularly relevant when a single capacity is used by multiple business units because the potential exists to disrupt other users if the capacity is not well-managed. For a list of items that may be included in the best practices review (such as reasonable dataset size and efficient calculations), see the [Mentoring and user enablement](#) article.
- Regularly use the [Premium monitoring app](#) to understand resource utilization and patterns for the Premium

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.

- Configure the [tenant setting](#) so Power BI notifies you if the [Premium capacity becomes overloaded](#), or an outage or incident occurs.

Autoscale

Autoscale is a capability of [Power BI Premium Gen 2](#) that's intended to handle occasional or unexpected bursts in Premium usage levels. It 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 Premium capacity is not well-managed, autoscale may trigger more often than expected. In this case, the [Premium monitoring app](#) can help you to determine underlying issues.

Decentralized Premium capacity management

Capacity administrators are responsible for [assigning workspaces](#) to a specific capacity. Note 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.

It's possible to set up multiple capacities to facilitate decentralized management by different business units. Decentralizing management of certain aspects of Power BI is a great way to balance agility and control. Here's an example to describe how it could be managed for Premium capacity:

- Purchase a P3 capacity node in Microsoft 365, which includes 32 virtual cores.
- Use 16 cores to create the first capacity—it will be used by the Sales team.
- Use 8 cores to create the second capacity—it will be used by the Operations team.
- Use the remaining 8 cores to create the third capacity—it will support general use.

This example has several advantages:

- Separate [capacity administrators](#) may be configured for each capacity, and so it facilitates decentralized management situations.
- If a capacity is not well-managed, the effect is confined to that capacity only. The other capacities are not impacted.

However, the example has disadvantages, too:

- The [limits per capacity](#) are lower. The maximum memory size allowed for datasets isn't the entire P3 capacity node size; rather, it's the assigned capacity size where the dataset is hosted.

Gateway architecture and management

The [on-premises data gateway](#) facilitates the secure and efficient transfer of data between organizational data sources and the Power BI service. A 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 the user. It supports use by one user only. It does not

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 datasets 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 options 1 or 3 should be strongly encouraged over option 2 because they are more scalable and manageable.

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 use the gateway data source.

Certain aspects of gateway management can be done effectively on a decentralized basis to balance agility and control. For example, the Operations group may 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 encryption keys.
- Monitoring of gateway logs to assess when scale-up or scale-out is necessary.
- Alerting of downtime or persistent low machine resources.

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 of the Power BI service needs a commercial license which is integrated with an Azure Active Directory identity. The user license may be Free, Power BI Pro, or Power BI Premium Per User. A user license is obtained via a subscription which authorizes a certain number of licenses with a start and end date.

There are two approaches to procuring subscriptions:

- **Centralized:** Microsoft 365 billing administrator purchases a subscription for [Power BI Pro or Premium Per User](#). 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 to the Power BI service very closely.

Trials

Another important governance decision is whether user trials are allowed. By default, trials are enabled. That means when content is shared with a colleague, if the recipient does not have a Power BI Pro or Premium Per User license, they will be prompted to start a trial to view the content (if the content doesn't reside within Premium capacity). The trial experience is a great convenience and allows people to continue with their normal workflow.

Generally, disabling trials is not recommended. It can encourage users to apply 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), and 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 Power BI service very closely.

TIP

Don't introduce too many barriers to obtaining a Power BI license. People who need to get work done will find a way, and that way may involve workarounds that aren't ideal. For instance, without a license to use the Power BI service, people may 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 Power BI, is an important activity. Here are several activities you may want to consider:

- Analyze who is using—and, more to the point, not using—their allocated Power BI licenses and make necessary adjustments. Power BI usage is analyzed using the [activity log](#).
- Analyze the cost effectiveness of [Premium capacity](#) or [Premium Per User](#). In addition to the [additional features](#), perform a cost/benefit analysis to determine whether Premium licensing is more cost-effective when there are a large number of consumers (unlimited content distribution is only available with Premium capacity, not PPU licensing).
- Carefully [monitor and manage Premium capacity](#). Understanding usage patterns over time will allow you to predict when to purchase [additional capacity](#). For example, you may 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 [Power BI Premium Gen 2](#) is recommended. It will scale up capacity resources for 24 hours, then scale them back down to normal levels (provided that sustained activity isn't present). Manage autoscale cost by constraining the maximum number of v-cores, and/or with spending limits set in Azure (because autoscale is supported by the Azure Power BI Embedded service). 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 Power BI tenant whenever possible. It will avoid incurring [Azure egress charges](#), which are minimal, but at scale can be considerable.

Security and data protection

Security and data protection are joint responsibilities among all content creators, consumers, as well as 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 may have a big 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 Power BI users to accept a self-service user acknowledgment. This is a document that explains the user's responsibilities and expectations for safeguarding organizational data. One way to automate its implementation is with an [Azure Active Directory terms of use policy](#). The user is required to agree to the policy before they are permitted to visit the Power BI service for the first time (or on a recurring basis, like an annual renewal).

Data security

In a [cloud shared responsibility model](#), securing the data itself is always the responsibility of the customer. With a self-service BI platform, self-service content creators have responsibility for properly securing the content that's 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, and by raising 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 very common occurrence for some organizations, and very rare for others. How you handle external users is a governance decision.

External user access is controlled by [tenant settings](#) in the Power BI service as well as certain Azure Active

Directory settings. For details of external user considerations, review the [Distribute Power BI content to external guest users using Azure Active Directory B2B](#) whitepaper.

Information protection

Power BI supports capabilities for information protection and data loss prevention through its integration with:

- [Microsoft Integration Protection](#) (MIP), which is a collection of features and capabilities with an objective to discover, classify, and protect sensitive information. Its philosophy is to *know your data, protect your data, prevent data loss, and govern your data*.
- [Microsoft Defender for Cloud Apps](#) (Defender for Cloud Apps), which is a cloud access security broker (CASB). It can audit, monitor, and raise alerts based on certain activities. See the [monitoring](#) section later in this article for examples of how Defender for Cloud Apps can be used for oversight of the Power BI service.

Power BI's information protection capabilities are built upon [sensitivity labels](#). Sensitivity labels are an important building block for data protection, data retention, data loss prevention, compliance, and insider risk management for Microsoft 365 services (including Power BI, as well as other Microsoft services).

IMPORTANT

A sensitivity label offers powerful data protection capabilities. However, it's not a replacement for standard data security practices, such as workspace roles, app security, individual item sharing, or row-level security.

The power of sensitivity labels is:

- Automated data loss prevention within the Power BI service, for instance, when [Defender for Cloud Apps](#) can invoke a policy to prohibit a file download based on a sensitivity label.
- Automated data loss prevention across system boundaries, such as when the [label follows the content](#) from when it's exported from the Power BI service to Excel or PowerPoint.
- User education, so users know what they can and cannot do with the data. It's not automated. Rather, it should be handled with a data governance policy and user education.

There are several [tenant settings](#) which relate to information protection. For more information, see the [Auditing and monitoring](#) section in this article.

Data residency

For organizations with requirements to store data within a geographic region, Premium capacity (not PPU) can be [configured for a specific region](#) that's different from the region of the Power BI home 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 cannot be exposed.

Note that [Premium Per User](#) only supports BYOK when it's enabled for the entire Power BI tenant.

Auditing and monitoring

A wealth of metadata is available for understanding what's happening within your Power BI tenant. The primary source of information is from the Power BI [activity log](#), which captures information about many different types of activities that users perform.

There are also a variety of [REST APIs](#) that provide additional information about workspaces, apps, datasets, and more. Of particular interest to administrators are the [admin APIs](#). These APIs provide a means to extract

metadata for the entire tenant. The [Power BI Management Module](#) is a set of PowerShell commands which make it easier to obtain metadata rather than dealing directly with the APIs. However, substantially more information is available directly from the APIs.

Long-term [usage and performance insights](#) are also available for workspaces backed by [Premium capacity](#). Administrators can analyze dataset activity, performance, and behavior. This capability is integrated with [Azure Log Analytics](#).

Auditing

Auditing data is valuable for informing and tracking your adoption goals, helping the COE be more effective, ideas for helpful documentation or new training, as well as for governance-related reporting.

The following table presents are a few ideas of what you can do with the information available from the Power BI activity log and APIs:

CATEGORY OF AUDITING DATA	TYPE OF QUESTIONS
Usage patterns and adoption	<p>What is the most often-used content, and by whom?</p> <p>How many users are active?</p> <p>Are report views trending up or down?</p> <p>Is there underutilized or abandoned content?</p> <p>Are viewers using browser or mobile apps more often?</p> <p>When is content published or updated, and by whom?</p>
Governance, security, and compliance	<p>When are workspace roles updated, and by whom?</p> <p>How many external users are accessing content?</p> <p>Who added or updated a sensitivity label?</p> <p>When does a tenant setting change, and by whom?</p> <p>What percentage of report views are based on certified datasets?</p> <p>What percentage of datasets support more than one report?</p> <p>How frequently is content downloaded, and by whom?</p> <p>Who generated an embed code for publish to web?</p>
Architecture reporting and analysis	<p>How many workspaces exist by type?</p> <p>How many reports exist by type?</p> <p>When is a gateway or data source created or updated?</p>

CATEGORY OF AUDITING DATA	TYPE OF QUESTIONS
User education and training opportunities	<p>Who started a Power BI trial?</p> <p>Who is doing too much sharing from their personal workspace?</p> <p>Who is publishing a lot of new datasets?</p> <p>Who is doing a lot of exporting?</p>

When considering needs for creating auditing reports, consider the following:

- What does success mean?
- What behaviors do you want to encourage?
- What do you want people to start doing?
- What do you want people to stop doing?

IMPORTANT

The raw data files that contain the auditing data should be stored in a very secure location, preferably one which is immutable (allowing no modifications or deletions). Immutable storage allows your auditors to rely on this data. A service like [Azure Data Lake Storage Gen2](#) is a flexible and low-cost alternative for this purpose.

Monitoring

[Microsoft Defender for Cloud Apps](#) is a cloud access security broker (CASB) that allows administrators to perform activities such as:

- Audit, monitor, and raise alerts based on activities.
- Create data loss prevention policies.
- Detect unusual behaviors and risky sessions.
- Limit activities performed by applications (in conjunction with [Azure Active Directory conditional access app control](#)).

Some very 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 whenever a tenant setting is updated in the Power BI service (for instance, an administrative activity is detected).
- Detect when suspicious or unusual behaviors have occurred, such as massive 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 assigned, such as exports from the Power BI service.
- Be notified when risky sessions are occurring, such as when the same user account connects from different geographical areas in a narrow time window.
- Determine when someone outside a predefined security group views specific content in the Power BI service.

Caution

Licensing, cost, and administrative permissions for Defender for Cloud Apps are all handled separately from Power BI. You can create an [application-specific admin](#) with permissions scoped to monitoring just the Power BI service.

Planning for change

Every month, new Power BI features and functionality are released. To be effective, it's crucial for those involved with system oversight to stay current.

The [Power BI blog](#) is the best place for customers to monitor announcements and new releases.

The [Power BI release plan](#) is where customers can find the public roadmap for future features and estimated dates. Sometimes a change that's coming is so important that it's useful to start planning for it well in advance. The planning cycle is in semesters: April-September, and October-March.

IMPORTANT

It's difficult to overestimate the importance of staying current. Being a few months behind on announcements can make it difficult to properly manage the Power BI service and support the user population efficiently.

Considerations and key actions

Considerations and key actions you can take to improve system oversight:

- Verify who is permitted to be a Power BI administrator. If possible, reduce the number of people granted this role if it's more than a few people.
- If you have people who *occasionally* need Power BI administrator rights, consider implementing [Privileged Identity Management](#) (PIM) in Azure Active Directory. It's designed to assign just-in-time role permissions that expire after a few hours.
- Check the status of cross-training and documentation in place for handling Power BI administration responsibilities.

Considerations and key actions you can take to improve management of the Power BI service:

- Conduct a review of all tenant settings to ensure they are aligned with [data culture](#) objectives and [governance](#) guidelines and policies. Verify which groups are assigned for each setting.
- Document the tenant settings for the internal Power BI community and post it in the centralized portal. Include which groups a user would need to request to be able to use a feature.
- 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.

Considerations and key actions you can take to improve management of user machines and devices:

- 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 (Power BI Pro or Premium Per User) can be handled together. It can simplify onboarding since new content creators won't always know what to ask for.
- Ensure an automated process is in place to install and update software, drivers, and settings to ensure all users have the same version.

Considerations and key actions you can take for data architecture planning:

- Assess what your end-to-end data architecture looks like. Make sure you're clear on:
 - How Power BI is currently used by the different business units in your organization versus how you want Power BI 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 Power BI users, and how they are documented and discovered.

- Review existing gateways that are used in 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.
- 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.

Considerations and key actions you can take to improve management of user licenses:

- Review the process to request a user license, including any prerequisites.
- Clarify whether self-service licensing purchasing is enabled.
- Confirm whether Power BI Pro and Premium Per User trials are enabled or disabled. Note that all user trials are Premium Per User, and they apply to Free licensed users signing up for a trial, as well as Power BI Pro users signing up for a Premium Per User trial.

Considerations and key actions you can take to improve cost management:

- Determine what the cost management objectives are and how to balance cost, features, usage patterns, and effective utilization of resources.
- Ensure you have access to the activity log data to assist with cost analysis.
- Schedule a routine process to evaluate costs, at least annually.

Considerations and key actions you can take to improve security and data protection:

- Clarify exactly what the expectations are for data protection, and ensure those expectations are communicated to the community of users.
- Understand and document the organizational policies around sharing Power BI content with external users, and ensure that Power BI service settings support those policies.
- If they do not currently exist, begin the process of deciding on sensitivity labels. Involve the relevant IT teams for decision-making and configuration in Microsoft 365.
- Investigate the use of Microsoft Defender for Cloud Apps to monitor user behavior and activities in the Power BI service.

Considerations and key actions you can take to begin or improve auditing and monitoring:

- Begin retrieving data from the Power BI activity log if you are not currently compiling the raw data. The easiest way to get started is to use the [Get-PowerBIActivityEvent](#) PowerShell cmdlet included with the Power BI Management Module. Retrieve and store the raw data without filtering or formatting, to ensure that all data elements are available for future analysis. A file system or data lake is an ideal location.
- Over time, determine what additional auditing data would be helpful to complement the activity log data.

Maturity levels

The following maturity levels will help you assess the current state of your Power BI system oversight:

LEVEL	STATE OF POWER BI SYSTEM OVERSIGHT
100: Initial	<p>Tenant settings are configured independently by one or more administrators based on their best judgment.</p> <p>Architecture needs, such as gateways and capacities, are satisfied on an as-needed basis, though without a strategic plan.</p> <p>Power BI activity logs are unused, or selectively used for tactical purposes.</p>

LEVEL	STATE OF POWER BI SYSTEM OVERSIGHT
200: Repeatable	<p>Tenant settings are reviewed on a regular basis, and purposefully align with established governance guidelines and policies.</p> <p>A small number of specific administrators are selected, and they have a good understanding of what users are trying to accomplish.</p> <p>An effective process exists for users to request licenses and software.</p> <p>Sensitivity labels are configured in Microsoft 365, though usage of labels remains inconsistent.</p>
300: Defined	<p>The tenant settings are fully documented in the community portal for users to reference, including how to request access to the correct groups.</p> <p>Cross-training and documentation exists for administrators to ensure continuity and stability.</p> <p>Sensitivity labels are assigned to content consistently.</p> <p>An automated process is in place to export Power BI activity log and API data to a secure location for reporting and auditing.</p>
400: Capable	<p>Automated policies are configured and actively monitored in Microsoft Defender for Cloud Apps for data loss prevention.</p> <p>Administrators work closely with the COE and governance teams to provide oversight of Power BI with an emphasis on user empowerment within the requisite guardrails.</p> <p>Decentralized management of data architecture (such as gateways or capacity management) is effectively handled to balance agility and control.</p> <p>Power BI activity log and API data is actively analyzed to monitor and audit Power BI activities. Proactive action is taken based on the data.</p>
500: Efficient	<p>Regular cost management analysis is done to ensure user needs are met in a cost-effective way.</p> <p>Power BI activity log and API data is actively used to inform and improve adoption efforts (in addition to monitoring and auditing of Power BI activities).</p>

Next steps

For more information about system oversight and Power BI administration, see the following resources:

- [Microsoft Learn – Administer Power BI – Part 1](#)
- [Microsoft Learn – 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](#)
- [Planning a Power BI enterprise deployment whitepaper](#)
- [Power BI adoption framework](#)

In the [next article](#) in the Power BI adoption roadmap series, in conclusion, learn about adoption-related resources that you might find valuable.

Power BI adoption roadmap conclusion

3/24/2022 • 3 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI adoption roadmap series of articles. For an overview of the series, see [Power BI adoption roadmap](#).

This article concludes the series on Power BI adoption. The strategic and tactical considerations and action items presented in this series will assist you in your Power BI adoption efforts, and with creating a productive data culture in your organization.

This series covered the following aspects of adoption:

- [Adoption overview](#)
- [Adoption maturity levels](#)
- [Data culture](#)
- [Executive sponsorship](#)
- [Content ownership and management](#)
- [Content delivery scope](#)
- [Center of Excellence](#)
- [Governance](#)
- [Mentoring and enablement](#)
- [Community of practice](#)
- [User support](#)
- [System oversight](#)

The rest of this article includes extra adoption-related resources that you might find valuable.

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 Power BI adoption roadmap series. The roadmap series focuses on the *why* and *what* of adopting Power BI, more so than the *how*.

Enterprise deployment whitepaper

The [Planning a Power BI enterprise deployment](#) whitepaper provides a comprehensive overview for Power BI implementers. Its primary goal is awareness of options, key considerations, decisions, and best practices. Because of the breadth of content, different sections of the whitepaper will appeal to managers, IT professionals, and self-service authors. The whitepaper is updated every 12-24 months.

The whitepaper goes deeper into the *what* and *how* of adopting Power BI, with a strong focus on technology. When you've finished reading the series of Power BI adoption articles, the whitepaper will fill you in with extra information to help put your plans into action.

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 may 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 series of modules to [use 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 Maturity \(DMM\) model](#) is a paid resource from ISACA. It's a comprehensive framework of data management practices in six key categories. It's designed to help organizations benchmark their capabilities, identify strengths and gaps, and leverage their data assets to improve business performance.

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.

These resources aren't required to take advantage of the guidance provided in this Power BI adoption series. They're reputable resources should you wish to continue your journey.

Partner community

Experienced Power BI partners are available to help your organization succeed with Power BI. To engage a Power

BI partner, visit the [Power BI partner portal](#).

Power BI implementation planning

3/24/2022 • 3 minutes to read • [Edit Online](#)

In this video, watch Matthew introduce you to the Power BI implementation planning series of articles.

Successfully implementing Power BI throughout the organization requires deliberate thought and planning. The Power BI implementation planning series of articles is intended to help you accomplish your Power BI implementation. The articles include key considerations, actions, decision-making criteria, recommendations, and describe implementation patterns for important common usage scenarios.

The series is a work in progress. The plan is to gradually release new and updated content over time. When completed, the series will:

- Complement the [Power BI adoption roadmap](#), which describes considerations for successful 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.
- Replace the [Planning a Power BI enterprise deployment](#) white paper, which was designed to describe a variety of technical factors when deploying Power BI. Relevant white paper content will be merged into this series, in a new format that's more discoverable and actionable.
- Replace the [Power BI adoption framework](#) (together with the [Power BI adoption roadmap](#)), which is a lightweight set of resources (videos and presentation slides) that were designed to help Microsoft partners deploy Power BI solutions for their customers. Relevant adoption framework action items will be merged into this series.

Recommendations

To set yourself up for success, it's recommended that you work through the following steps:

1. Read the complete [Power BI adoption roadmap](#), familiarizing yourself with each roadmap subject area. Assess your current state of Power BI adoption, and gain clarity on the data culture objectives for your organization.
2. Explore Power BI implementation planning articles that are relevant to you. It's recommended that you start with the [Power BI usage scenarios](#), which convey how Power BI can be used in diverse ways. Be sure to understand which usage scenarios are used in your organization, and by whom. Also, consider how these usage scenarios may influence the implementation strategies you decide on.
3. When necessary, refer to additional [Power BI documentation](#) for details on specific topics.

Target audience

The intended audience of this series of articles may 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 of articles is certain to be helpful for organizations that are in the early stages of a Power BI implementation or are planning an expanded implementation. It may 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, it's recommended that you become familiar with the [Power BI adoption roadmap](#) first.

Acknowledgments

This series of articles was written by Melissa Coates, Data Platform MVP and owner of [Coates Data Strategies](#) with significant contributions from Peter Myers, Matthew Roche, Alex Powers, and Chris Webb.

Next steps

In the [next article in this series](#), learn about usage scenarios that describe how Power BI can be implemented in many different ways.

Other helpful resources include:

- [Power BI adoption roadmap](#)
- [Power BI migration overview](#)

Experienced Power BI partners are available to help your organization succeed with the migration process. To engage a Power BI partner, visit the [Power BI partner portal](#).

Power BI usage scenarios

3/24/2022 • 6 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. 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 may 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 Power BI free license.

NOTE

Additional information about [content ownership and management](#) and [content delivery scope](#) are described in the [Power BI adoption roadmap](#).

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 datasets. This scenario describes decoupling the report creation process from the dataset creation process. To encourage report authors to find and reuse an existing shared dataset, it should be endorsed and made discoverable in the datasets hub in the Power BI service.
- **Customizable managed self-service BI:** The focus is on the dataset creator customizing or extending an existing dataset 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 dataset.
- **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 datasets.
- **Prototyping and sharing:** Prototyping techniques are very useful for validating requirements for visuals and calculations by subject matter experts. Prototyping solutions may be temporary, short-lived solutions, or they may 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 [Power BI 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.
- **Advanced data model management (usage scenario article not currently available):** The focus is on using more sophisticated modeling techniques. This scenario describes managing a data model using Tabular Editor, which is an external tool. The data model is deployed to the Power BI service using the XMLA endpoint available with Power BI Premium.
- **Enterprise content publishing (usage scenario article not currently available):** The focus is on using more sophisticated and programmatic techniques to publish content through development, test, and production workspaces. In this scenario, Azure DevOps coordinates and executes Power BI REST API operations to deploy content.

Real-time scenarios

The articles for this scenario aren't currently available.

Real-time scenarios describe different techniques to allow presenting data updates in near real-time. Monitoring data in real-time allows the organization to react faster when time-sensitive decisions must be made.

Embedding and hybrid scenarios

There are two *embedding and hybrid* scenarios: enterprise 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.

- **Enterprise embedding (usage scenario article not currently available):** 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.
- **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.

Next steps

In the [next article in this series](#), learn about enabling private analytics for an individual with the personal BI usage scenario.

Power BI usage scenarios: Personal BI

3/24/2022 • 4 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

As described in the [Power BI 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

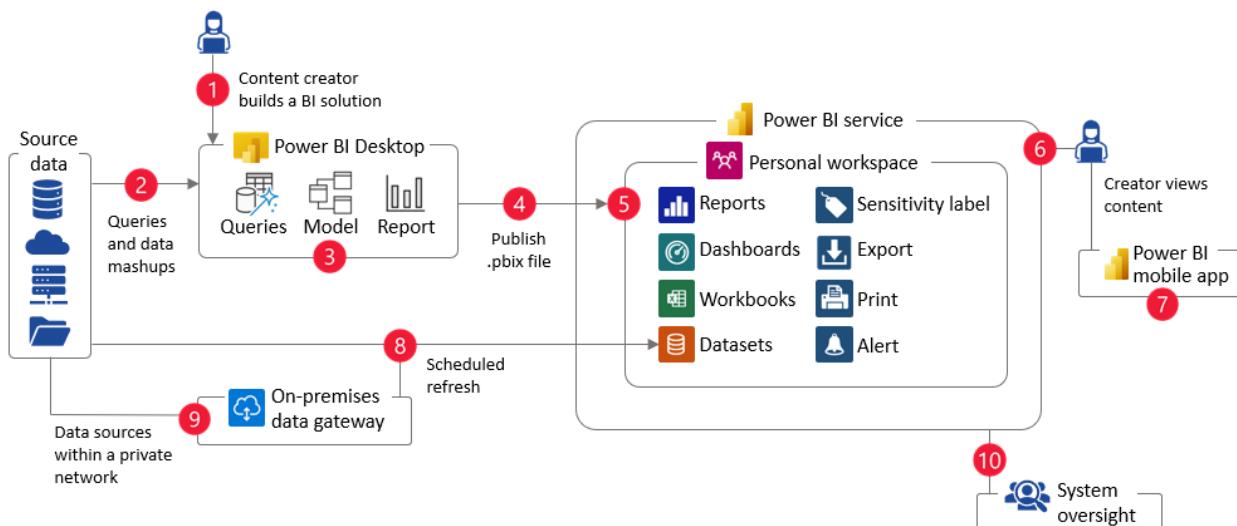
There are four *content collaboration and delivery* scenarios that build upon each other. The personal BI scenario is the first of the four scenarios. A list of all scenarios can be found in the [Power BI usage scenarios overview](#) article.

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



The scenario diagram depicts the following user actions, tools, and features:

ITEM	DESCRIPTION
1	The Power BI content creator develops a BI solution using Power BI Desktop .

ITEM	DESCRIPTION
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) 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.
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	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
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. Since only one person can manage and edit the content stored in a personal workspace, it represents risk to the organization. For example, when a person changes role or leaves the organization, it can be disruptive and challenging to move that content so that their successor can continue to work with it.

Use of Power BI 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 Power BI 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 Power BI free license can view content when it's hosted in a Premium capacity.

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](#) dataset (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 may 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.

Next steps

In the [next article in this series](#), learn about small team collaboration with the team BI usage scenario.

Power BI usage scenarios: Team BI

3/24/2022 • 7 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. 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 [Power BI 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.

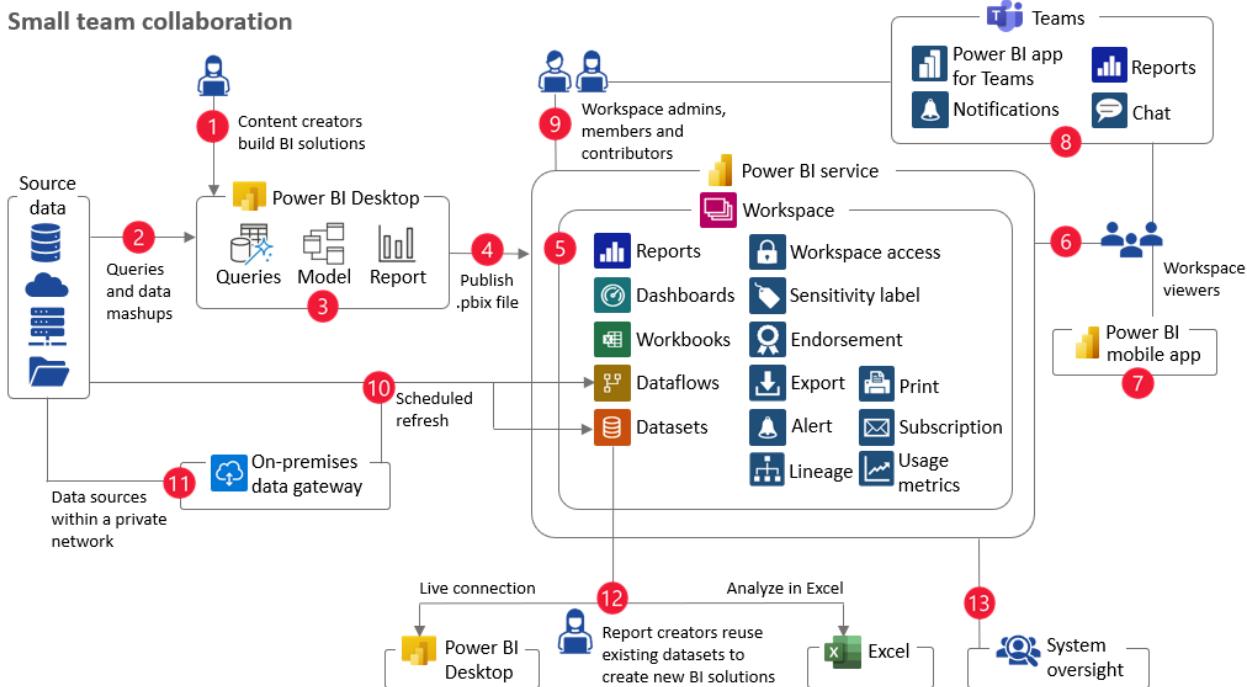
The [managed self-service BI](#) scenario introduces an important concept about decoupling dataset and report development. For simplicity, this concept isn't explicitly discussed in this article. You're encouraged to apply the concepts discussed in the managed self-service BI scenario whenever possible.

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.

Team BI

Small team collaboration



The scenario diagram depicts the following user actions, tools, and features:

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) 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.
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.

ITEM	DESCRIPTION
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 datasets or dataflows—up to date.
11	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
12	Other self-service content creators can author new reports using an existing dataset. They can choose to use Power BI Desktop, Excel, or Power BI Report Builder (not depicted in the scenario diagram). The reuse of existing datasets in this manner is highly encouraged.
13	Power BI administrators oversee and monitor activity in the Power BI service. Team BI solutions may 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 Business 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 dataset 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 dataset 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 artifacts, such as datasets 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 artifacts 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 artifacts 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, Power BI 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 datasets

The reuse of existing datasets is important for team collaboration. It helps to promote a *single version of the truth*. It's particularly important when a small number of dataset creators support many report creators. A Power BI Desktop [live connection](#) can connect a report to an existing dataset, avoiding the need to create another dataset. 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 datasets.
- Reduces the risk of inconsistent data and calculations.
- Supports all slicing, dicing, and pivoting capabilities within the visuals while remaining connected to the dataset that's stored in the Power BI service.

To access an existing dataset, the content creator must have [Build permission](#) for the dataset. 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 an artifact. The [managed self-service BI](#) scenario explores the reuse of shared datasets 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](#) dataset (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.

Next steps

In the [next article in this series](#), learn about distributing content to a larger number of viewers in the departmental BI usage scenario.

Power BI usage scenarios: Departmental BI

3/24/2022 • 7 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

As described in the [Power BI 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.

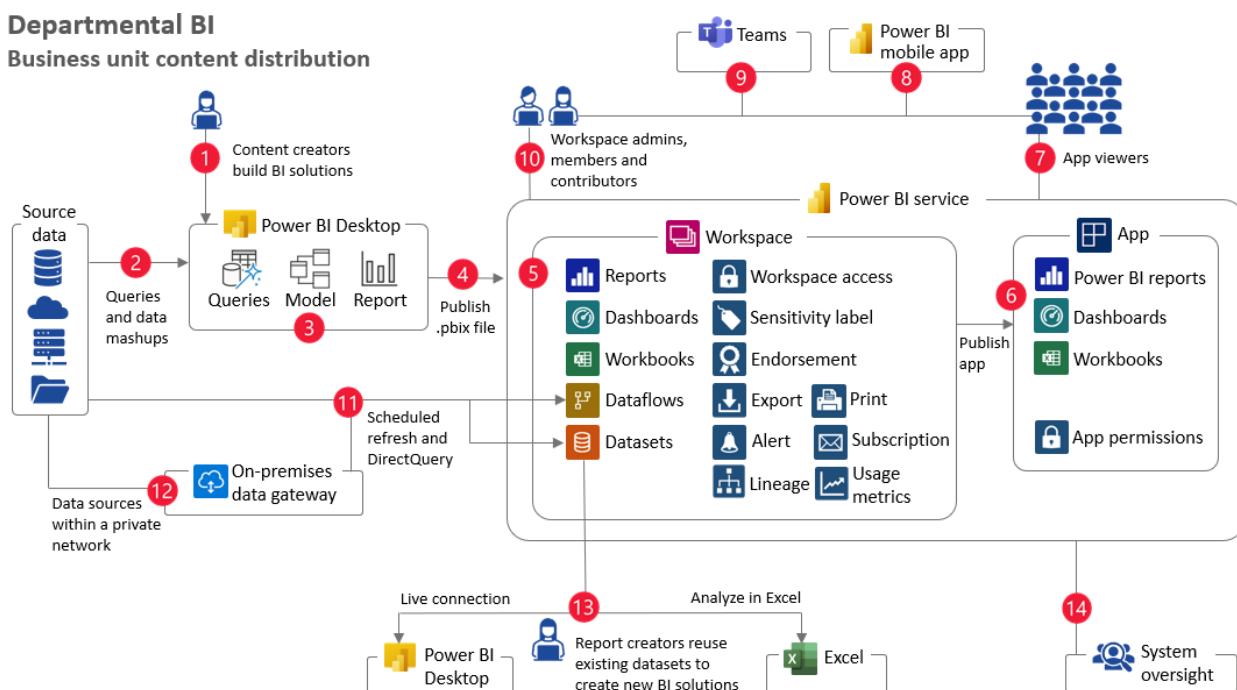
The [managed self-service BI](#) scenario introduces an important concept about decoupling dataset and report development. For simplicity, this concept isn't explicitly discussed in this article. You're encouraged to apply the concepts discussed in the managed self-service BI scenario whenever possible.

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.

Departmental BI

Business unit content distribution



The scenario diagram depicts the following user actions, tools, and features:

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) 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.
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 assigned read-only permissions. App permissions are managed separately from the workspace.
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 datasets or dataflows—up to date.
12	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
13	Other self-service content creators can author new reports using an existing dataset. They can choose to use Power BI Desktop, Excel, or Power BI Report Builder (not depicted in the scenario diagram). The reuse of existing datasets in this manner is highly encouraged.

ITEM	DESCRIPTION
14	Power BI administrators oversee and monitor activity in the Power BI service. Departmental BI solutions may 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 Business 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 dataset 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 dataset 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 artifacts, such as datasets 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, Power BI free license users (with proper permissions) can view the workspace (and/or app) content. This approach is described in the [enterprise BI](#) scenario.

Reuse existing datasets

The reuse of existing datasets is important for team collaboration. It helps to promote a *single version of the truth*. It's particularly important when a small number of dataset creators support many report creators. A Power BI Desktop [live connection](#) can connect a report to an existing dataset, avoiding the need to create another dataset. Alternatively, when users prefer to create an Excel report, they can use the [Analyze in Excel](#) feature. Retaining connectivity to the dataset is preferred to exporting data to Excel because it:

- Avoids creating duplicate datasets.
- Reduces the risk of inconsistent data and calculations.
- Supports all slicing, dicing, and pivoting capabilities within the visuals while remaining connected to the dataset that's stored in the Power BI service.

To access an existing dataset, the content creator must have [Build permission](#) for the dataset. 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 an artifact. The [managed self-service BI](#) scenario explores the reuse of shared datasets 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](#) dataset (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.

Next steps

In the [next article in this series](#), learn about organization-wide content distribution at scale in the enterprise BI scenario.

Power BI usage scenarios: Enterprise BI

3/24/2022 • 9 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

As described in the [Power BI 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 [Power BI Premium capacity](#), which allows content to be widely [distributed to consumers](#) who have a Power BI free license.

Consumers can include users within the organization, as well as [guest users](#) who are external to the organization.

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.

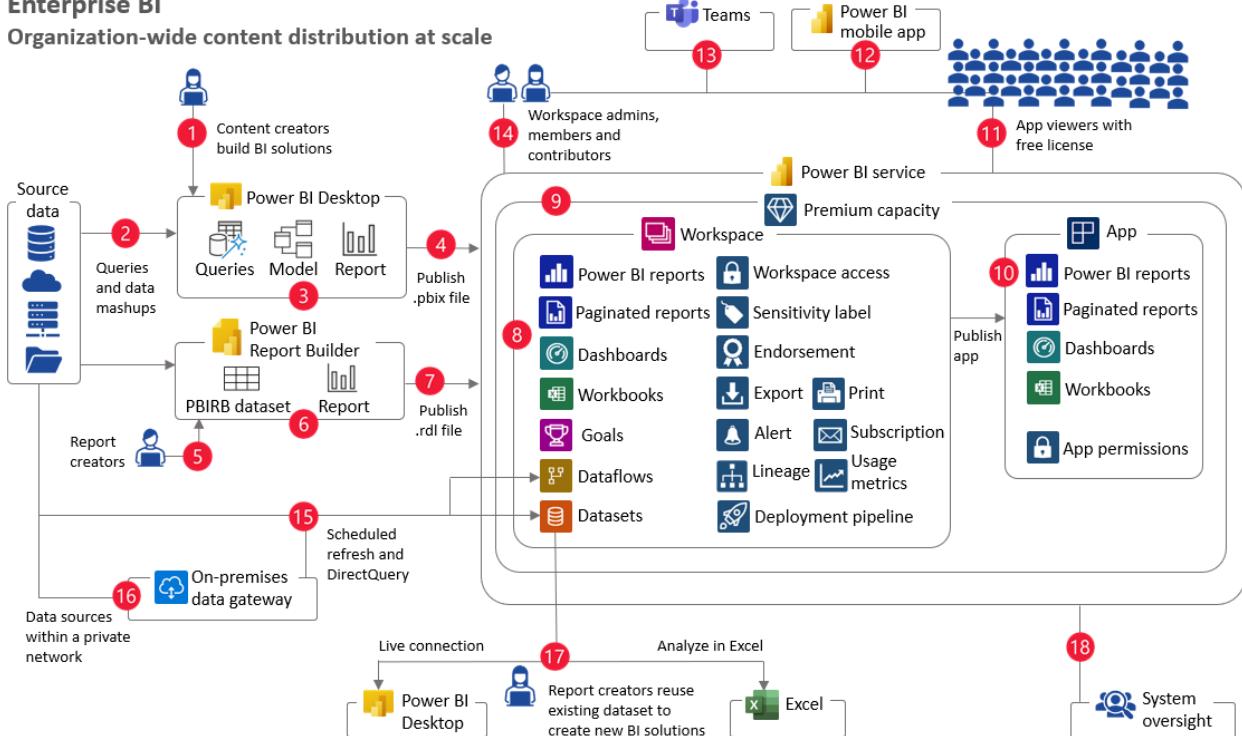
The [managed self-service BI](#) scenario introduces an important concept about decoupling dataset and report development. For simplicity, this concept isn't explicitly discussed in this article. You're encouraged to apply the concepts discussed in the managed self-service BI scenario whenever possible.

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](#).

Enterprise BI

Organization-wide content distribution at scale



The scenario diagram depicts the following user actions, tools, and features:

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 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) to the Power BI service.
5	Report creators develop paginated reports using Power BI Report Builder.
6	Power BI Report Builder queries data from one or more data source types. A paginated report is produced to meet requirements for a highly formatted, print-ready report.
7	When ready, report creators publish their Power BI Report Builder file (.rdl) to the Power BI service.
8	Multiple artifact types can be published to a Premium workspace . This workspace license mode supports paginated reports.

ITEM	DESCRIPTION
9	In the enterprise BI scenario, use of 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 Power BI 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 assigned read-only permissions. App permissions are managed separately from the workspace. In an enterprise BI scenario, users with any type of Power BI license (free, Power BI Pro, or PPU) can be assigned as a viewer of the app. This feature applies only when the workspace is assigned a license mode of Premium per 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 datasets or dataflows—up to date.
16	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
17	Other self-service content creators can author new reports using an existing dataset. They can choose to use Power BI Desktop, Excel, or Power BI Report Builder. The reuse of existing datasets in this manner is highly encouraged. The managed self-service BI scenario explores dataset reuse further.
18	Power BI administrators oversee and monitor activity in the Power BI service. 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

When the workspace license mode is **Premium** (via either Power BI capacity or Premium Per User), paginated reports can be published to the workspace. [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 Business 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 dataset, 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 dataset 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 artifacts, such as datasets 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 Power BI free license users

Users with a Power BI 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) [capacity-based license](#) 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 datasets

The reuse of existing datasets is important for team collaboration. It helps to promote a *single version of the truth*. It's particularly important when a small number of dataset creators support many report creators. A Power BI Desktop [live connection](#) can connect a report to an existing dataset, avoiding the need to create another dataset. Alternatively, when users prefer to create an Excel report, they can use the [Analyze in Excel](#) feature. Retaining connectivity to the dataset is preferred to exporting data to Excel because it:

- Avoids creating duplicate datasets.
- Reduces the risk of inconsistent data and calculations.
- Supports all slicing, dicing, and pivoting capabilities within the visuals while remaining connected to the dataset that's stored in the Power BI service.

To access an existing dataset, the content creator must have [Build permission](#) for the dataset. 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 an artifact. The [managed self-service BI](#) scenario explores the reuse of shared datasets 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](#) dataset (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.

Next steps

In the [next article in this series](#), learn more about the importance of reusing datasets in the managed self-service BI scenario.

Power BI usage scenarios: Managed self-service BI

3/24/2022 • 7 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

As described in the [Power BI 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 dataset 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 dataset.

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.

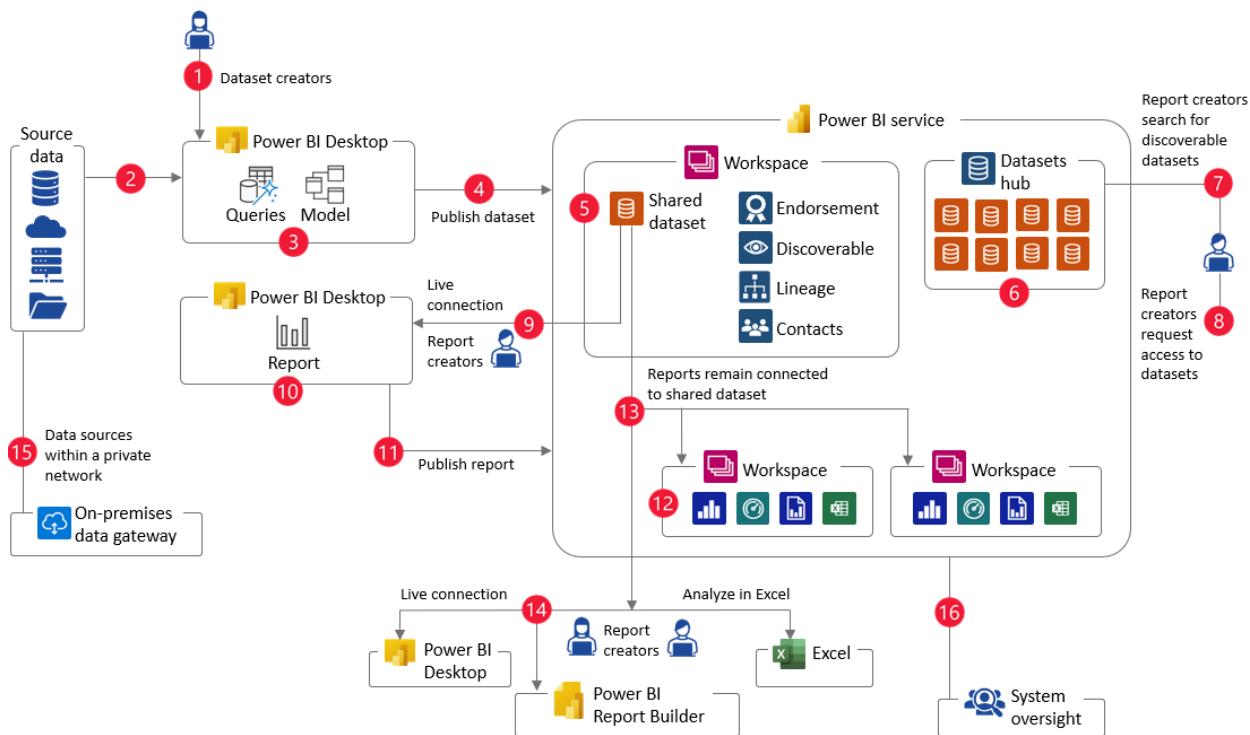
For brevity, some aspects described in the [content collaboration and delivery scenarios](#) topic aren't covered in this article. For complete coverage, read those articles first.

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 datasets. To accomplish that, this scenario focuses on decoupling the model development process from the report creation process.

Managed self-service BI

Reuse of a centralized shared dataset by other report creators



The scenario diagram depicts the following user actions, tools, and features:

ITEM	DESCRIPTION
1	Dataset creators develop models using Power BI Desktop. For datasets 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.
4	When ready, dataset creators publish their Power BI Desktop file (.pbix) <i>that contains only a model</i> to the Power BI service.
5	The dataset is published to a workspace dedicated to storing and securing shared datasets. Since the dataset is intended for reuse, it's endorsed (certified or promoted, as appropriate). The dataset 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 artifacts, including reports connected to the dataset.
6	Dataset discovery in the datasets hub is enabled because the dataset is marked as discoverable. Discoverability allows the existence of a dataset to be visible in the datasets hub by other Power BI content creators who are looking for data.

ITEM	DESCRIPTION
7	Report creators use the datasets hub in the Power BI service to search for discoverable datasets.
8	If report creators don't have permission, they can request Build permission on the dataset. This starts a workflow to request Build permission from an authorized approver.
9	Report creators create new reports using Power BI Desktop. Reports use a live connection to a shared dataset.
10	Report creators develop reports in Power BI Desktop.
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 datasets that are stored in a different workspace. Any changes to the shared dataset affect all reports connected to it.
14	Other self-service report creators can author new reports using the existing shared dataset. Report creators can choose to use Power BI Desktop, Power BI Report Builder, or Excel.
15	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
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 dataset

The key aspect of making *managed self-service BI*/work is to minimize the number of datasets. This scenario is about [shared datasets](#) that help achieve a *single version of the truth*.

NOTE

For simplicity, the scenario diagram depicts just one shared dataset. However, it's not usually practical to model all organizational data in a single dataset. The other extreme is to create a new dataset 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 datasets and creating new datasets when it makes sense to do so.

Decouple dataset and reports

When the dataset is decoupled from reports, it facilitates the separation of effort and responsibility. A shared dataset 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.

Dataset endorsement

Because shared datasets are intended for reuse, it's helpful to [endorse](#) them. A *certified* dataset conveys to report creators that the data is trustworthy and meets the organization's quality standards. A *promoted* dataset highlights that the dataset 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.

Dataset discovery

The [datasets hub](#) helps report creators find, explore, and use datasets across the organization. In addition to dataset endorsement, [enabling dataset discovery](#) is critical for promoting its reuse. A discoverable dataset is visible in the datasets hub for report creators who are searching for data.

NOTE

If a dataset isn't configured to be discoverable, only Power BI users with Build permission can find it.

Request dataset access

A report creator may find a dataset in the [datasets hub](#) that they want to use. If they don't have Build permission for the dataset, they can request access. Depending on the [request access setting](#) for the dataset, an email will be submitted to the dataset owner or custom instructions will be presented to the person who is requesting access.

Live connection to the shared dataset

A Power BI Desktop [live connection](#) connects a report to an existing dataset. 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 dataset. However, the [customizable managed self-service BI](#) scenario describes how a dataset 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 dataset 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 datasets. 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 dataset, RLS still applies. For more information, see [Using RLS with workspaces in Power BI](#).

Dependency and impact analysis

When a shared dataset 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 dataset change, first perform [impact analysis](#) to understand which dependent reports may 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](#) dataset.

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 datasets. A high report-to-dataset ratio indicates good reuse of datasets.

Next steps

In the [next article in this series](#), learn about ways to customize and extend a shared dataset to meet additional types of requirements.

Power BI usage scenarios: Customizable managed self-service BI

3/24/2022 • 8 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

As described in the [Power BI 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, dataset creators can extend, personalize, or customize existing shared datasets. New specialized datasets can be created that meet business requirements not met by existing centrally delivered datasets. 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 dataset (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.

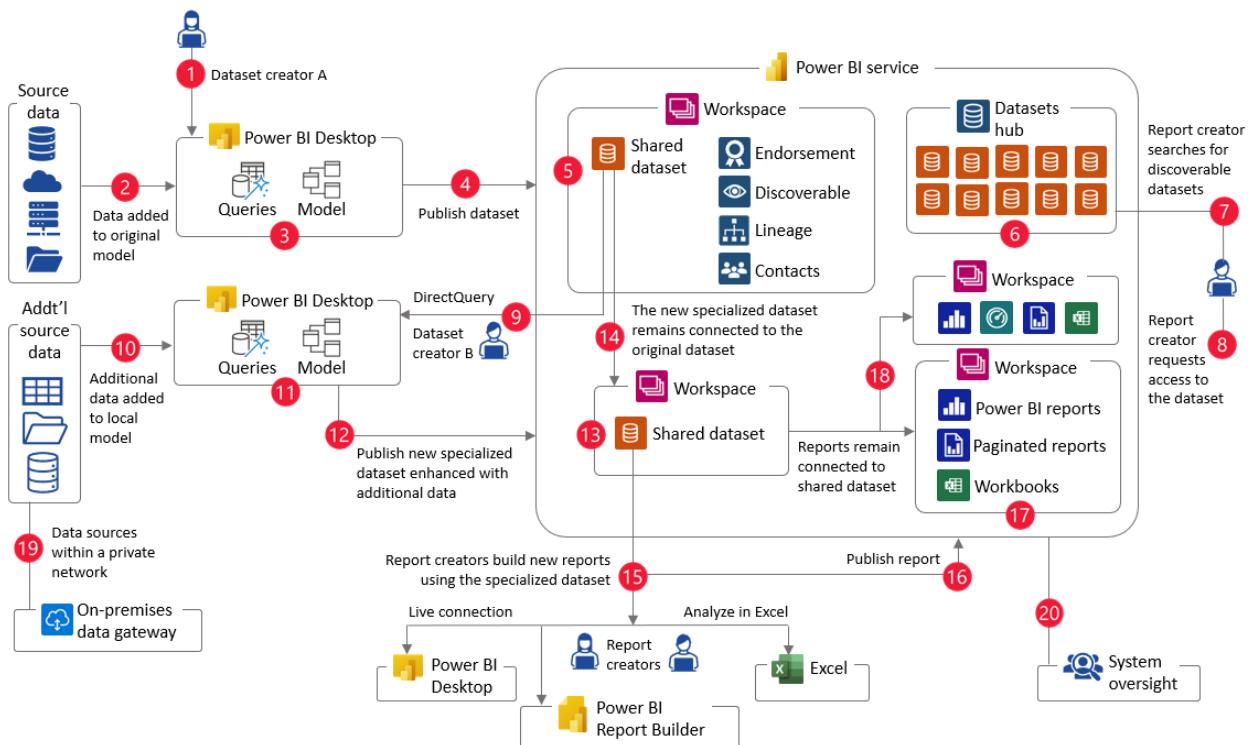
For brevity, some aspects described in the [content collaboration and delivery scenarios](#) topic aren't covered in this article. For complete coverage, read those articles first.

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 dataset. The goal is to achieve reusability whenever possible and to allow flexibility to meet additional analytical requirements.

Customizable managed self-service BI

Creating a new specialized dataset by extending and personalizing an existing dataset



The scenario diagram depicts the following user actions, tools, and features:

ITEM	DESCRIPTION
1	Dataset creator A develops a model using Power BI Desktop. For a dataset 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 because it may be used as a data source by many self-service report creators.
4	When ready, dataset creator A publishes their Power BI Desktop file (.pbix) <i>that contains only a model</i> to the Power BI service.
5	The dataset is published to a workspace dedicated to storing and securing shared datasets. Since the dataset is intended for reuse, it's endorsed (certified or promoted, as appropriate). The dataset 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 artifacts.
6	Dataset discovery in the datasets hub is enabled because the dataset is marked as discoverable. Discoverability allows the existence of a dataset to be visible in the datasets hub by other Power BI content creators who are looking for data.

ITEM	DESCRIPTION
7	Dataset creator B uses the datasets hub in the Power BI service to search for discoverable datasets.
8	If dataset creator B doesn't have permission, they can request Build permission on the dataset. This starts a workflow to request Build permission from an authorized approver.
9	In Power BI Desktop, dataset creator B creates a live connection to the original shared dataset that's located in the Power BI service. Since the intention is to extend and customize the original dataset, 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 dataset so that additional analytical requirements are met by the new specialized dataset.
11	Relationships are created in Power BI Desktop between the existing tables (from the shared dataset, 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 model.
12	When ready, dataset creator B publishes their Power BI Desktop file to the Power BI service.
13	The new specialized dataset is published to a workspace dedicated to storing and securing datasets that are owned and managed by the department.
14	The specialized dataset remains connected to the original Power BI shared dataset. Any changes to the original shared dataset will affect downstream specialized datasets that have dependency on it .
15	Other self-service report creators can author new reports connected to the specialized dataset. Report creators can choose to use Power BI Desktop, Power BI Report Builder, or Excel.
16	Report creators create new reports using Power BI Desktop.
17	Reports are published to a workspace dedicated to storing and securing reports and dashboards.
18	Published reports remain connected to the specialized dataset that's stored in a different workspace. Any changes to the specialized dataset affect all reports connected to it.

ITEM	DESCRIPTION
19	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
20	Power BI administrators oversee and monitor activity in the Power BI service.

Key points

The following are some key points to emphasize about the customizable managed self-service BI scenario.

Shared dataset

The key aspect of making *managed self-service BI*/work is to minimize the number of datasets. This scenario depicts a [shared dataset](#) that contributes towards achieving a *single version of the truth*.

NOTE

For simplicity, the scenario diagram depicts just one shared dataset. However, it's not usually practical to model all organizational data in a single dataset. The other extreme is to create a new dataset for every report, as less experienced content creators often do. The goal is to find the right balance, leaning toward relatively few datasets and creating new datasets when it makes sense to do so.

Augment the initial shared dataset

Sometimes self-service creators need to augment an existing dataset with, for instance, additional data that's specific to their department. In this case, they can use [DirectQuery connections to Power BI datasets](#). 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 dataset (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 dataset. 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.

Dataset endorsement

Because shared datasets are intended for reuse, it's helpful to [endorse](#) them. A *certified* dataset conveys to report creators that the data is trustworthy and meets the organization's quality standards. A *promoted* dataset highlights that the dataset 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.

Dataset discovery

The [datasets hub](#) helps report creators find, explore, and use datasets across the organization. In addition to dataset endorsement, [enabling dataset discovery](#) is critical for promoting its reuse. A discoverable dataset is visible in the datasets hub for report creators who are searching for data.

NOTE

If a dataset isn't configured to be discoverable, only Power BI users with Build permission can find it.

Request dataset access

A report creator may find a dataset in the [datasets hub](#) that they want to use. If they don't have Build permission for the dataset, they can request access. Depending on the [request access setting](#) for the dataset, an email will be submitted to the dataset 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 dataset 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 datasets. 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 dataset is used by other datasets or reports, those dependent objects can exist in many workspaces. The [lineage view](#) helps identify and understand the downstream dependencies. When planning a dataset change, first perform [impact analysis](#) to understand which datasets 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](#) dataset.

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 dataset as well as dependent datasets.

Next steps

In the [next article in this series](#), learn about reusing data preparation work with dataflows in the self-service data preparation scenario.

Power BI usage scenarios: Self-service data preparation

3/24/2022 • 10 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. 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. Streaming dataflows, which enable near real-time reporting, aren't covered in this scenario.

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.

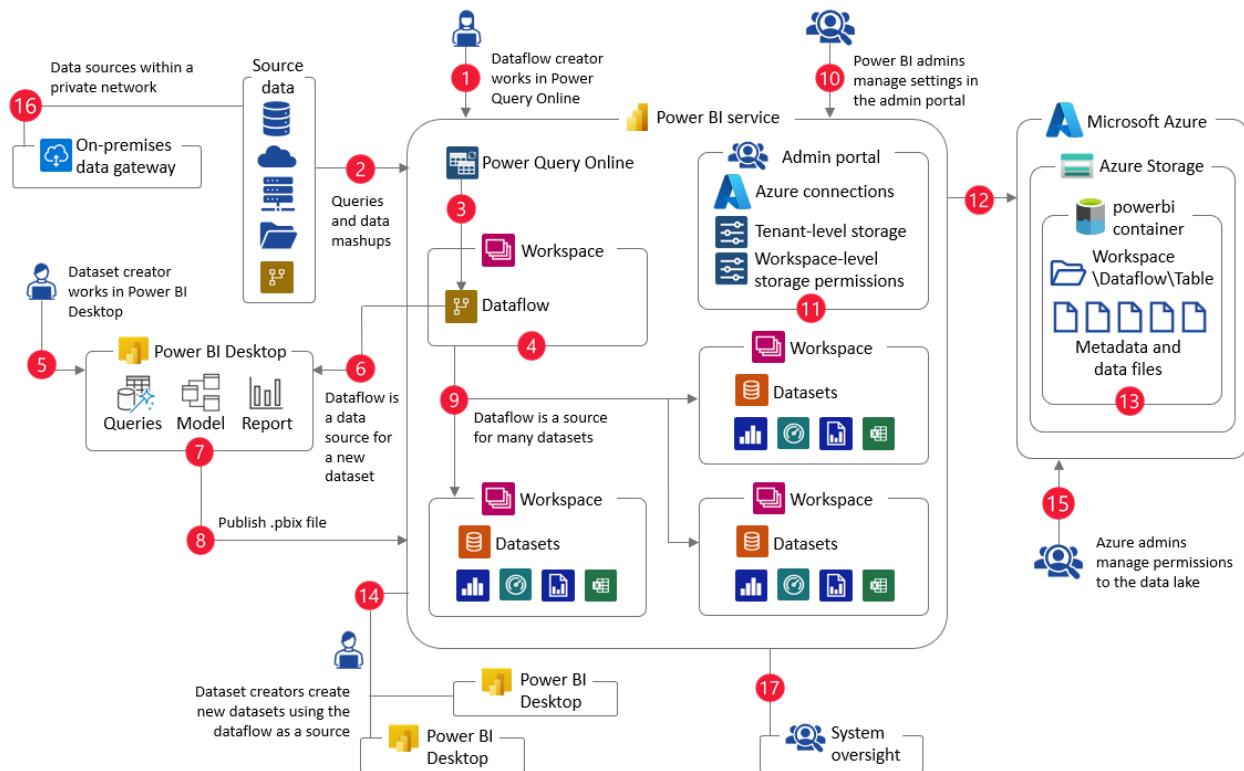
For brevity, some aspects described in the [content collaboration and delivery scenarios](#) topic aren't covered in this article. For complete coverage, read those articles first.

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 datasets. The goal is for many datasets to leverage the data preparation that's done once by the dataflow.

Self-service data preparation

Dataflows for centralizing data cleansing and transformation work



The scenario diagram depicts the following user actions, tools, and features:

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.
3	Dataflows are developed using Power Query Online, which is a web-based version of Power Query . The familiar Power Query interface in Power Query Online makes the transition from Power BI Desktop easy.
4	The dataflow is saved as an artifact 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).
5	The dataset creator develops a new data model using Power BI Desktop.
6	The dataflow is a data source for the new data model.
7	The dataset creator can use the full capabilities of Power Query within Power BI Desktop. They can optionally apply additional query steps to further transform the dataflow data or merge the dataflow output.

ITEM	DESCRIPTION
8	When ready, the dataset creator publishes the Power BI Desktop file (.pbix) that contains the data model to the Power BI service. Refresh for the dataset is managed separately from the dataflow (not depicted in the scenario diagram).
9	The dataflow can be reused as a data source by other datasets that could reside in different workspaces.
10	Power BI administrators manage settings in the Admin portal.
11	In the Admin portal, Power BI administrators can configure 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.
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 <i>bring your own data lake</i> . 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 <i>filesystem</i> . 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	Other self-service dataset creators can create new data models in Power BI Desktop using the dataflow as a data source.
15	Azure administrators manage permissions for the organization's ADLS Gen2 account.
16	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
17	Power BI administrators oversee and monitor activity in the Power BI service.

Key points

The following are some key points to emphasize about the self-service data preparation scenario.

Power BI dataflows

A [Power BI dataflow](#) comprises a collection of tables (previously called *entities*). All work to create a dataflow is done in [Power Query Online](#) and is saved to a [workspace](#).

NOTE

Dataflows cannot be created in personal workspaces.

Support dataset creators

The scenario diagram depicts using a Power BI dataflow to provide prepared data to other self-service dataset creators.

NOTE

Datasets use the dataflow as a data source. A report cannot connect directly to a dataflow.

Here are some advantages of using Power BI dataflows:

- Dataset 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 may 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 dataset creators. Reuse is particularly helpful for commonly used tables—especially [dimension tables](#), like date, customer, and product.
- The level of effort required by dataset creators is reduced because the data preparation work has been *decoupled* from the data modeling work.
- Fewer dataset creators need direct access to source systems. Source systems can be complex to query and may require specialized access permissions.
- The number of refreshes executed on source systems is reduced because dataset 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 datasets.
- Decoupling data preparation logic into dataflows can help improve dataset refresh success. If a dataflow refresh fails, datasets 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 datasets. Also, refine the dataflow output to apply friendly names and use specific data types. These techniques promote consistency in dependent datasets and helps reduce the amount of work that dataset creators need to do.

Dataset creator flexibility

When a dataset 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

Dataflows with Power BI Premium (via [capacity licensing](#) or [Premium Per User licensing](#)) can use [advanced features](#). These features are extremely helpful for managing data in the enterprise (not depicted in the scenario diagram). There are many techniques, patterns, and [best practices](#) for dataflow design that can take dataflows from self-service to enterprise-ready.

NOTE

One of the [advanced features](#) is incremental refresh for dataflows. Although incremental refresh for *datasets* is a Power BI Pro feature, incremental refresh for dataflows is a Premium feature.

Dataflow and dataset refresh

As previously mentioned, a dataflow is a source of data for datasets. In most cases, multiple data refresh schedules are involved: one for the dataflow and one for each dataset. Alternatively, it's possible to use [DirectQuery from the dataset to the dataflow](#), which is a Premium feature (not depicted in the scenario diagram).

NOTE

There's a different type of dataflow called a [streaming dataflow](#) (not depicted in the scenario diagram). A streaming dataflow is one technique to achieve near real-time analytics in Power BI.

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 may *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 configured, a [workspace is set 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 dataset 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 datasets 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

Dataflows cannot be created in personal workspaces.

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](#) dataset (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. With a self-service data preparation scenario, it's particularly helpful to track usage of dataflows.

Next steps

For other useful scenarios to help you with Power BI implementation decisions, see the [Power BI usage scenarios](#) article.

Power BI usage scenarios: Prototyping and sharing

3/24/2022 • 5 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

As described in the [Power BI 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 may be shared with others to get feedback during development iterations. The solution may be a temporary, short-lived solution, or it may 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 may 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.

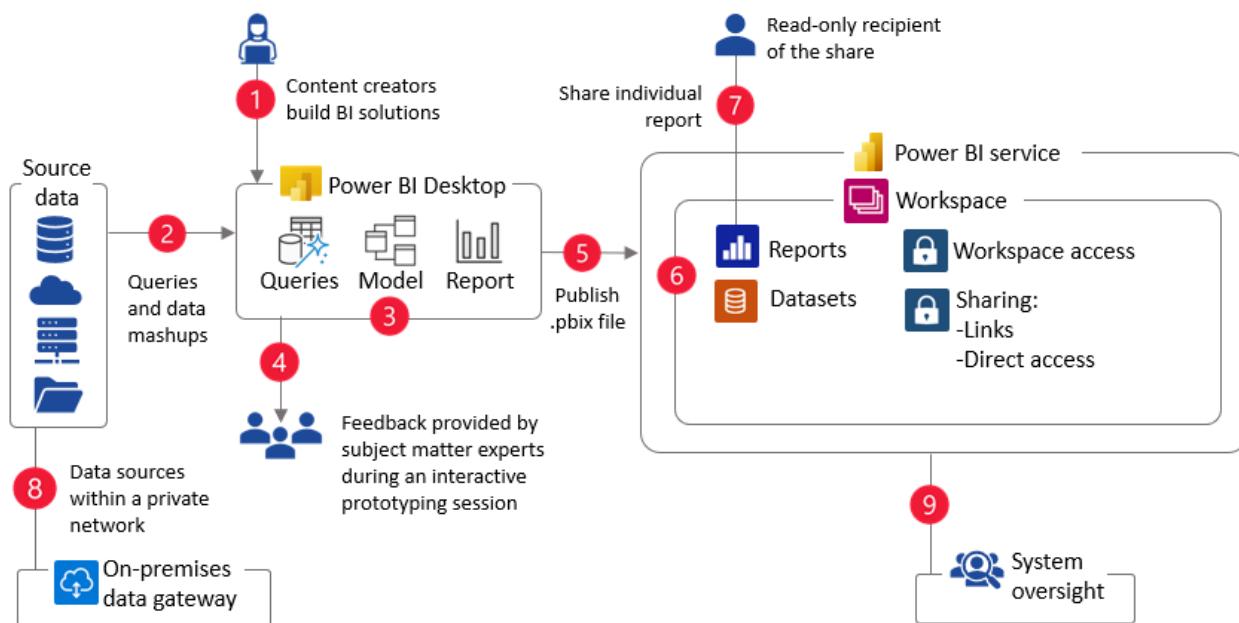
For brevity, some aspects described in the [content collaboration and delivery scenarios](#) topic aren't covered in this article. For complete coverage, read those articles first.

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.

Prototyping and sharing

Rapid exploration of user requirements



The scenario diagram depicts the following user actions, tools, and features:

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.
5	If desired, content creators publish their Power BI Desktop file (.pbix) 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.

ITEM	DESCRIPTION
8	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required for data refresh.
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 may 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 artifact 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 artifacts* 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](#) dataset (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.

Next steps

For other useful scenarios to help you with Power BI implementation decisions, see the [Power BI usage scenarios](#) article.

Power BI usage scenarios: Self-service content publishing

3/24/2022 • 9 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. 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.

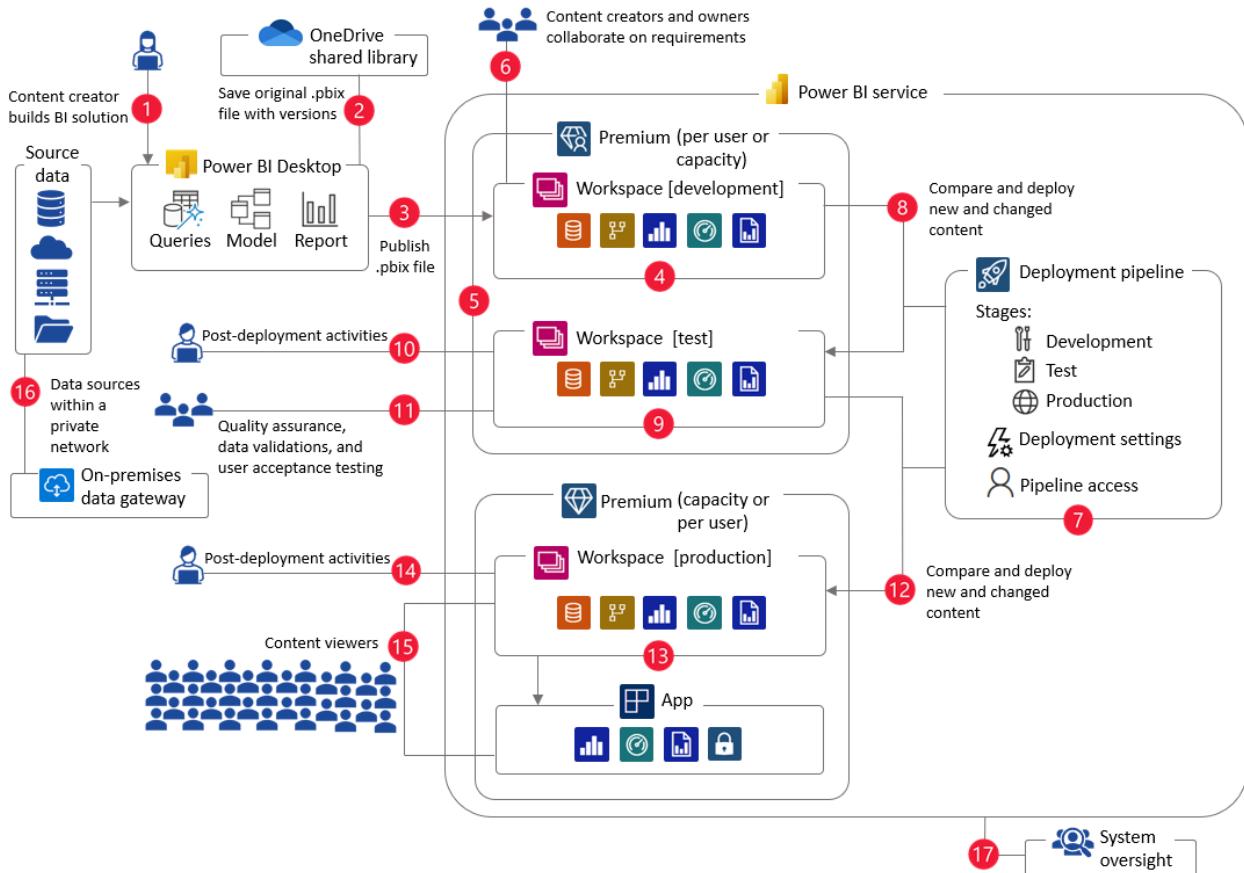
For brevity, some aspects described in the [content collaboration and delivery scenarios](#) topic aren't covered in this article. For complete coverage, read those articles first.

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

Publishing content to development, test, and production with deployment pipelines



The scenario diagram depicts the following user actions, tools, and features:

ITEM	DESCRIPTION
1	The Power BI content creator develops a BI solution using Power BI Desktop.
2	The Power BI Desktop file (.pbix) is saved to a shared library 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	The development (or test) workspace is set to Premium per user , Premium per capacity , or Embedded license mode .
6	Content creators and owners collaborate in the development workspace to ensure all requirements are met.
7	A deployment pipeline administrator configures 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 configured for the deployment pipeline.

ITEM	DESCRIPTION
8	When the development content is ready, the deployment pipeline compares the content between the development and test stages. Some, or all, artifacts are deployed to a workspace that's dedicated to testing .
9	The test (or development) workspace is set to Premium per user , Premium per capacity , or Embedded license mode .
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. Some, or all, artifacts are deployed to a workspace that's dedicated to production .
13	The production workspace is set to Premium per user , Premium per capacity , or Embedded license mode . For a production workspace, Premium per capacity license mode is often more appropriate when there's a large number of read-only consumers.
14	Once 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	To connect to data sources that reside within a private organizational network, an On-premises data gateway is required.
17	Power BI administrators oversee and monitor activity in the Power BI service. Content that's deemed critical enough to have separate development, test, and production workspaces may be subject to stricter governance requirements than less critical content.

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. Artifacts 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 artifacts 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 may 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

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) [capacity-based license](#) 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.
- **Artifact properties:** Certain artifact properties, such as sensitivity labels, may be overwritten during a deployment in [certain circumstances](#).
- **Unsupported artifacts:** Additional manual steps may need to be taken for [artifacts 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](#) dataset (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.

Next steps

For other useful scenarios to help you with Power BI implementation decisions, see the [Power BI usage scenarios](#) article.

Power BI usage scenarios: On-premises reporting

3/24/2022 • 4 minutes to read • [Edit Online](#)

NOTE

This article forms part of the Power BI planning implementation series of articles. For an introduction to the series, see [Power BI implementation planning](#).

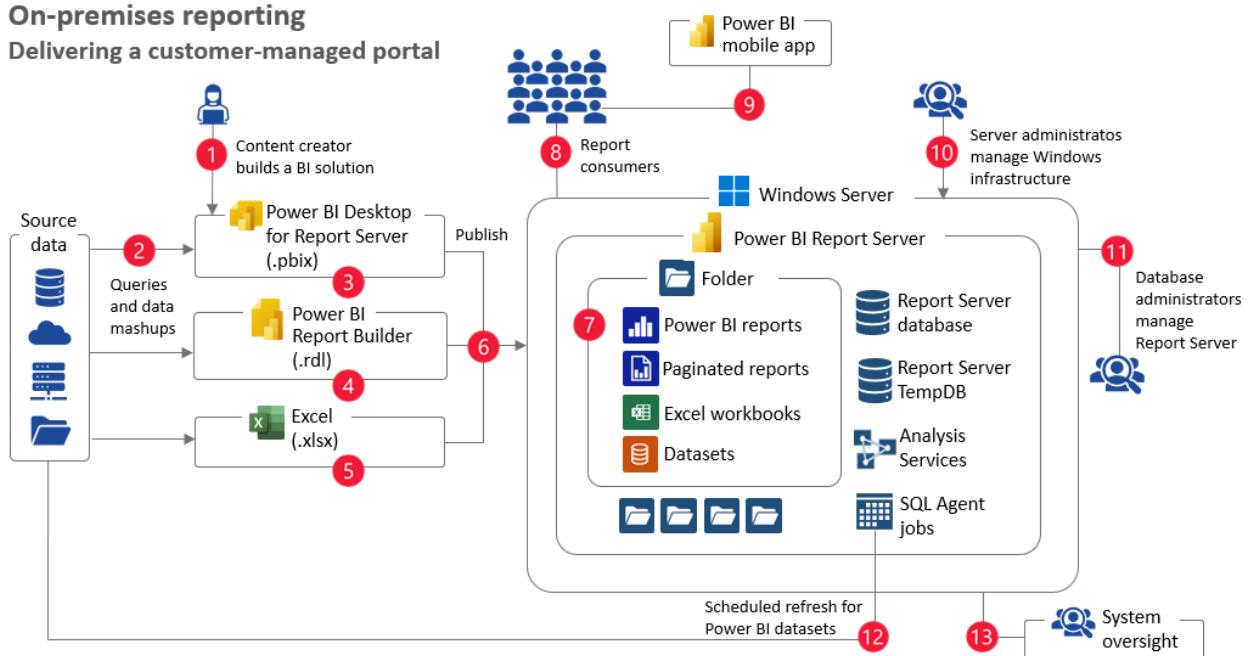
The *on-premises reporting* scenario is one of several *hybrid and custom scenarios* for deploying Power BI solutions without using the Power BI service.

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 may 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.

On-premises reporting Delivering a customer-managed portal



The scenario diagram depicts the following user actions, tools, and features:

ITEM	DESCRIPTION
1	A Power BI content creator builds a BI solution.

ITEM	DESCRIPTION
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 datasets .
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.

With the purchase of Power BI Premium capacity, Power BI Report Server may 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).

Next steps

For other useful scenarios to help you with Power BI implementation decisions, see the [Power BI usage scenarios](#) article.

Microsoft's BI transformation

3/24/2022 • 7 minutes to read • [Edit Online](#)

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 [Power BI 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 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.

Next steps

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

- [Establish a Center of Excellence](#)
- [Power BI adoption roadmap: Center of Excellence](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

In the [next article in this series](#), learn how a COE helped us at Microsoft 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 engage a Power BI partner, visit the [Power BI partner portal](#).

You can also engage with experienced consulting partners. They can help you [assess](#), [evaluate](#), or [implement](#) Power BI.

How Microsoft established a Center of Excellence

3/24/2022 • 7 minutes to read • [Edit Online](#)

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 [Power BI 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 datasets 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 datasets 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.

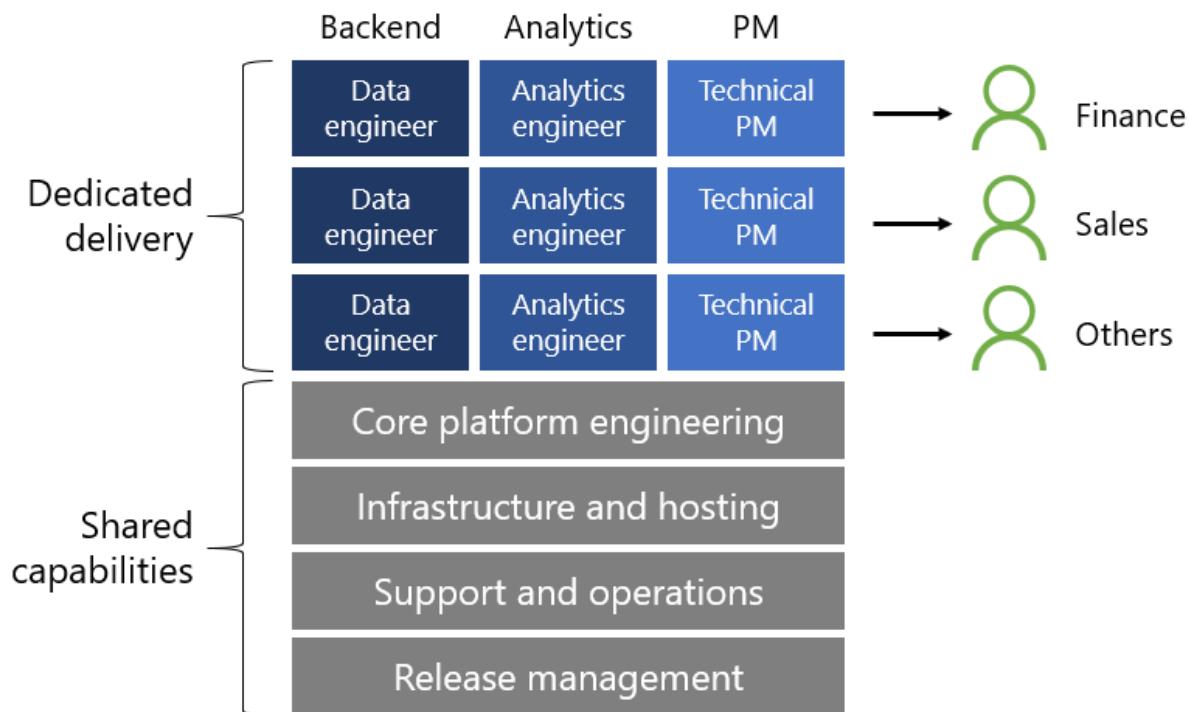
Importantly, a division of responsibility between the core and satellite COEs allows departmental analysts to focus on breaking new ground, rather than managing a data platform. At times, there can even be a mutually beneficial relationship between the satellite COEs and the core COE. For example, a 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 datasets into the centralized data warehouse. Onboarding a dataset 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 [Power BI 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.

Next steps

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

- [BI solution architecture in the COE](#)
- [Power BI adoption roadmap: Center of Excellence](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

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 engage a Power BI partner, visit the [Power BI partner portal](#).

You can also engage with experienced consulting partners. They can help you [assess](#), [evaluate](#), or [implement](#) Power BI.

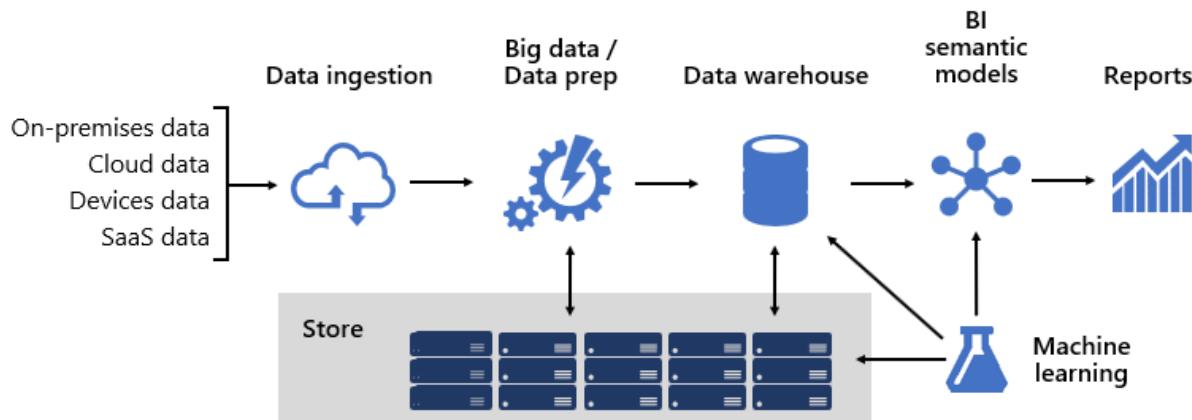
BI solution architecture in the Center of Excellence

3/24/2022 • 12 minutes to read • [Edit Online](#)

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](#) or [Power BI Premium capacities](#). 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.

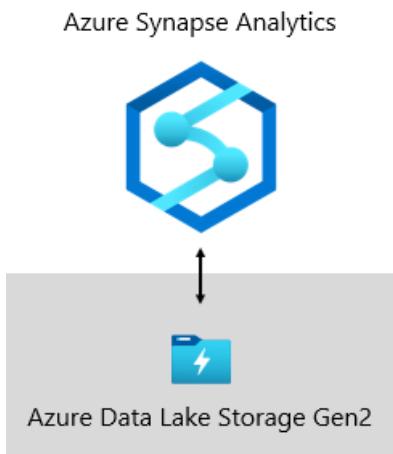
In a cloud BI platform, you can use [Azure Machine Learning](#) to train, deploy, automate, manage, and track ML models.

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

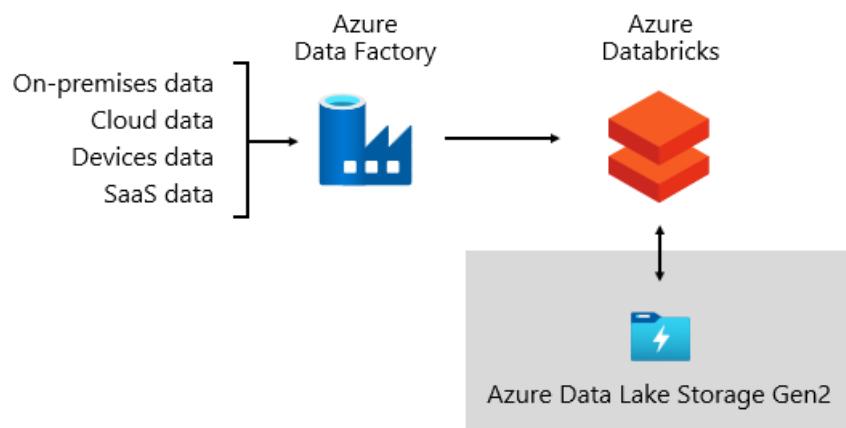
On a periodic basis, and according to the rhythms of the business, data is ingested from source systems and loaded into the data warehouse. It could be once a day or at more frequent intervals. Data ingestion is concerned with extracting, transforming, and loading data. Or, perhaps the other way round: 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. It 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.

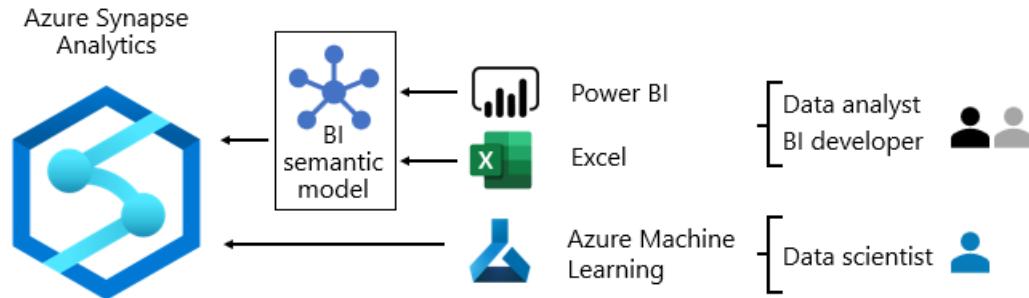
We publish data dictionaries, which provide reference information about our data models. They're made available to our users so they can discover information about our BI platform. Dictionaries document model designs, providing descriptions about entities, formats, structure, data lineage, relationships, 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.



Next steps

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

- [Power BI adoption roadmap: Center of Excellence](#)
- [Enterprise BI in Azure with Azure Synapse Analytics](#)
- Questions? [Try asking the Power BI Community](#)
- Suggestions? [Contribute ideas to improve Power BI](#)

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 engage a Power BI partner, visit the [Power BI partner portal](#).

You can also engage with experienced consulting partners. They can help you [assess](#), [evaluate](#), or [implement](#) Power BI.

White papers for Power BI

3/24/2022 • 2 minutes to read • [Edit Online](#)

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.

WHITE PAPER	DESCRIPTION	DATE
Planning a Power BI Enterprise Deployment	This updated technical white paper outlines considerations and best practices for a well-performing and secure organizational Power BI deployment.	June 2020
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
Distribute Power BI content to external guest users using Azure Active Directory B2B	This paper outlines how to distribute content to users outside the organization using the integration of Azure Active Directory Business-to-business (AAD B2B).	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

WHITE PAPER	DESCRIPTION	DATE
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 paper.	April 2018
Power BI migration overview	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](#)

Power BI security white paper

3/24/2022 • 47 minutes to read • [Edit Online](#)

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, datasets, 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, datasets, 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 are 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 and 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 very 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 very 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 cannot 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?*

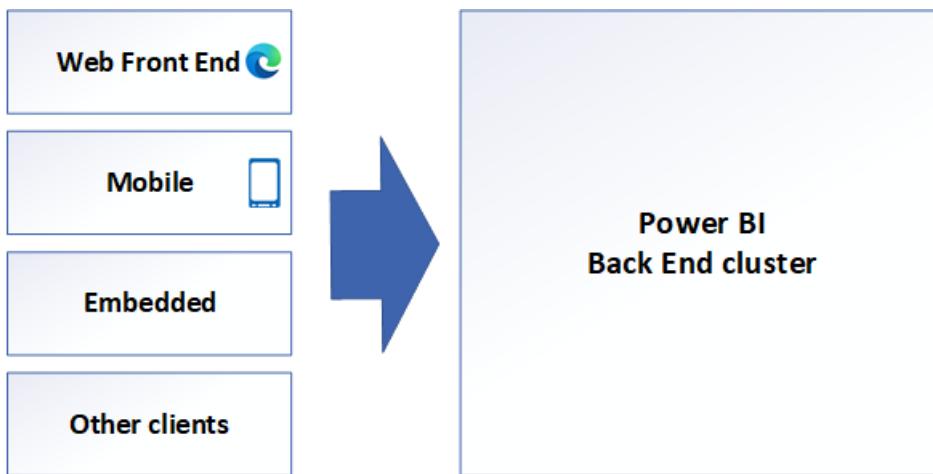
This article provides a comprehensive answer to all these questions. It starts with an overview of the service architecture and explains how the main flows in the system work. It then moves on to describe how users authenticate to Power BI, how data connections are established, and how Power BI stores 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](#).

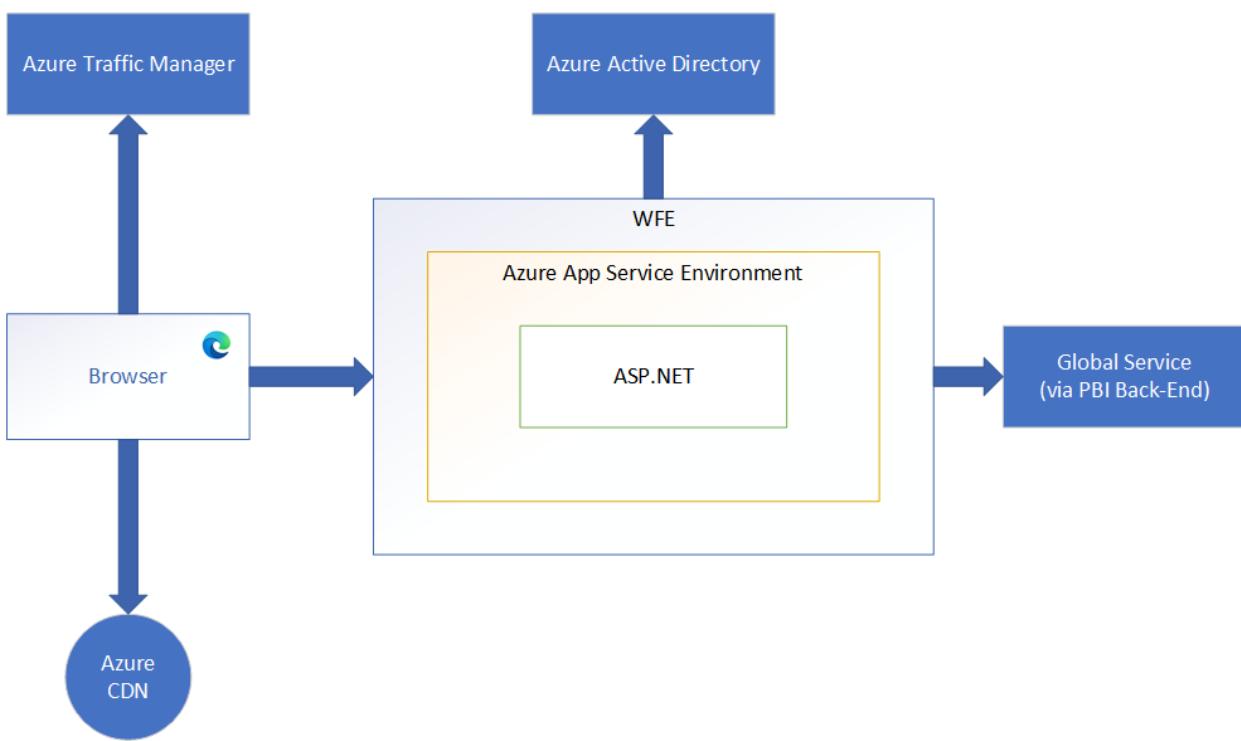
Power BI architecture

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 WFE cluster provides the user's browser with the initial HTML page contents on site load and manages the initial connection and authentication process for Power BI, using Azure Active Directory (Azure AD) to authenticate clients and provide tokens for subsequent client connections to the Power BI back-end service.



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 may 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](#).

The WFE cluster assigned to the user manages the login and authentication sequence (described later in this article) and obtains an Azure AD access token once authentication is successful. The ASP.NET component within the WFE cluster parses the token to determine which organization the user belongs to, and then consults the Power BI Global Service. The WFE specifies to the browser which back-end cluster houses the organization's tenant. Once a user is authenticated, subsequent client interactions for customer data occur with the back-end or Premium cluster directly, without the WFE being an intermediately for those requests.

Static resources such as `*.js`, `*.css`, and image files are mostly stored on Azure Content Delivery Network (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 (BE)

The back-end cluster is the backbone of all the functionality available in Power BI. It consists of several service endpoints consumed by Web Front End 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.

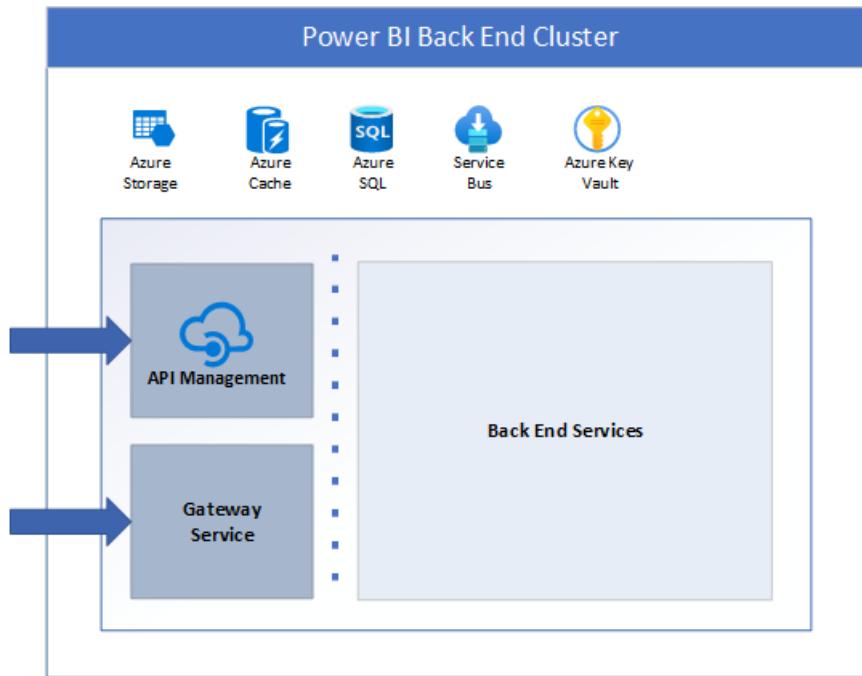
Each back-end cluster is stateful and hosts all the data of all the tenants assigned to that cluster. A cluster that contains the data of a specific tenant is referred to as the tenant's home cluster. An authenticated user's home cluster information is provided by Global Service and used by the Web Front End 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 are not accessible from the outside, except for two components that can be accessed from the public internet:

- Gateway Service
- Azure API Management

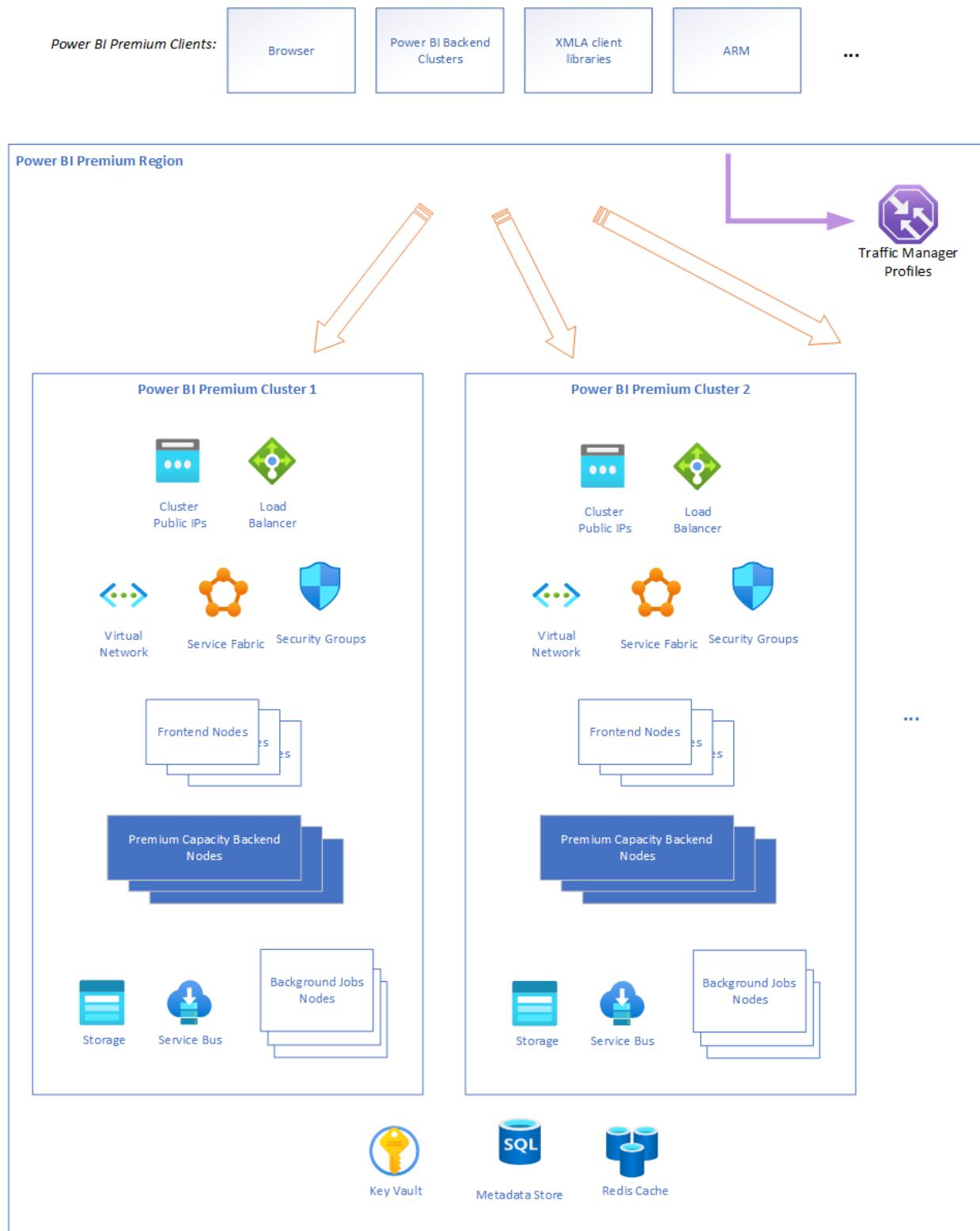


Power BI Premium infrastructure

Power BI Premium offers a service for subscribers who require premium Power BI features, such as Dataflows, Paginated Reports, AI, 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 – see 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 a number of 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). The majority of the 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 are 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 which 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 Azure AD exclusively.

Any request that comes to Power BI Premium infrastructure goes to front-end nodes first – they are 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 mobile device platform:

CBA SUPPORT	IOS	ANDROID	WINDOWS
Power BI (sign in to service)	Supported	Supported	Not supported
SSRS ADFS on-prem (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:

- Azure AD 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 is 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 which is accessible only to the app. For the Windows app, the data is only accessible by the user (and system admin).
- Geolocation is enabled or disabled explicitly by the user. If enabled, geolocation data is not saved on the device and is not shared with Microsoft.

- Notifications are enabled or disabled explicitly by the user. If enabled, Android and iOS do not 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 cannot 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 1st party apps (such as Microsoft Authenticator) and are managed by the Azure Active Directory Authentication Library (ADAL) SDK.

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 does not 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

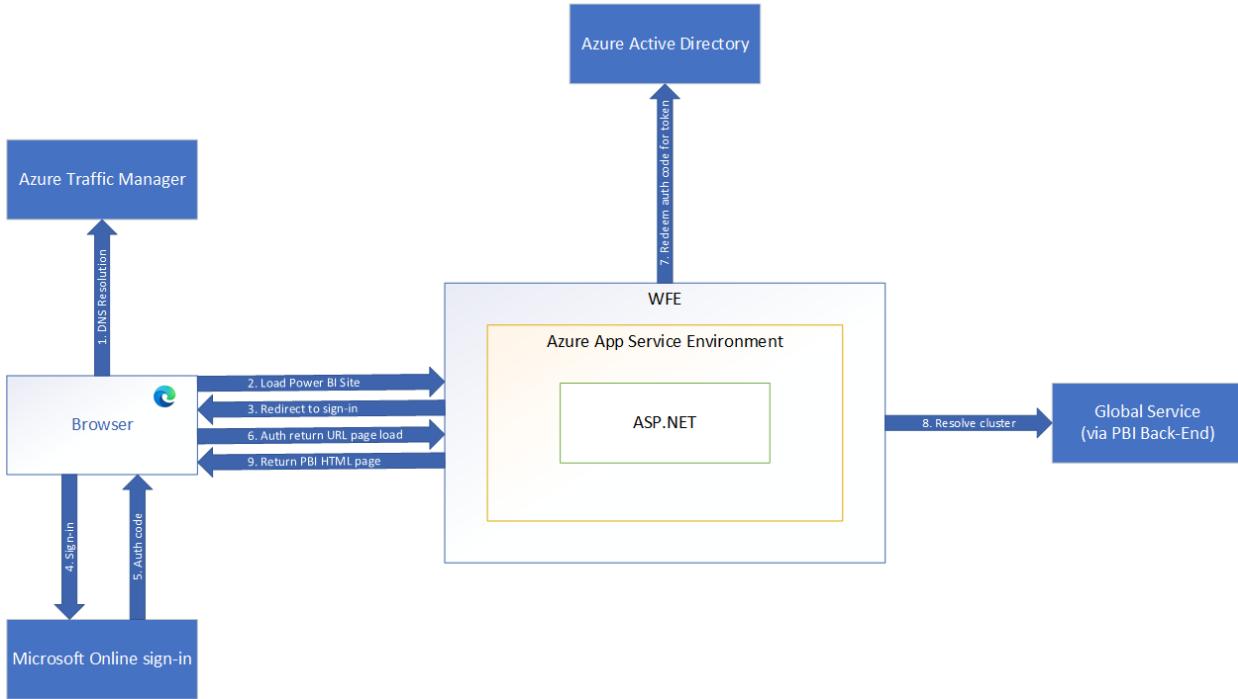
User authentication to the Power BI service consists of a series of requests, responses, and redirects between the user's browser and the Power BI service or the Azure services used by Power BI. That sequence describes the process of user authentication in Power BI, which follows the [Azure Active Directory's 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 landing 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 login page.
4. After the user has been authenticated, the login page redirects the user to the previously determined nearest Power BI service WFE cluster with an auth code.
5. The WFE cluster checks with the Azure AD service to obtain an Azure AD security token by using the auth code. When Azure AD returns the successful authentication of the user and returns an Azure AD 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 Azure AD access token in the Authorization header. The Power BI back-end cluster reads the Azure AD access token and validates the signature to ensure that the identity for the request is valid. The [Azure AD access token has a default lifetime of 1 hour](#), and to maintain the current session the user's browser will make periodic requests to renew the access token before it expires.



Data residency

Unless otherwise indicated in documentation, Power BI stores customer data in an Azure geography that is assigned when an [Azure AD tenant](#) signs up for Power BI services for the first time. An Azure AD 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 Azure AD 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 may require Power BI services in multiple Azure geographies. For example, a business may have their headquarters in the United States but may also do business in other geographical areas, such as Australia. In such cases the business may 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, may remain stored at rest in the tenant's home geo. Additionally, some data transit and processing may still happen in the tenant's home geo, even for workspaces that are hosted in a multi-geo Premium capacity.

Please see [Configure Multi-Geo support for Power BI Premium](#) for more information about creating and managing Power BI deployments that span multiple Azure geographies.

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 is used, please 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 clouds, see [Power BI national 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 the majority of 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 dataset. 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 will not be exposed – something that cannot easily be achieved using transparent service-side encryption. Please see [Bring your own encryption keys for Power BI](#) for more information.

Power BI datasets allow for a variety of data source connection modes which determine whether the data source data is persisted in the service or not.

DATASET MODE (KIND)	DATA PERSISTED IN POWER BI
Import	Yes
Direct Query	No
Live Connect	No
Composite	If contains an Import data source
Streaming	If configured to persist

Regardless of the dataset mode utilized, Power BI may temporarily cache any retrieved data to optimize query and report load performance.

Data in processing

Data is in processing when it is 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 is not 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 login and accesses the data with the credential. After the dataset 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 does not 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 dataset owner's credentials are used to connect to the data source.

If a data source is connected directly using pre-configured credentials, the pre-configured 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, Azure AD authentication is used for single sign on; for on-prem data sources, Kerberos, Security Assertion Markup Language (SAML), and Azure AD 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 row level security, and users who do not have sufficient credentials to access the underlying data (which could be a query used in a dashboard, report, or other data artifact) will not 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 may create a dataflow. Users in the viewer role may view data processed by the dataflow but may not make changes to its composition. Once a dataflow has been authored, any member, contributor, or admin of the workspace may 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 may 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 is encrypted. See the [Data in Processing](#) section above.

When customer specified data sources require credentials for access, the owner/creator of the dataflow will provide them during authoring. They are stored using standard product-wide credential storage. See the [Authentication to Data Sources](#) section above. There are various approaches users may 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 the [Data at Rest](#) section 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 ADLS storage account. Data is always transmitted to Blob storage using

encryption.

Since performance when accessing storage accounts may 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 a 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 Azure AD to determine if the user has sufficient rights to view the data. If so, a SaaS 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 security and privacy-related operations.

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. They're also called pixel perfect because you can control their report page layout exactly.

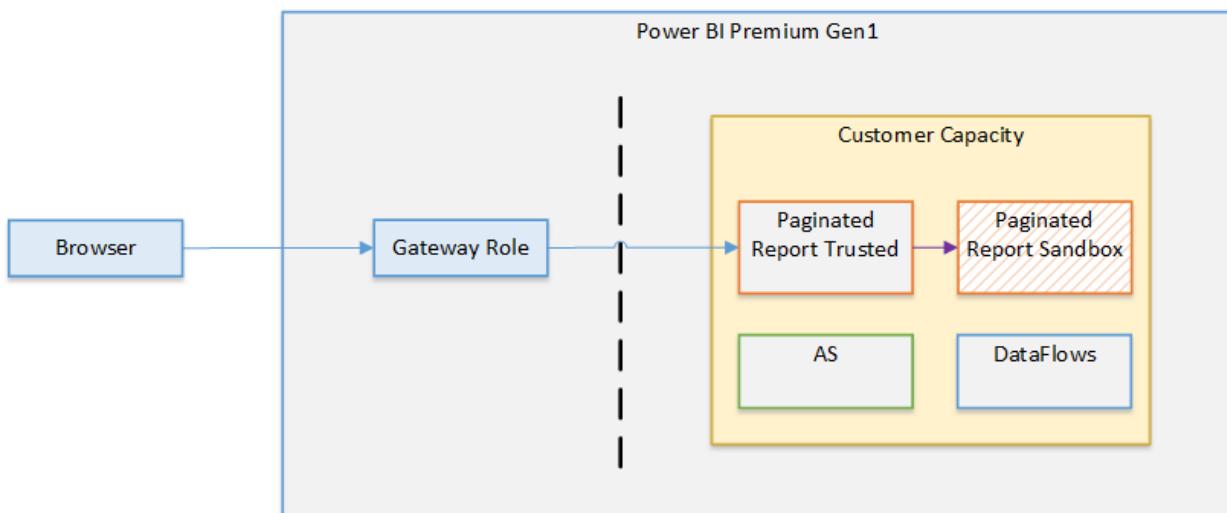
Paginated reports support rich and powerful expressions written in Microsoft Visual Basic .NET. Expressions are widely used throughout Power BI Report Builder paginated reports to retrieve, calculate, display, group, sort, filter, parameterize, and format data.

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 the [Authentication to the Power BI Service](#) section above.

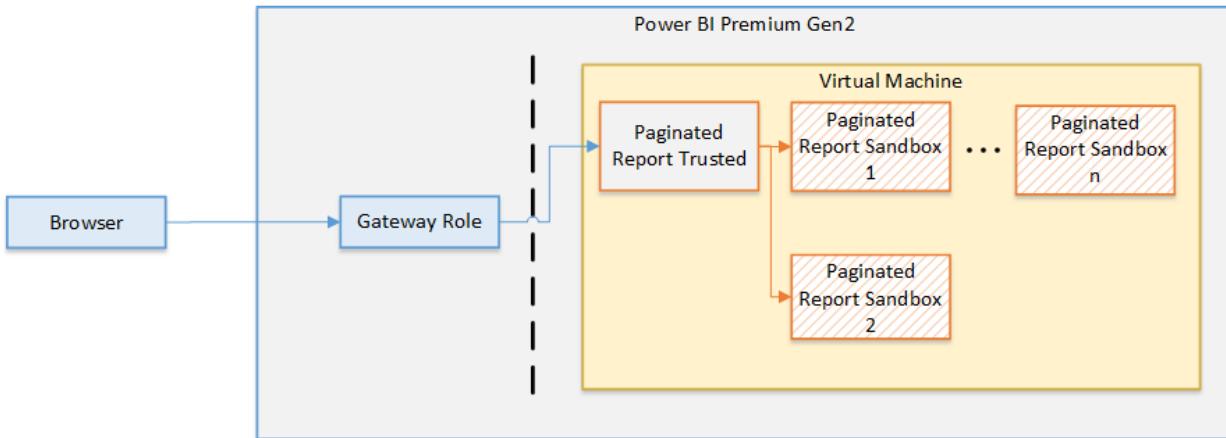
The Azure AD token obtained during the authentication is used to communicate directly from the browser to the Power BI Premium cluster.

For Premium Gen1, a single sandbox exists per each one of the capacities of the tenant, and is shared by the workspaces assigned to the capacity.



For Premium Gen2, an individual and exclusive ephemeral sandbox is created for each one of the renders of a

report, providing a higher level of isolation between users.



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, Azure AD 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 Row Level Security (RLS) and object-level security (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 artifacts (dashboards, reports, datasets etc.). 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 are not 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](#) Azure AD methods of authentication and authorization.

Power BI embedded analytics and its REST APIs support all Power BI network isolation capabilities described in this article: e.g., [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, AzureML, 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 ML model (e.g. 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/dataset 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 dataset or dataflow. Note that these functions can also be accessed from current dataflow/dataset 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 an Azure AD token for the Power BI user who is using the AI function. These AI data sources are special because they do not surface any of their own data and they only supply these functions/transforms. During execution, these features do not 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 AzureML service to select a suitable model and hyper-parameters for the current iteration. In this outbound call, only relevant experiment metadata (e.g. 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 does not 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 datasets. While authoring a dataset 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 Azure AD 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 Virtual Private Network (VPN) configured.

See [Private links for accessing Power BI](#) for additional information.

VNet connectivity (preview - coming soon)

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) will eliminate the overhead of installing and monitoring on-premises data gateways for connecting to data sources associated with a VNet. They will, however, 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 service (PP VNet).
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.

This feature will be available in public preview soon.

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 dataset tasks with service principals](#) for additional details.

Data loss prevention (DLP)

Microsoft 365 sensitivity labels

Power BI has a deep integration with Microsoft Information Protection (MIP) sensitivity labels, which enable organizations to have a single, integrated solution for DLP policy management, audit, and compliance across the Office suite.

When sensitivity labels are 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 familiar Microsoft Information Protection sensitivity labels used in Office and in Azure Purview.
- Governance policies can be enforced, even when Power BI content is exported to Excel, PowerPoint, PDF or .pbix files, to help ensure data is protected even when it leaves Power BI.
- .pbix files can be encrypted according to MIP label policies when a MIP label is applied on the .pbix file in Desktop, ensuring that only authorized users can edit this file.

- It's easy to classify and protect .pbix files just like it is done with Excel, Word, and PowerPoint files. With just two clicks, a file can be tagged according to its level of sensitivity, and, even further, be encrypted if it contains business-confidential data.
- Excel workbooks automatically inherit the sensitivity labels when they connect to Power BI (preview), making it possible to maintain end-to-end classification and apply protection when the Power BI dataset is analyzed in Excel.
- Sensitivity labels applied on Power BI reports and dashboards will be visible in the Power BI iOS and Android mobile apps.
- Sensitivity labels will persist when a Power BI report is embedded in Teams, SharePoint, or a secure website (preview). 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 the label applied on a dataset in the Power BI service will be applied on new content created on top of the dataset.
- [Power BI admin scan APIs](#) can extract a Power BI artifact's sensitivity label, enabling Power BI and InfoSec admins to monitor labeling in the Power BI service and produce executive reports.
- Power BI makes sure that only authorized users can change or remove labels with protection settings in the Power BI service.
- Coming soon:
 - Power BI admin APIs for applying MIP labels to enable central teams to programmatically label content in the Power BI service.
 - Admins will be able to enforce applying labels on new or edited content with a mandatory label policy in the Power BI service (preview).
 - Automatic downstream artifact labeling within the Power BI service. When a label on a dataset is applied or changed, the label will automatically be applied on all downstream content connected to this artifact.

See the [Microsoft Information Protection sensitivity label documentation in Power BI](#) for additional details.

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, 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 are 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.

See [Using Microsoft Defender for Cloud Apps Controls in Power BI](#) for additional details.

Preview security features

This topic lists features that are planned to release through March 2021. Because this topic lists features that may not have released yet, **delivery timelines may change and projected functionality may be released later than March 2021, or may not be released at all**. For more information about previews, please review the [Online Services Terms](#).

Bring Your Own Log Analytics (BYOLA)

Bring Your Own Log Analytics enables integration between Power BI and Azure Log Analytics. This integration includes Azure Log Analytics' advanced analytic engine, interactive query language, and built-in machine learning constructs.

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 dataset, 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 login and accesses the data with the credential. After the dataset 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 dashboard does not 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 dataset 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 pre-configured credential, the pre-configured 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, Row Level Security (RLS) and/or object-level security (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, Azure AD authentication is used for single sign-on; for on-prem data sources, Kerberos, SAML and Azure AD 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 datasource. 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 Row Level Security

(RLS) and/or object-level security (OLS) is configured, the Power BI service will apply that row level security, and users who do not have sufficient credentials to access the underlying data (which could be a query used in a dashboard, report, or other data artifact) will not see data for which the user does not have sufficient privileges.

[Row Level security 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 role.

[Object-level security \(OLS\)](#) can be used to secure sensitive tables or columns. However, unlike row-level security, object-level security 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 cannot access secured metadata objects via DAX or any other method. From the standpoint of users without proper access permissions, secured tables and columns simply do not exist.

Object-level security, together with row-level security, enables enhanced enterprise grade security on reports and datasets, 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 does not 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 the [Authentication to Data Sources](#) section earlier in this document.

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-Role Level Security (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 is shared to view and interact with. Power BI *does not* re-authenticate 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 dataset 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 the [Authentication to Data Sources](#) section earlier in this document.

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 [Power BI Personal Gateway](#).

Which ports are used by 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 at the following link: [Gateway ports](#)

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 Content Delivery Network (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 independent service providers), browser configuration files used to initiate and establish any subsequent connections with the Power BI service, as well as the initial secure Power BI login 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 Azure AD 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 is 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 does not 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 is 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 is 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 data centers located in specific geographies, to ensure data doesn't leave the country borders?

- Some customers in certain geographies have an option to create a tenant in a national cloud, where data storage and processing is kept separate from all other datacenters. National clouds have a slightly different type of security, since a separate data trustee operates the national 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 clouds is different from the generally available commercial Power BI service. For more information about Power BI service availability for national clouds, see [Power BI national 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 an [Azure Business-to-Business \(B2B\)](#) authorization process, using Azure AD 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 Azure AD user.

Additional resources

For more information on Power BI, see the following resources.

- [Getting Started with Power BI Desktop](#)
- [Power BI REST API - Overview](#)
- [Power BI API reference](#)
- [On-premises data gateway](#)
- [Power BI National Clouds](#)
- [Power BI Premium](#)
- [Overview of single sign-on \(SSO\) for gateways in Power BI](#)

Power BI enterprise deployment whitepaper

3/24/2022 • 2 minutes to read • [Edit Online](#)

Deploying Power BI in a large enterprise is a complex task that requires a lot of thought and planning. Getting proper guidance and best practices can help you understand the choices you will make, gather requirements, and learn best practices that can make your Power BI enterprise deployment a success. To facilitate those steps, and more, Microsoft is providing the Power BI Enterprise Deployment whitepaper.

About the whitepaper

The purpose of the Enterprise Deployment whitepaper is to help make your Power BI deployment a success: it covers key considerations, the decisions which will be necessary throughout the process, and potential issues you may encounter. Best practices and suggestions are offered when possible.

The target audience for this whitepaper is technology professionals. Some knowledge of Power BI and general business intelligence concepts is assumed.

You can [download the complete enterprise deployment whitepaper](#) at the following link:

- [Planning a Power BI Enterprise Deployment whitepaper](#)

Next steps

For more information on Power BI, see the following resources:

- [Whitepapers for Power BI](#)
- [Power BI security whitepaper](#)
- [Power BI Premium](#)

Deploying and Managing Power BI Premium Capacities

3/24/2022 • 2 minutes to read • [Edit Online](#)

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.

ARTICLES	DESCRIPTION
Basic concepts for designers in the Power BI service Datasets in the Power BI service Dataset modes in the Power BI service	Background information about Power BI service capacities, workspaces, dashboards, reports, workbooks, datasets, 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.
Optimizing Premium capacities	Best practices for performance optimization, optimizing models, capacity planning, and testing approaches.
Premium capacity scenarios	Common issues in real-world scenarios, with a focus on identifying and resolving those issues.
Monitor capacities in the Admin portal	Monitoring with Power BI Premium 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.

Distribute Power BI content to external guest users using Azure Active Directory B2B

3/24/2022 • 42 minutes to read • [Edit Online](#)

Summary: This is a technical whitepaper outlining how to distribute content to users outside the organization using the integration of Azure Active Directory Business-to-business (Azure AD B2B).

Writers: Lukasz Pawlowski, Kasper de Jonge

Technical Reviewers: Adam Wilson, Sheng Liu, Qian Liang, Sergei Gundorov, Jacob Grimm, Adam Saxton, Maya Shenhav, Nimrod Shalit, Elisabeth Olson

NOTE

You can save or print this whitepaper by selecting **Print** from your browser, then selecting **Save as PDF**.

Introduction

Power BI gives organizations a 360-degree view of their business and empowers everyone in these organizations to make intelligent decisions using data. Many of these organizations have strong and trusted relationships with external partners, clients, and contractors. These organizations need to provide secure access to Power BI dashboards and reports to users in these external partners.

Power BI integrates with [Azure Active Directory Business-to-business \(Azure AD B2B\)](#) to allow secure distribution of Power BI content to guest users outside the organization – while still maintaining control and governing access to internal data.

This white paper covers the all the details you need to understand Power BI's integration with Azure Active Directory B2B. We cover its most common use case, setup, licensing, and row level security.

NOTE

Throughout this white paper, we refer to Azure Active Directory as Azure AD and Azure Active Directory Business to Business as Azure AD B2B.

Scenarios

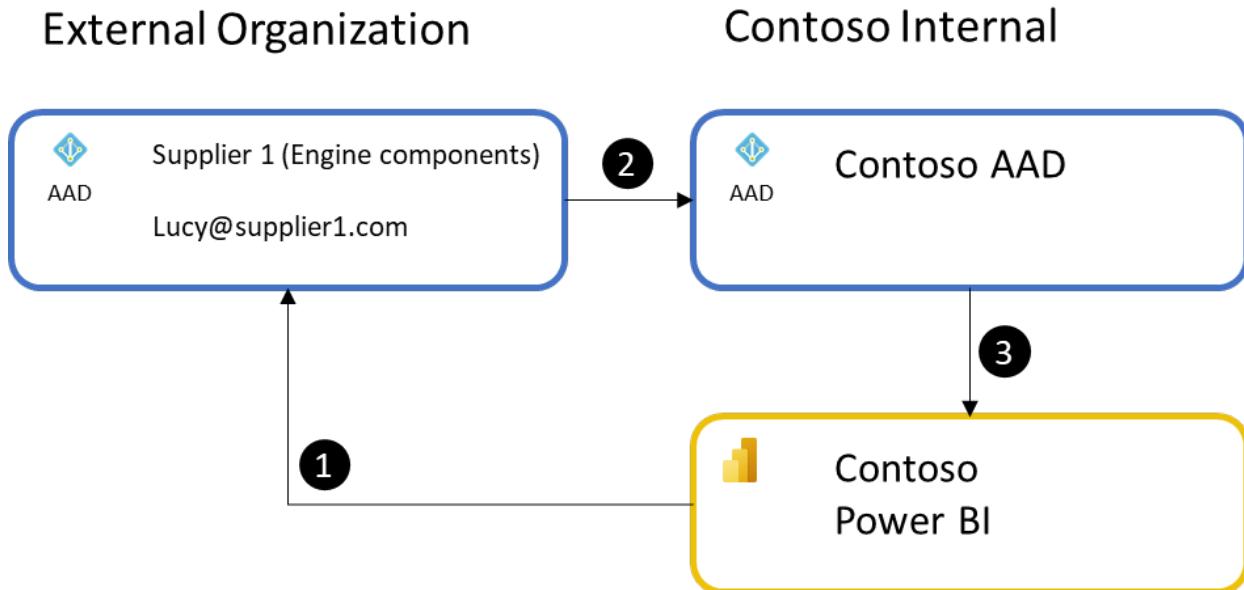
Contoso is an automotive manufacturer and works with many diverse suppliers who provide it with all the components, materials, and services necessary to run its manufacturing operations. Contoso wants to streamline its supply chain logistics and plans to use Power BI to monitor key performance metrics of its supply chain. Contoso wants to share with external supply chain partners analytics in a secure and manageable way.

Contoso can enable the following experiences for external users using Power BI and Azure AD B2B.

Ad hoc per item sharing

Contoso works with a supplier who builds radiators for Contoso's cars. Often, they need to optimize the reliability of the radiators using data from all of Contoso's cars. An analyst at Contoso uses Power BI to share a radiator reliability report with an Engineer at the supplier. The Engineer receives an email with a link to view the report.

As described above, this ad-hoc sharing is performed by business users on an as needed basis. The link sent by Power BI to the external user is an Azure AD B2B invite link. When the external user opens the link, they are asked to join Contoso's Azure AD organization as a Guest user. After the invite is accepted, the link opens the specific report or dashboard. The Azure Active Directory admin delegates permission to invite external users to the organization and chooses what those users can do once they accept the invite as described in the Governance section of this document. The Contoso analyst can invite the Guest user only because the Azure AD administrator allowed that action and the Power BI administrator allowed users to invite guests to view content in Power BI's tenant settings.



1. The process starts with a Contoso internal user sharing a dashboard or a report with an external user. If the external user is not already a guest in Contoso's Azure AD, they are invited. An email is sent to their email address that includes an invite to Contoso's Azure AD
2. The recipient accepts the invite to Contoso's Azure AD and is added as a Guest user in Contoso's Azure AD.
3. The recipient is then redirected to the Power BI dashboard, report, or app, which are read-only for the user.

The process is considered ad-hoc since business users in Contoso perform the invite action as needed for their business purposes. Each item shared is a single link the external user can access to view the content.

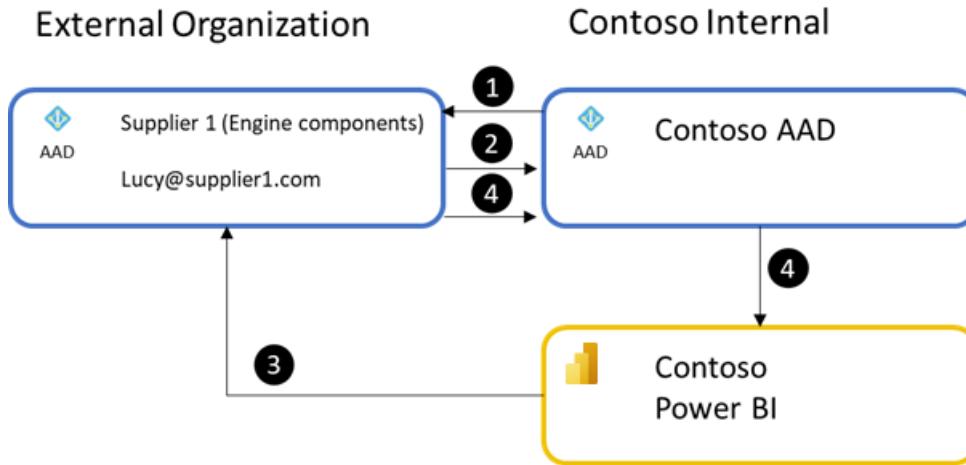
Once the external user has been invited to access Contoso resources, a shadow account may be created for them in Contoso Azure AD and they do not need to be invited again. The first time they try to access a Contoso resource like a Power BI dashboard, they go through a consent process, which redeems the invitation. If they do not complete the consent, they cannot access any of Contoso's content. If they have trouble redeeming their invitation via the original link provided, an Azure AD administrator can resend a specific invitation link for them to redeem.

Planned per item sharing

Contoso works with a subcontractor to perform reliability analysis of radiators. The subcontractor has a team of 10 people who need access to data in Contoso's Power BI environment. The Contoso Azure AD administrator is involved to invite all the users and to handle any additions/changes as personnel at the subcontractor change. The Azure AD administrator creates a security group for all the employees at the subcontractor. Using the security group, Contoso's employees can easily manage access to reports and ensure all required subcontractor personnel have access to all the required reports, dashboards, and Power BI apps. The Azure AD administrator can also avoid being involved in the invitation process altogether by choosing to delegate invitation rights to a trusted employee at Contoso or at the subcontractor to ensure timely personnel management.

Some organizations require more control over when external users are added, are inviting many users in an external organization, or many external organizations. In these cases, planned sharing can be used to manage

the scale of sharing, to enforce organizational policies, and even to delegate rights to trusted individuals to invite and manage external users. Azure AD B2B supports planned invites to be sent directly from the Azure portal by an IT administrator, or through PowerShell using the invitation manager API where a set of users can be invited in one action. Using the planned invites approach, the organization can control who can invite users and implement approval processes. Advanced Azure AD capabilities like dynamic groups can make it easy to maintain security group membership automatically.

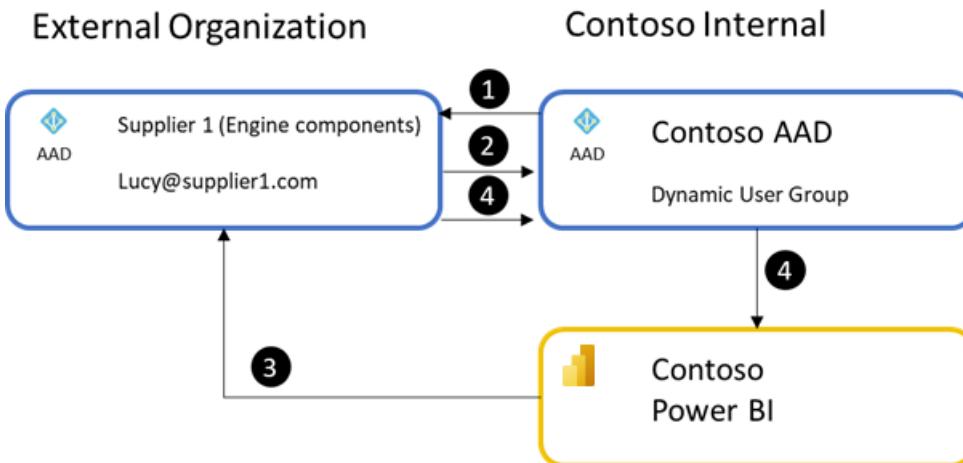


1. The process starts with an IT administrator inviting the guest user either manually or through the API provided by Azure Active Directory
2. The user accepts the invite to the organization.
3. Once the user has accepted the invitation, a user in Power BI can share a report or dashboard with the external user, or a security group they are in. Just like with regular sharing in Power BI the external user receives an email with the link to the item.
4. When the external user accesses the link, their authentication in their directory is passed to Contoso's Azure AD and used to gain access to the Power BI content.

Ad hoc or planned sharing of Power BI Apps

Contoso has a set of reports and dashboards they need to share with one or more Suppliers. To ensure all required external users have access to this content, it is packaged as a Power BI app. The external users are either added directly to the app access list or through security groups. Someone at Contoso then sends the app URL to all the external users, for example in an email. When the external users open the link, they see all the content in a single easy to navigate experience.

Using a Power BI app makes it easy for Contoso to build a BI Portal for its suppliers. A single access list controls access to all the required content reducing wasted time checking and setting item level permissions. Azure AD B2B maintains security access using the Supplier's native identity so users don't need additional login credentials. If using planned invites with security groups, access management to the app as personnel rotate into or out of the project is simplified. Membership in security groups manually or by using dynamic groups, so that all external users from a supplier are automatically added to the appropriate security group.



Apps also have a unique feature that allows app authors to install the application automatically for the user, so it is available when the user logs in. This feature only installs automatically for external users who are already part of Contoso's organization at the time the application is published or updated. Thus, it is best used with the planned invites approach, and depends on the app being published or updated after the users are added to Contoso's Azure AD. External users can always install the app using the app link.

Commenting and subscribing to content across organizations

As Contoso continues to work with its subcontractors or suppliers, the external Engineers need to work closely with Contoso's analysts. Power BI provides several collaboration features that help users communicate about content they can consume. Dashboard commenting (and soon Report commenting) allows users to discuss data points they see and communicate with report authors to ask questions.

Currently, external guest users can participate in comments by leaving comments and reading the replies. However, unlike internal users, guest users cannot be @mentioned and do not receive notifications that they've received a comment. Guest users cannot use the subscriptions feature within Power BI at the time of writing. In an upcoming release, those restrictions will be lifted and the Guest user will receive an email when a comment @mentions them, or when a subscription is delivered to their email that contains a link to the content in Power BI.

Access content in the Power BI mobile apps

In an upcoming release, when Contoso's users share reports or dashboards with their external Guest counterparts, Power BI will send an email notifying the Guest. When the guest user opens the link to the report or dashboard on their mobile device, the content will open in the native Power BI mobile apps on their device, if they're installed. The guest user will then be able to navigate between content shared with them in the external tenant, and back to their own content from their home tenant.

NOTE

The guest user cannot open the Power BI mobile app and immediately navigate to the external tenant, they must start with a link to an item in the external tenant. Common workarounds are described in the [Distributing links to content in the Parent organization's Power BI](#) section later in this document.

Cross-organization editing and management of Power BI content

Contoso and its Suppliers and subcontractors work increasingly closely together. Often an analyst at the subcontractor needs additional metrics or data visualizations to be added to a report Contoso has shared with them. The data should reside in Contoso's Power BI tenant, but external users should be able to edit it, create new content, and even distribute it to appropriate individuals.

Power BI provides an option that enables **External guest users can edit and manage content** in the organization. By default, external users have a read-only consumption-oriented experience. However, this new setting allows the Power BI admin to choose which external users can edit and manage content within their own organization. Once allowed, the external user can edit reports, dashboards, publish or update apps, work in workspaces, and connect to data they have permission to use.

This scenario is described in detail in the section [Enabling external users to edit and manage content within Power BI](#) later in this document.

Organizational relationships using Power BI and Azure AD B2B

When all the users of Power BI are internal to the organization, there is no need to use Azure AD B2B. However, once two or more organizations want to collaborate on data and insights, Power BI's support for Azure AD B2B makes it easy and cost effective to do so.

Below are typically encountered organizational structures that are well suited for Azure AD B2B style cross-organization collaboration in Power BI. Azure AD B2B works well in most cases, but in some situations the Common alternative approaches covered at the end of this document are worth considering.

Case 1: Direct collaboration between organizations

Contoso's relationship with its radiator supplier is an example of direct collaboration between organizations. Since there are relatively few users at Contoso and its supplier who need access to radiator reliability information, using Azure AD B2B based external sharing is ideal. It is easy to use and simple to administer. This is also a common pattern in consulting services where a consultant may need to build content for an organization.

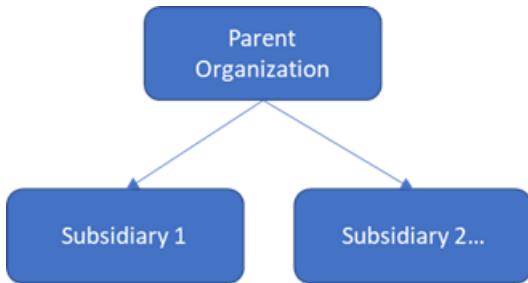


Typically, this sharing occurs initially using Ad hoc per item sharing. However, as teams grow or relationships deepen, the Planned per item sharing approach becomes the preferred method to reduce management overhead. Additionally, the Ad hoc or planned sharing of Power BI Apps, Commenting and subscribing to content across organizations, access to content in mobile apps can come into play as well, and cross-organization editing and management of Power BI content. Importantly, if both organizations' users have Power BI Pro licenses in their respective organizations, they can use those Pro licenses in each other's Power BI environments. This provides advantageous licensing since the inviting organization may not need to pay for a Power BI Pro license for the external users. This is discussed in more detail in the Licensing section later in this document.

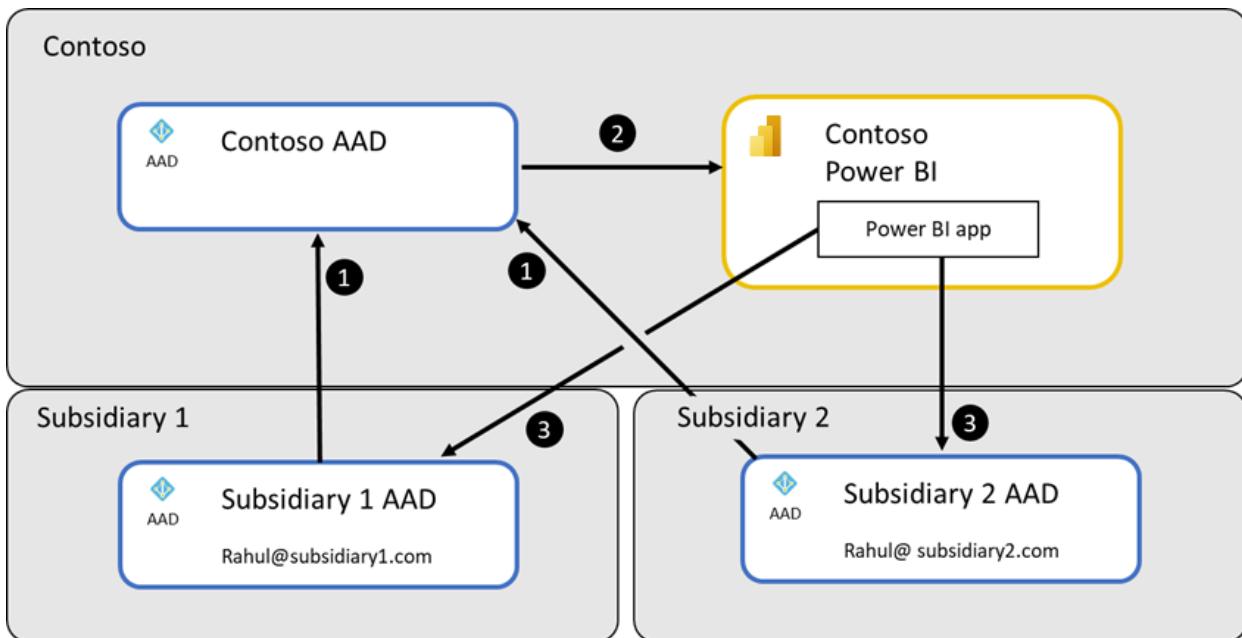
Case 2: Parent and its subsidiaries or affiliates

Some organization structures are more complex, including partially or wholly owned subsidiaries, affiliated companies, or managed service provider relationships. These organizations have a parent organization such as a holding company, but the underlying organizations operate semi-autonomously, sometimes under different

regional requirements. This leads to each organization having its own Azure AD environment and separate Power BI tenants.



In this structure, the parent organization typically needs to distribute standardized insights to its subsidiaries. Typically, this sharing occurs using the Ad hoc or planned sharing of Power BI Apps approach as illustrated in the following image, since it allows distribution of standardized authoritative content to broad audiences. In practice a combination of all the Scenarios mentioned earlier in this document is used.



This follows the following process:

1. Users from each Subsidiary are invited to Contoso's Azure AD
2. Then the Power BI app is published to give these users access to the required data
3. Finally, the users open the app through a link they've been given to see the reports

Several important challenges are faced by organizations in this structure:

- How to distribute links to content in the Parent organization's Power BI
- How to allow subsidiary users to access data source hosted by the parent organization

Distributing links to content in the Parent organization's Power BI

Three approaches are commonly used to distribute links to the content. The first and most basic is to send the link to the app to the required users or to place it in a SharePoint Online site from which it can be opened. Users can then bookmark the link in their browsers for faster access to the data they need.

The second approach relies on the cross-organization editing and management of Power BI content capability. The Parent organization allows users from the subsidiaries to access its Power BI and controls what they can access through permission. This gives access to Power BI Home where the user from the subsidiary sees a comprehensive list of content shared to them in the Parent organization's tenant. Then the URL to the Parent organizations' Power BI environment is given to the users at the subsidiaries.

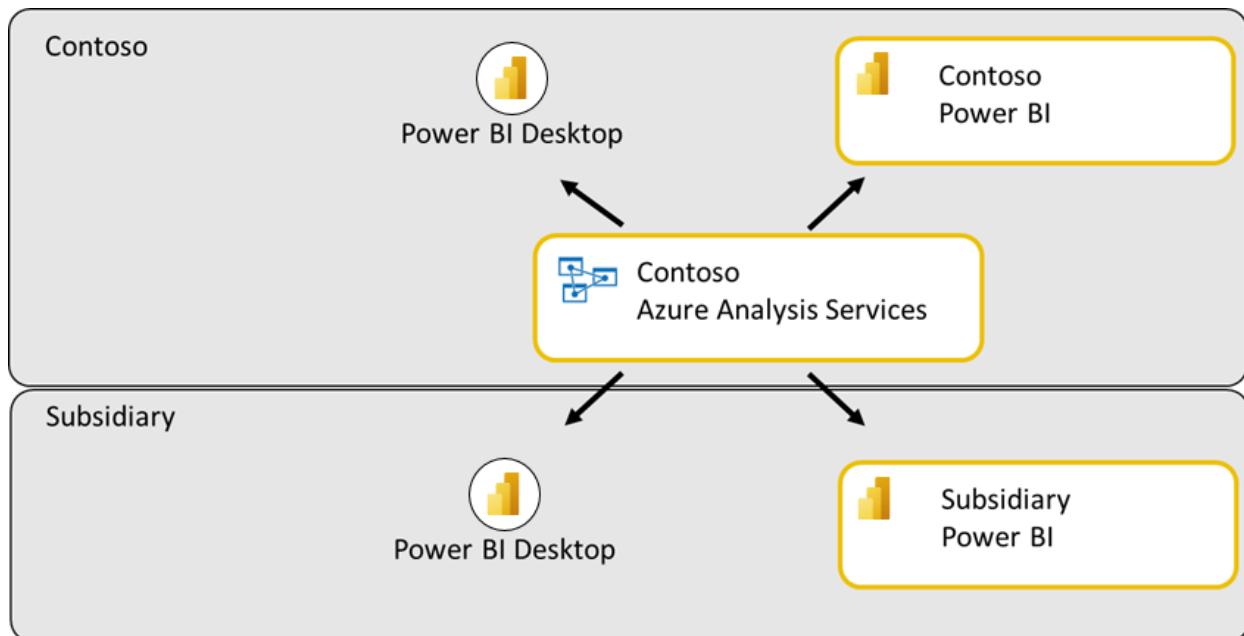
The final approach uses a Power BI app created within the Power BI tenant for each subsidiary. The Power BI app

includes a dashboard with tiles configured with the external link option. When the user presses the tile, they are taken to the appropriate report, dashboard, or app in the parent organization's Power BI. This approach has the added advantage that the app can be installed automatically for all users in the subsidiary and is available to them whenever they sign in to their own Power BI environment. An added advantage of this approach is that it works well with the Power BI mobile apps that can open the link natively. You can also combine this with the second approach to enable easier switching between Power BI environments.

Allowing subsidiary users to access data sources hosted by the parent organization

Often analysts at a subsidiary need to create their own analytics using data supplied by the parent organization. In this case, commonly cloud data sources are used to address the challenge.

The first approach leverages [Azure Analysis Services](#) to build an enterprise grade data warehouse that serves the needs of Analysts across the parent and its subsidiaries as shown the following image. Contoso can host the data and use capabilities like row level security to ensure users in each subsidiary can access only their data. Analysts at each organization can access the data warehouse through Power BI Desktop and publish resulting analytics to their respective Power BI tenants.

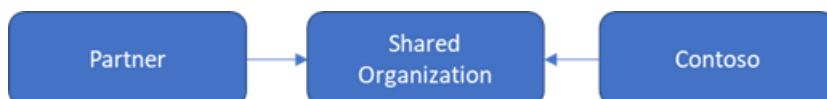


The second approach leverages [Azure SQL Database](#) to build a relational data warehouse to provide access to data. This works similarly to the Azure Analysis Services approach, though some capabilities like row level security may be harder to deploy and maintain across subsidiaries.

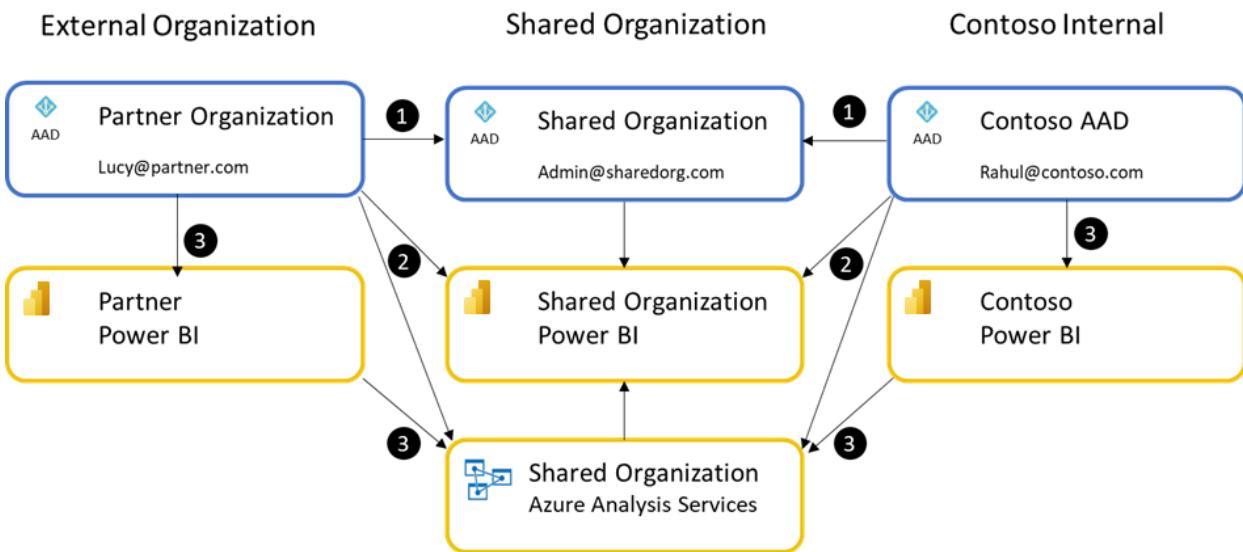
More sophisticated approaches are also possible, however the above are by far the most common.

Case 3: Shared environment across partners

Contoso may enter into a partnership with a competitor to jointly build a car on a shared assembly line, but to distribute the vehicle under different brands or in different regions. This requires extensive collaboration and co-ownership of data, intelligence, and analytics across organizations. This structure is also common in the consulting services industry where a team of consultants may do project-based analytics for a client.



In practice, these structures are complex as shown in the following image, and require staff to maintain. To be effective this structure relies on the cross-organization editing and management of Power BI content capability since it allows organizations to reuse Power BI Pro licenses purchased for their respective Power BI tenants.



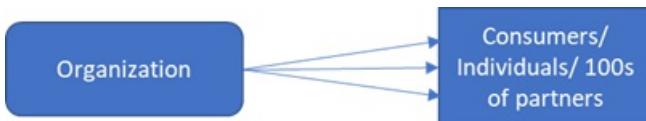
To establish a shared Power BI tenant, an Azure Active Directory needs to be created and at least one Power BI Pro user account needs to be purchased for a user in that active directory. This user invites the required users to the shared organization. Importantly, in this scenario, Contoso's users are treated as external users when they operate within the Shared Organization's Power BI.

The process is as follows:

1. The Shared Organization is established as a new Azure Active Directory and at least one user account is created in the new organization. That user should have a Power BI Pro license assigned to them.
2. This user then establishes a Power BI tenant and invites the required users from Contoso and the Partner organization. The user also establishes any shared data assets like Azure Analysis Services. Contoso and the Partner's users can access the shared organization's Power BI as guest users. If allowed to edit and manage content in Power BI the external users can use Power BI home, use workspaces, upload, or edit content and share reports. Typically, all shared assets are stored and accessed from the shared organization.
3. Depending on how the parties agree to collaborate, it is possible for each organization to develop their own proprietary data and analytics using shared data warehouse assets. They can distribute those to their respective internal users using their internal Power BI tenants.

Case 4: Distribution to hundreds or thousands of external partners

While Contoso created a radiator reliability report for one Supplier, now Contoso desires to create a set of standardized reports for hundreds of Suppliers. This allows Contoso to ensure all suppliers have the analytics they need to make improvements or to fix manufacturing defects.



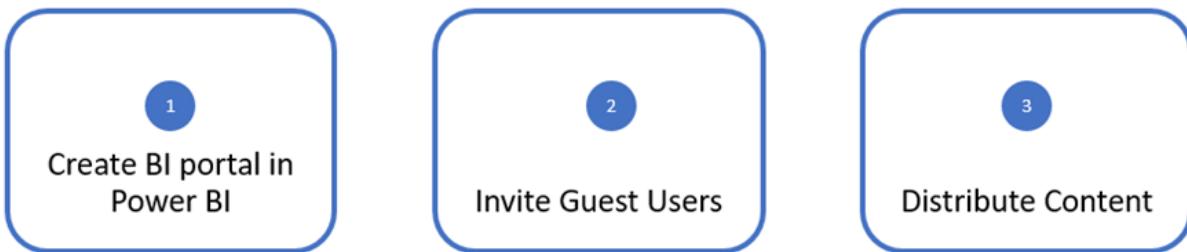
When an organization needs to distribute standardized data and insights to many external users/organizations, they can use the Ad hoc or planned sharing of Power BI Apps scenario to build a BI Portal quickly and without extensive development costs. The process to build such a portal using a Power BI app is covered in the Case Study: Building a BI Portal using Power BI + Azure AD B2B – Step-by-Step instructions later in this document.

A common variant of this case is when an organization is attempting to share insights with consumers, especially when looking to use Azure B2C with Power BI. Power BI does not natively support Azure B2C. If you're evaluating options for this case, consider using Alternative Option 2 in the Common alternative approaches the section later in this document.

Case Study: Building a BI Portal using Power BI + Azure AD B2B –

Step-by-Step instructions

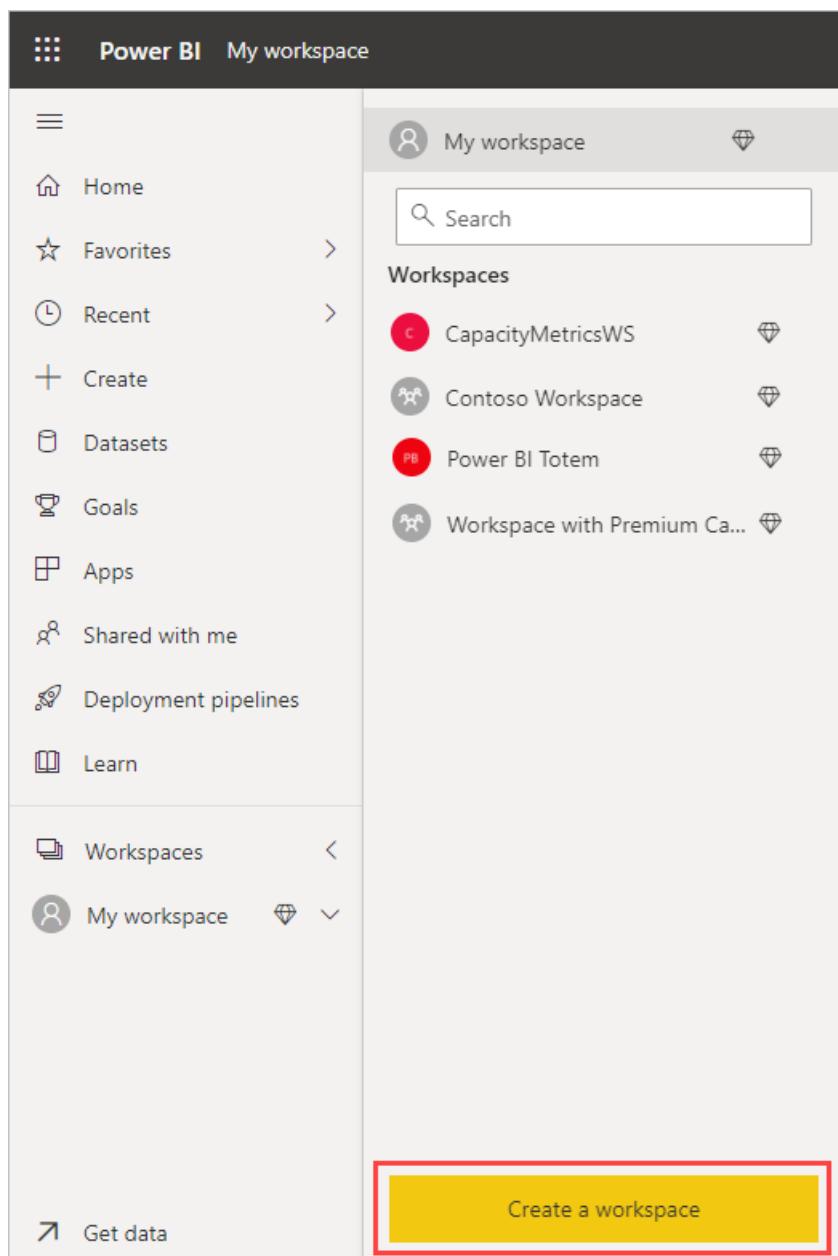
Power BI's integration with Azure AD B2B gives Contoso a seamless, hassle-free way to provide guest users with secure access to its BI portal. Contoso can set this up with three steps:



1. Create BI portal in Power BI

The first task for Contoso is to create their BI portal in Power BI. Contoso's BI portal will consist of a collection of purpose-built dashboards and reports that will be made available to many internal and guest users. The recommended way for doing this in Power BI is to build a Power BI app. Learn more about [apps in Power BI](#).

- Contoso's BI team creates a workspace in Power BI



- Other authors are added to the workspace

Workspace name

Power BI Portal

Available

Description

Describe this workspace

[Learn more about workspace settings](#)

Advanced ▾

Contact list

Workspace admins

Specific users and groups

Enter users and groups

Workspace OneDrive

Save Cancel

- Content is created inside the workspace

Name	Type	Owner
COGS by Date	Report	Power BI Portal
Financial Sample	Dataset	Power BI Portal

Now that the content is created in a workspace, Contoso is ready to invite guest users in partner organizations to consume this content.

2. Invite Guest Users

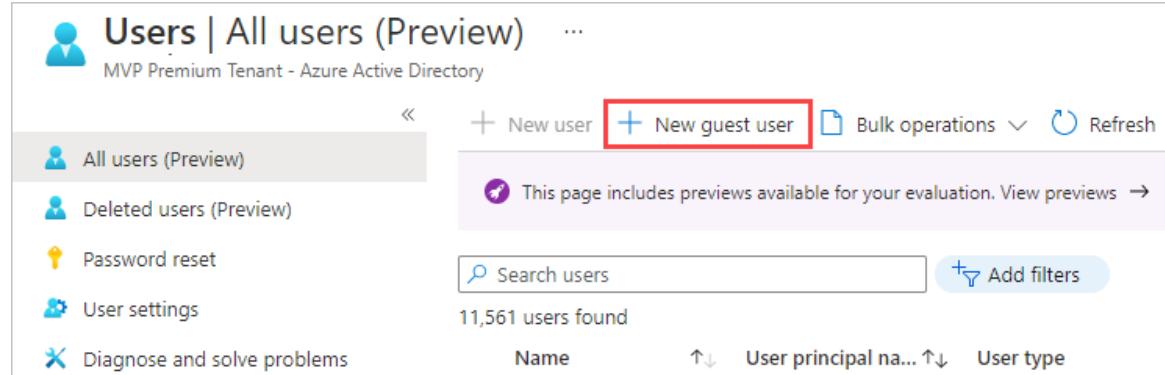
There are two ways for Contoso to invite guest users to its BI portal in Power BI:

- Planned Invites
- Ad hoc Invites

Planned Invites

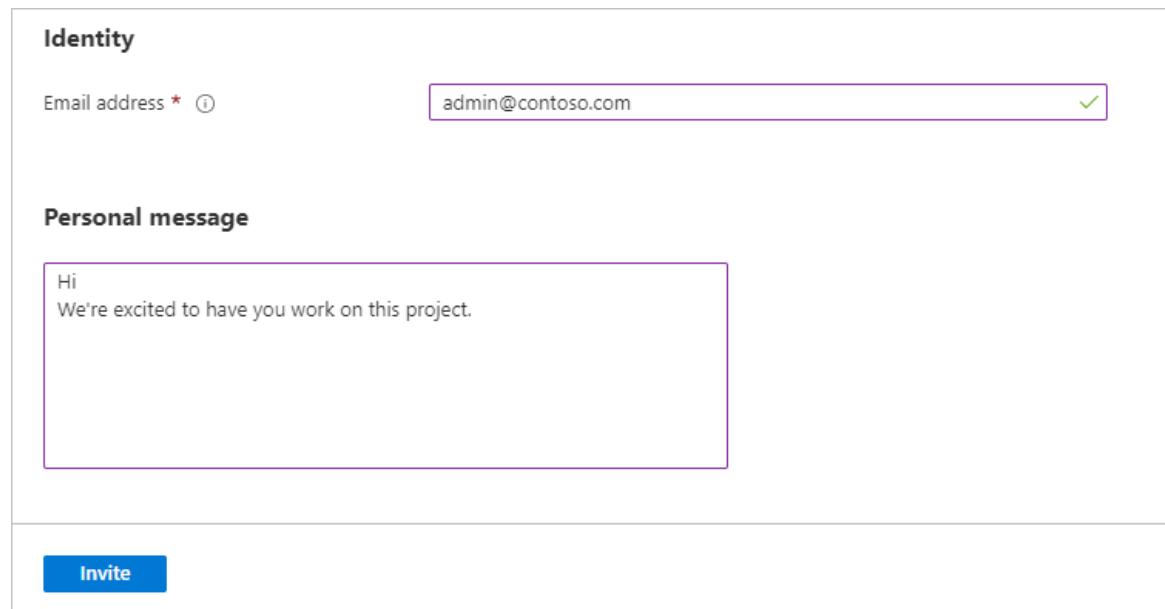
In this approach, Contoso invites the guest users to its Azure AD ahead of time and then distributes Power BI content to them. Contoso can invite guest users from the Azure portal or using PowerShell. Here are the steps to invite guest users from the Azure portal:

- Contoso's Azure AD administrator navigates to **Azure portal > Azure Active Directory > Users > All users > New guest user**



The screenshot shows the 'Users | All users (Preview)' page in the Azure portal. The top navigation bar includes 'MVP Premium Tenant - Azure Active Directory'. Below the navigation, there are several buttons: '+ New user', '+ New guest user' (which is highlighted with a red box), 'Bulk operations', and 'Refresh'. A note on the page says, 'This page includes previews available for your evaluation. View previews →'. On the left, a sidebar lists 'All users (Preview)', 'Deleted users (Preview)', 'Password reset', 'User settings', and 'Diagnose and solve problems'. On the right, there is a search bar 'Search users' and a link 'Add filters'. Below the search bar, it says '11,561 users found'. At the bottom, there are filters for 'Name', 'User principal na...', 'User type', and sorting options.

- Add an invitation message for the guest users and click **Invite**



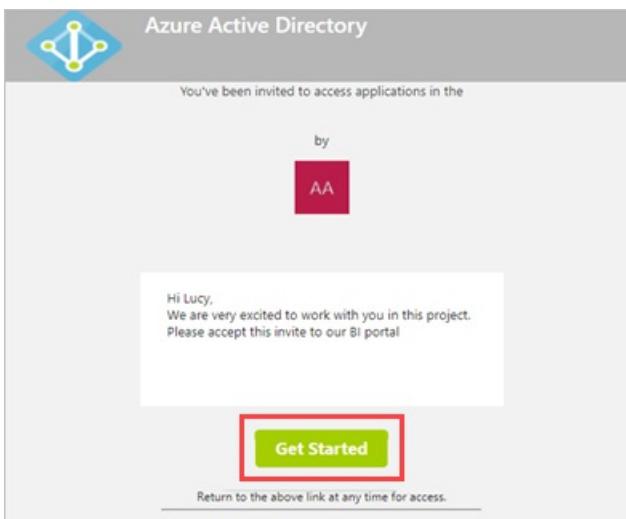
The screenshot shows the 'Identity' section of the 'New guest user' form. It has a field for 'Email address *' containing 'admin@contoso.com' with a green checkmark. Below this is the 'Personal message' section, which contains a text area with the message: 'Hi
We're excited to have you work on this project.' At the bottom is a blue 'Invite' button.

NOTE

To invite guest users from the Azure portal, you need to be an administrator for the Azure Active Directory of your tenant.

If Contoso wants to invite many guest users, they can do so using PowerShell. Contoso's Azure AD administrator stores the email addresses of all the guest users in a CSV file. Here are [Azure Active Directory B2B collaboration code and PowerShell samples](#) and instructions.

After the invitation, guest users receive an email with the invitation link.



Once the guest users click the link, they can access content in the Contoso Azure AD tenant.

NOTE

It is possible to change the layout of the invitation email using the Azure AD branding feature as described [here](#).

Ad hoc Invites

What if Contoso does not know all the guest users it wants to invite ahead of time? Or, what if the analyst in Contoso who created the BI portal wants to distribute content to guest users herself? We also support this scenario in Power BI with ad-hoc invites.

The analyst can just add the external users to the access list of the app when they are publishing it. The guest users gets an invite and once they accept it, they are automatically redirected to the Power BI content.

A screenshot of the Power BI Portal App settings. The left sidebar shows navigation options like Home, Favorites, Recent, Create, Datasets, Goals, Apps, Shared with me, Deployment pipelines, Learn, Workspaces, Power BI Portal, and Get data. The main area is titled "Power BI Portal App" with tabs for Branding, Navigation, Control, Parameters, Authentication, and Access (which is selected). It says "App access" and "Allow others in your organization to install and use your app. Please note this applies only to test phase." There are two radio buttons: "Entire organization" and "Specific individuals or group" (which is selected). Below is a text input field "Enter email addresses". At the bottom are buttons for "Create app" (yellow), "Save changes", and "Close".

NOTE

Invites are needed only the first time an external user is invited to your organization.

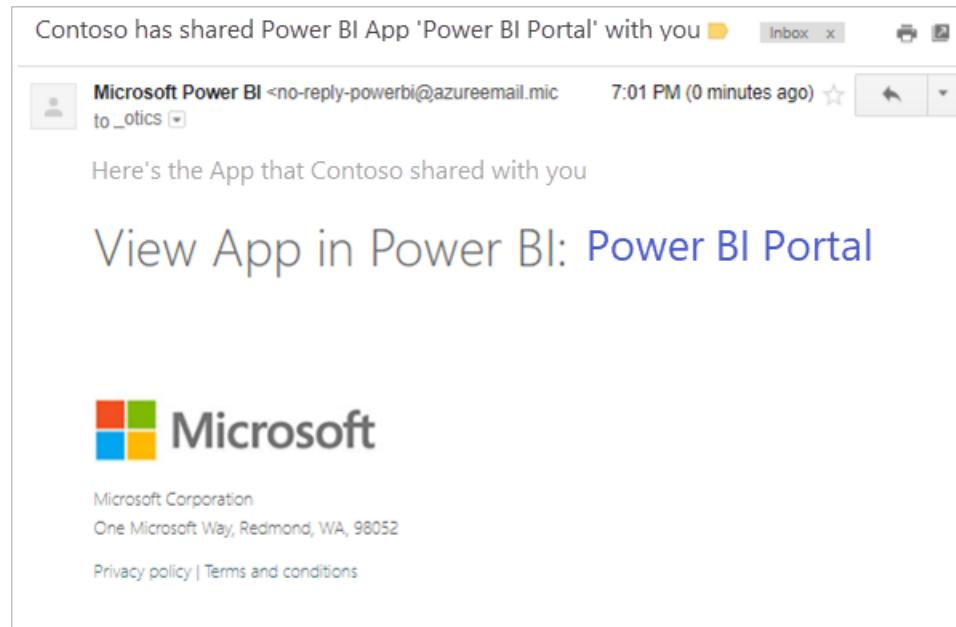
3. Distribute Content

Now that Contoso's BI team has created the BI portal and invited guest users, they can distribute their portal to their end users by giving guest users access to the app and publishing it. Power BI auto-completes names of guest users who have been previously added to the Contoso tenant. Adhoc invitations to other guest users can also be added at this point.

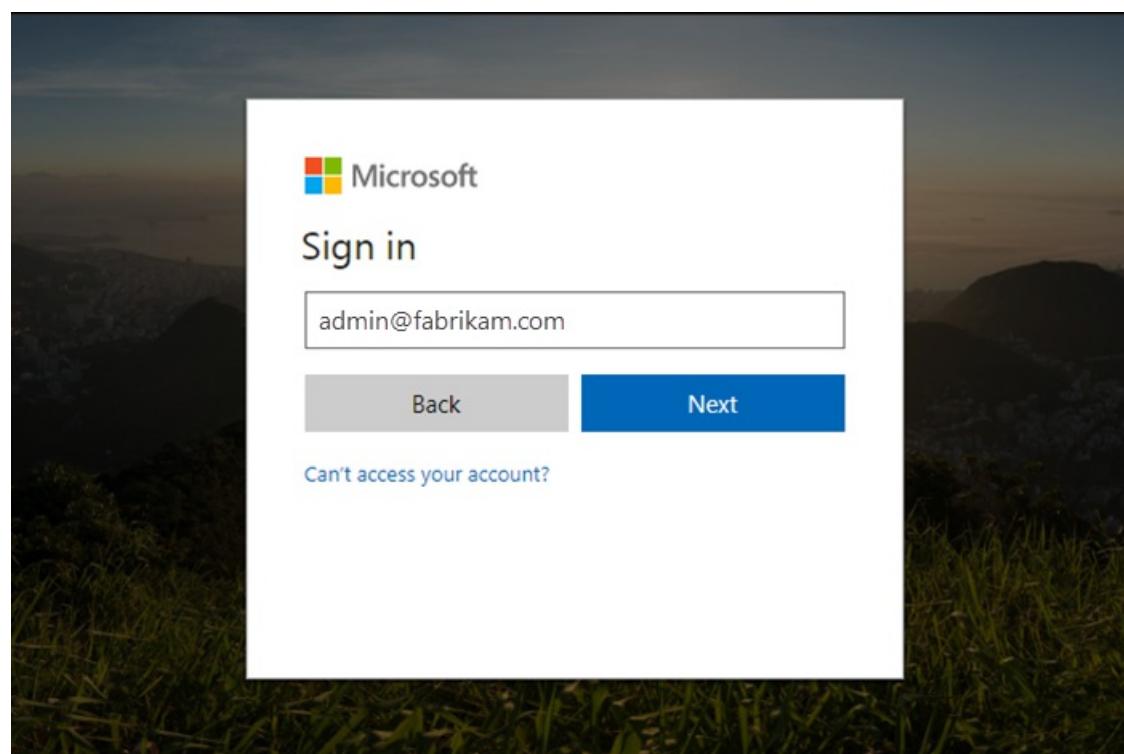
NOTE

If using Security groups to manage access to the app for external users, use the Planned Invites approach and share the app link directly with each external user who must access it. Otherwise, the external user may not be able to install or view content from within the app.

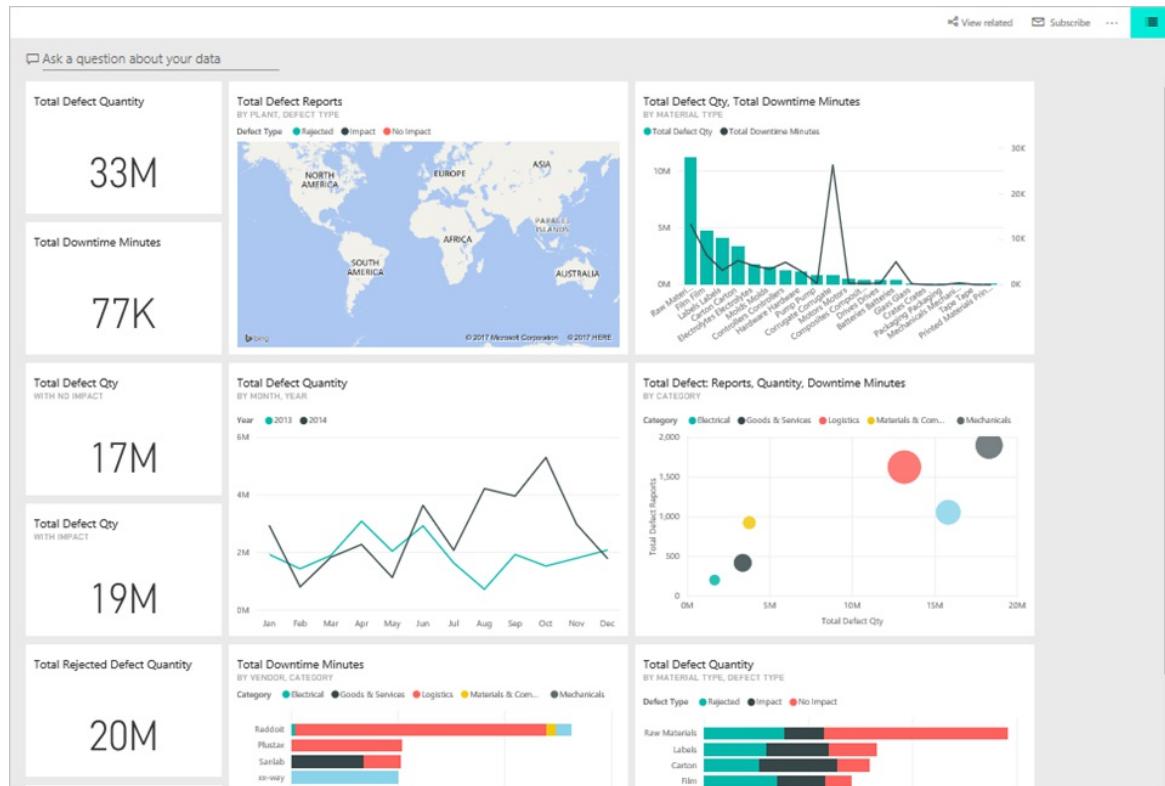
Guest users get an email with a link to the app.



On clicking this link, guest users are asked to authenticate with their own organization's identity.



Once they are successfully authenticated, they are redirected to Contoso's BI app.



Guest users can subsequently get to Contoso's app by clicking the link in the email or bookmarking the link. Contoso can also make it easier for guest users by adding this link to any existing extranet portal that the guest users already use.

4. Next steps

Using a Power BI app and Azure AD B2B, Contoso was able to quickly create a BI Portal for its suppliers in a no-code way. This greatly simplified distributing standardized analytics to all the suppliers who needed it.

While the example showed how a single common report could be distributed among suppliers, Power BI can go much further. To ensure each partner sees only data relevant to themselves, Row Level Security can be added easily to the report and data model. The Data security for external partners section later in this document describes this process in details.

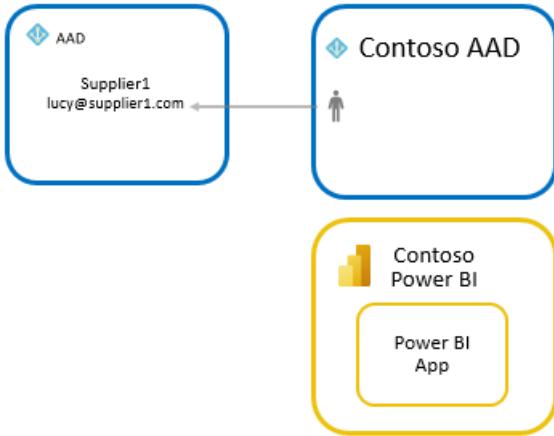
Often individual reports and dashboards need to be embedded into an existing portal. This can also be accomplished reusing many of the techniques shown in the example. However, in those situations it may be easier to embed reports or dashboards directly from a workspace. The process for inviting and assigning security permission to the require users remain the same.

Under the hood: How is Lucy from Supplier1 able to access Power BI content from Contoso's tenant?

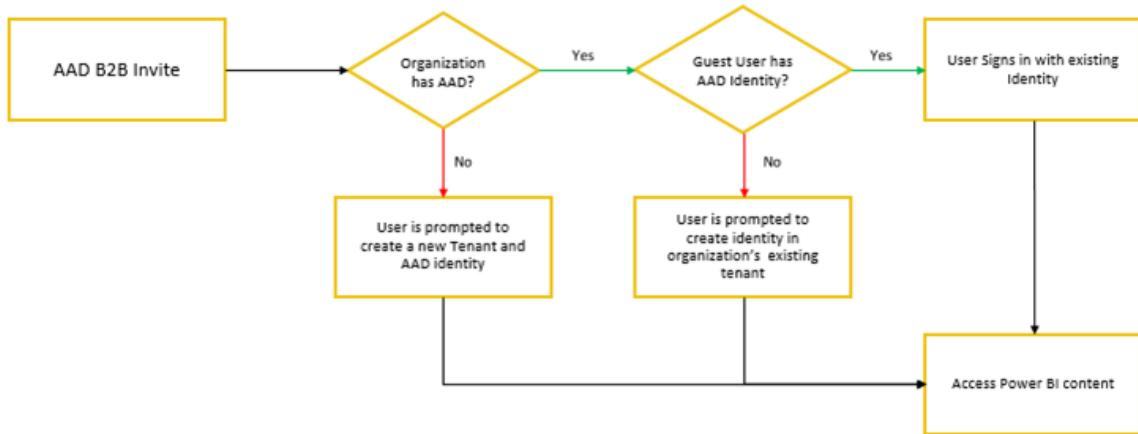
Now that we have seen how Contoso is able to seamlessly distribute Power BI content to guest users in partner organizations, let's look at how this works under the hood.

When Contoso invited lucy@supplier1.com to its directory, Azure AD creates a link between [Lucy@supplier1.com](mailto:lucy@supplier1.com) and the Contoso Azure AD tenant. This link lets Azure AD know that Lucy@supplier1.com can access content in the Contoso tenant.

When Lucy tries to access Contoso's Power BI app, Azure AD verifies that Lucy can access the Contoso tenant and then provides Power BI a token that indicates that Lucy is authenticated to access content in the Contoso tenant. Power BI uses this token to authorize and ensure that Lucy has access to Contoso's Power BI app.



Power BI's integration with Azure AD B2B works with all business email addresses. If the user does not have an Azure AD identity, they may be prompted to create one. The following image shows the detailed flow:



It is important to recognize that the Azure AD account will be used or created in the external party's Azure AD, this will make it possible for Lucy to use their own username and password and their credentials will automatically stop working in other tenants whenever Lucy leaves the company when their organization also uses Azure AD.

Licensing

Contoso can choose one of three approaches to license guest users from its suppliers and partner organizations to have access to Power BI content.

NOTE

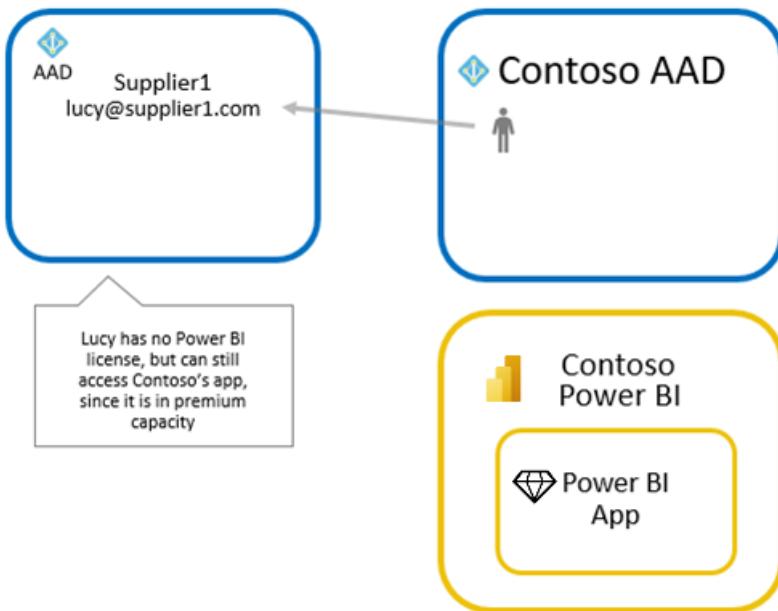
The Azure AD B2B's free tier is enough to use Power BI with Azure AD B2B. Some advanced Azure AD B2B features like dynamic groups require additional licensing. Please refer to the Azure AD B2B documentation for additional information: <https://docs.microsoft.com/azure/active-directory/b2b/licensing-guidance>

Approach 1: Contoso uses Power BI Premium

With this approach, Contoso purchases Power BI Premium capacity and assigns its BI portal content to this capacity. This allows guest users from partner organizations to access Contoso's Power BI app without any Power BI license.

External users are also subject to the consumption only experiences offered to "Free" users in Power BI when consuming content within Power BI Premium.

Contoso can also take advantage of other Power BI premium capabilities for its apps like increased refresh rates, capacity, and large model sizes.

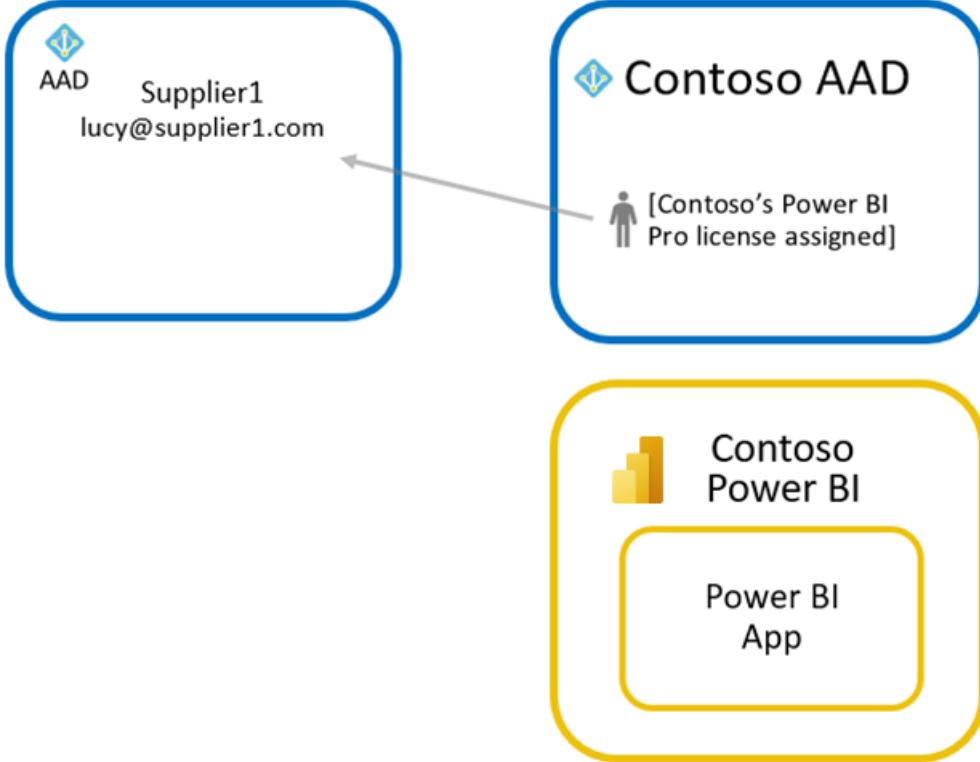


Approach 2: Contoso assigns Power BI Pro licenses to guest users

With this approach, Contoso assigns pro licenses to guest users from partner organizations – this can be done from Contoso's Microsoft 365 admin center. This allows guest users from partner organizations to access Contoso's Power BI app without purchasing a license themselves. This can be appropriate for sharing with external users whose organization has not adopted Power BI yet.

NOTE

Contoso's pro license applies to guest users only when they access content in the Contoso tenant. Pro licenses enable access to content that is not in a Power BI Premium capacity. However, external users with a Pro license are restricted by default to a consumption only experience. This can be changed by using the approach described in the *Enabling external users to edit and manage content within Power BI* section later in this document.

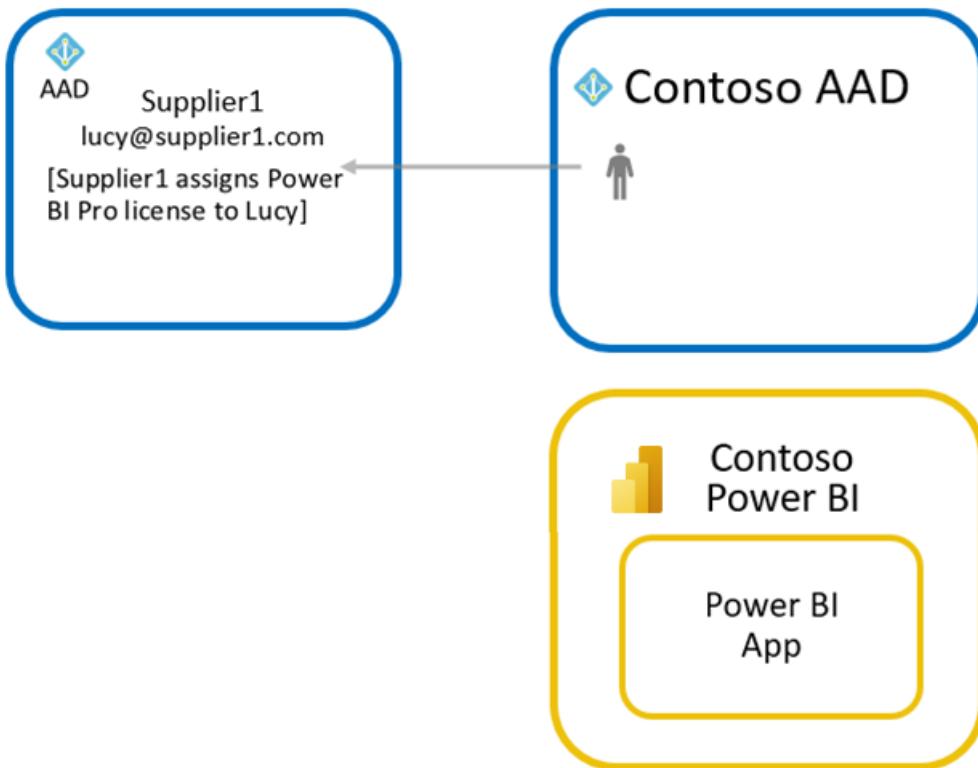


Approach 3: Guest users bring their own Power BI Pro license

With this approach, Supplier 1 assigns a Power BI Pro license to Lucy. They can then access Contoso's Power BI app with this license. Since Lucy can use their Pro license from their own organization when accessing an external Power BI environment, this approach is sometimes referred to as *bring your own license* (BYOL). If both organizations are using Power BI, this offers advantageous licensing for the overall analytics solution and minimizes overhead of assigning licenses to external users.

NOTE

The pro license given to Lucy by Supplier 1 applies to any Power BI tenant where Lucy is a guest user. Pro licenses enable access to content that is not in a Power BI Premium capacity. However, external users with a Pro license are restricted by default to a consumption only experience. This can be change by using the approach described in the *Enabling external users to edit and manage content within Power BI* section later in this document.



Data security for external partners

Commonly when working with multiple external suppliers, Contoso needs to ensure that each supplier sees data only about its own products. User-based security and dynamic row level security make this easy to accomplish with Power BI.

User-based security

One of the most powerful features of Power BI is Row Level Security. This feature allows Contoso to create a single report and dataset but still apply different security rules for each user. For an in-depth explanation, see [Row-level security \(RLS\)](#).

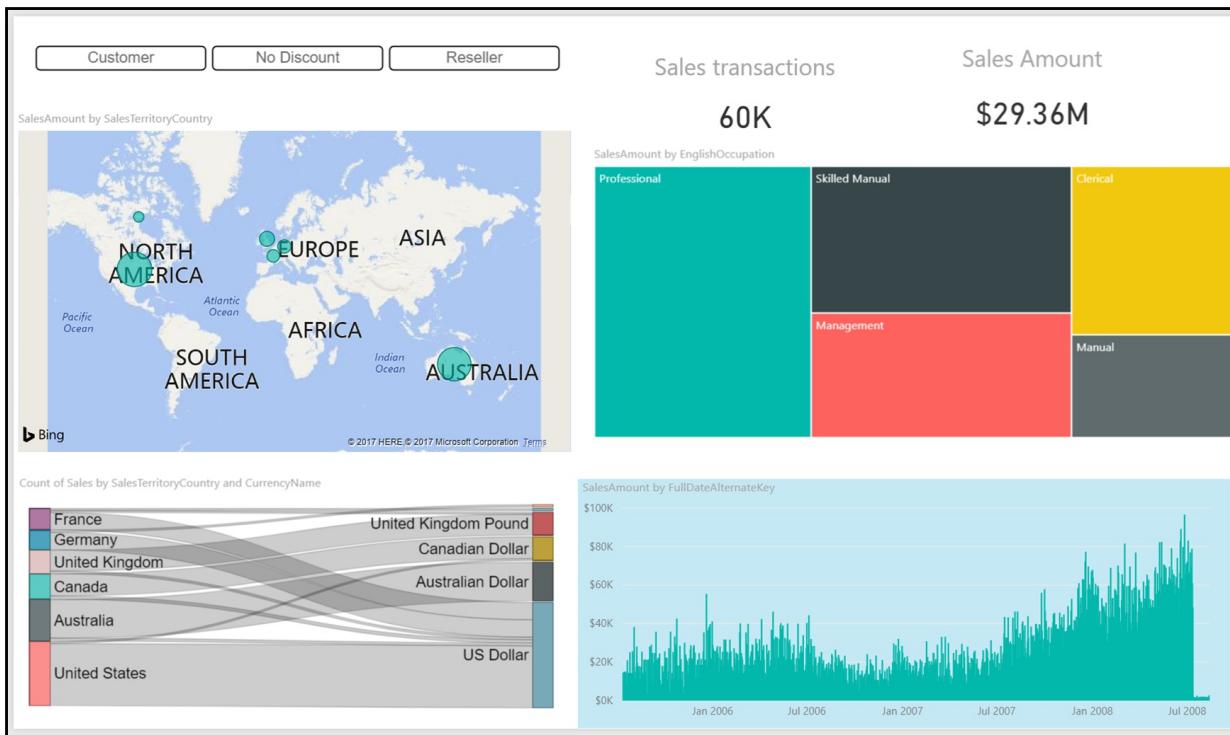
Power BI's integration with Azure AD B2B allows Contoso to assign Row Level Security rules to guest users as soon as they are invited to the Contoso tenant. As we have seen before, Contoso can add guest users through either planned or ad-hoc invites. If Contoso wants to enforce row level security, it is strongly recommended to use planned invites to add the guest users ahead of time and assigning them to the security roles before sharing the content. If Contoso instead uses ad-hoc invites, there might be a short period of time where the guest users will not be able to see any data.

NOTE

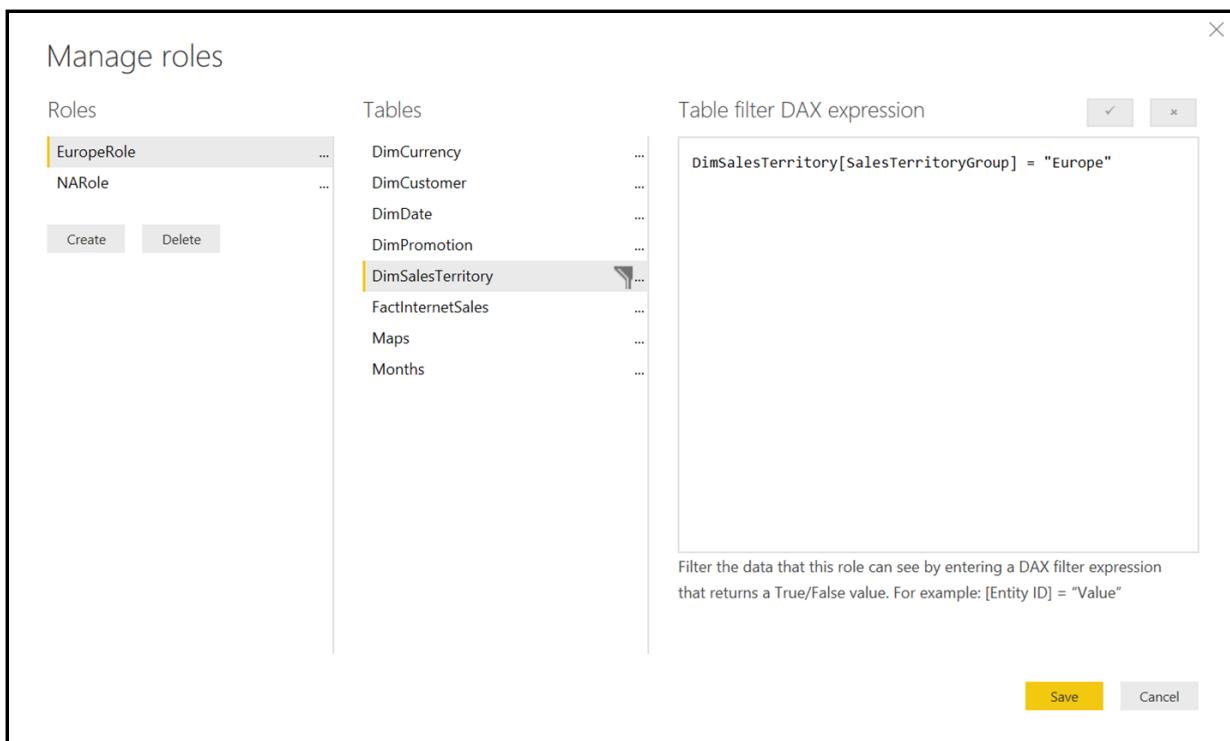
This delay in accessing data protected by RLS when using ad-hoc invites can lead to support requests to your IT team because users will see either blank or broken looking reports/dashboards when opening a sharing link in the email they receive. Therefore, it is strongly recommended to use planned invites in this scenario.**

Let's walk through this with an example.

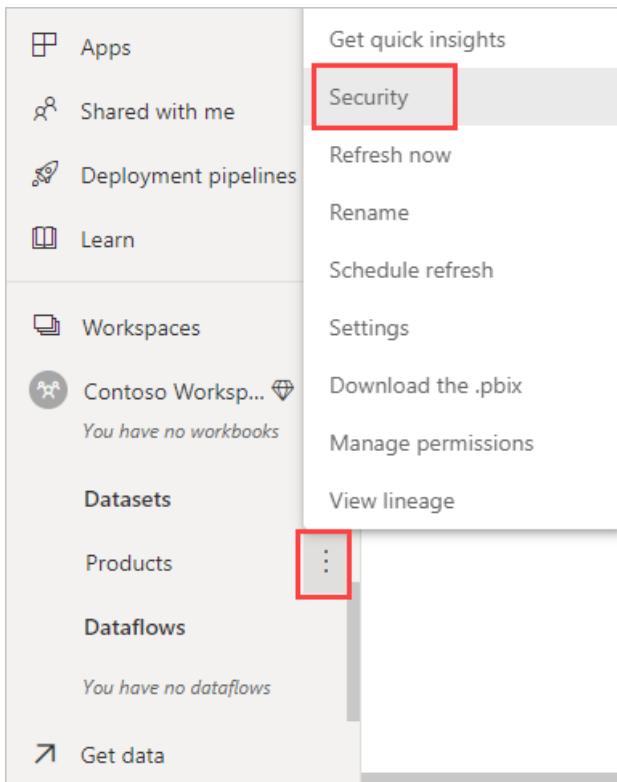
As mentioned before, Contoso has suppliers around the globe, and they want to make sure that the users from their supplier organizations get insights from data from just their territory. But users from Contoso can access all the data. Instead of creating several different reports, Contoso creates a single report and filters the data based the user viewing it.



To make sure Contoso can filter data based on who is connecting, two roles are created in Power BI desktop. One to filter all the data from the SalesTerritory "Europe" and another for "North America".



Whenever roles are defined in the report, a user must be assigned to a specific role for them to get access to any data. The assignment of roles happens inside the Power BI service (**Datasets > Security**)



This opens a page where Contoso's BI team can see the two roles they created. Now Contoso's BI team can assign users to the roles.

The screenshot shows the 'Row-Level Security' page in Power BI. The left sidebar has options like Home, Favorites, Recent, Create, Datasets, and Goals. The main area is titled 'Row-Level Security' and shows a list of roles: 'EuropeRole (0)', 'NARole (0)', and 'New role (0)'. The 'EuropeRole (0)' item is selected and highlighted in grey. To the right, there is a 'Members (0)' section with a sub-section 'People or groups who belong to this role' containing a text input field 'Enter email addresses' and a 'Add' button.

In the example Contoso is adding a user in a partner organization with email address `admin@fabrikam.com` to the Europe role:

The screenshot shows the 'Row-Level Security' page again. The 'EuropeRole (0)' role is selected. In the 'Members (0)' section, the 'People or groups who belong to this role' field contains the email address `admin@fabrikam.com`, which is highlighted with a yellow background. Below the input field is a 'Add' button.

When this gets resolved by Azure AD, Contoso can see the name show up in the window ready to be added:

Row-Level Security

EuropeRole (0)

NARole (0)

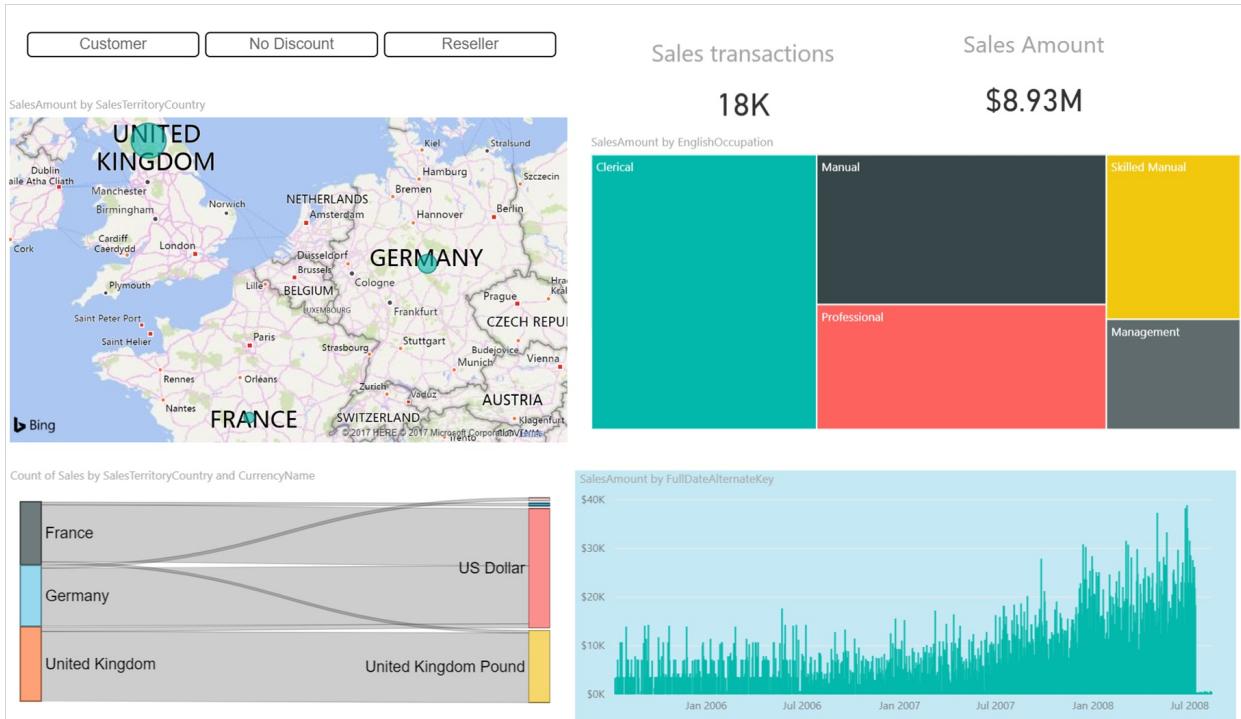
Members (0)

People or groups who belong to this role

admin Enter email addresses

Add

Now when this user opens the app that was shared with them, they only see a report with data from Europe:

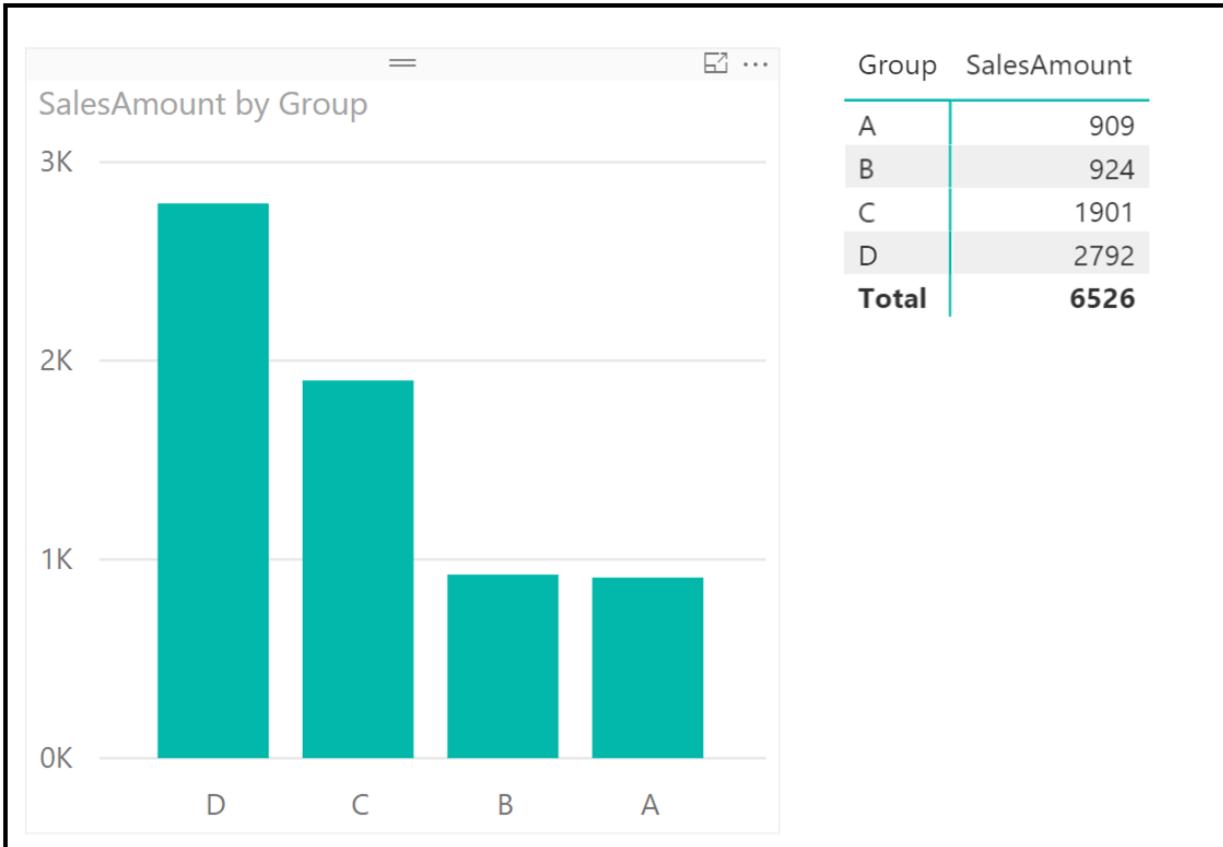


Dynamic row level security

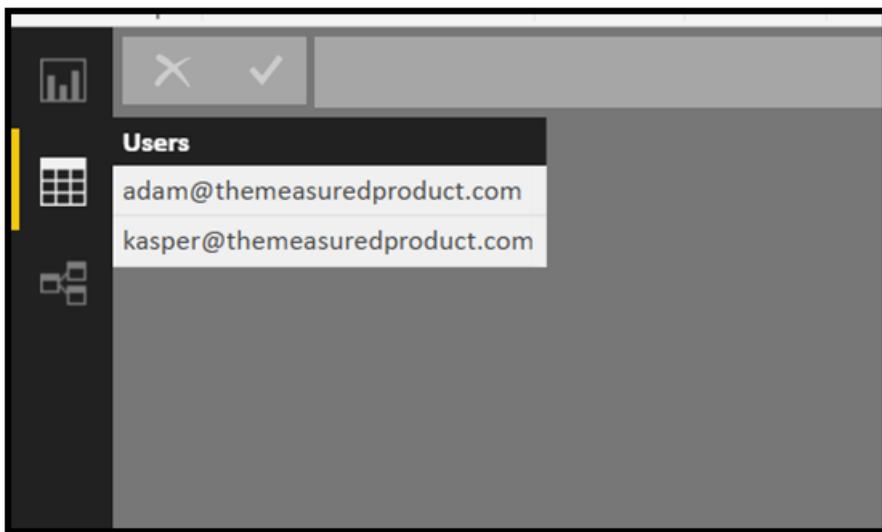
Another interesting topic is to see how dynamic row level security (RLS) work with Azure AD B2B.

In short, Dynamic row level security works by filtering data in the model based on the username of the person connecting to Power BI. Instead of adding multiple roles for groups of users, you define the users in the model. We won't describe the pattern in detail here. Kasper de Jong offers a detailed write up on all the flavors of row level security in [Power BI Desktop Dynamic security cheat sheet](#), and in [this whitepaper](#).

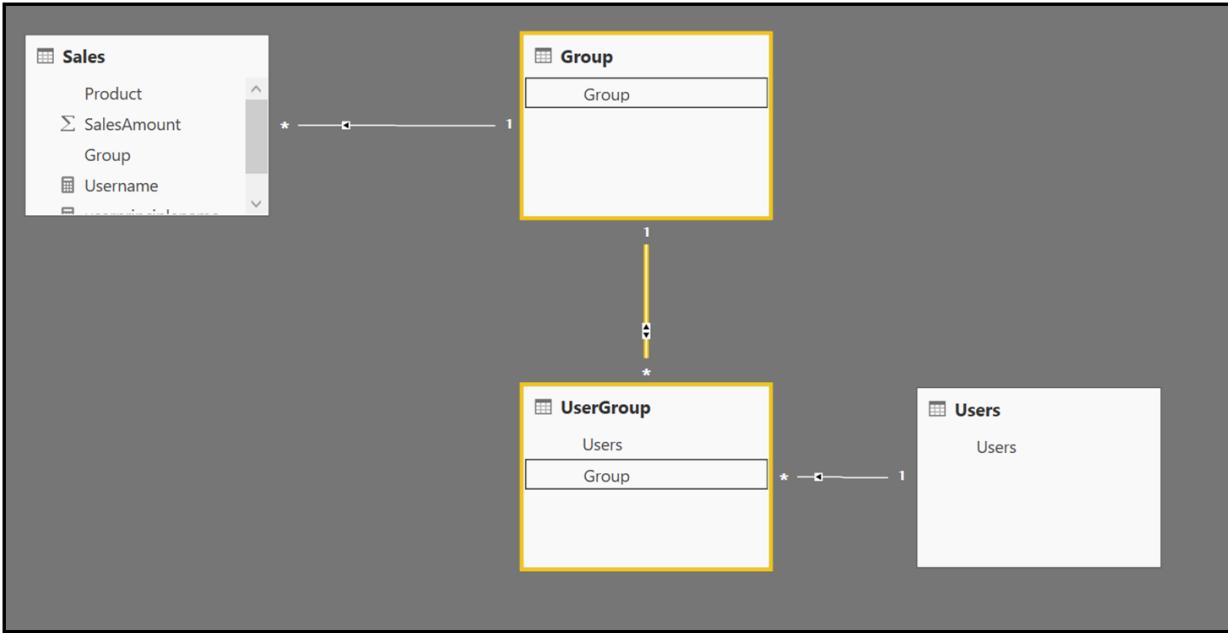
Let's look at a small example - Contoso has a simple report on sales by groups:



Now this report needs to be shared with two guest users and an internal user - the internal user can see everything, but the guest users can only see the groups they have access to. This means we must filter the data only for the guest users. To filter the data appropriately, Contoso uses the Dynamic RLS pattern as described in the whitepaper and blog post. This means, Contoso adds the usernames to the data itself:



Then, Contoso creates the right data model that filters the data appropriately with the right relationships:



To filter the data automatically based on who is logged in, Contoso needs to create a role that passes in the user who is connecting. In this case, Contoso creates two roles – the first is the "securityrole" that filters the Users table with the current username of the user logged in to Power BI (this works even for Azure AD B2B guest users).

Manage roles

Roles	Tables	Table filter DAX expression
AllRole	Group Sales UserGroup Users	Users[Users] = userprincipalname()
SecurityRole		
Create		
Delete		

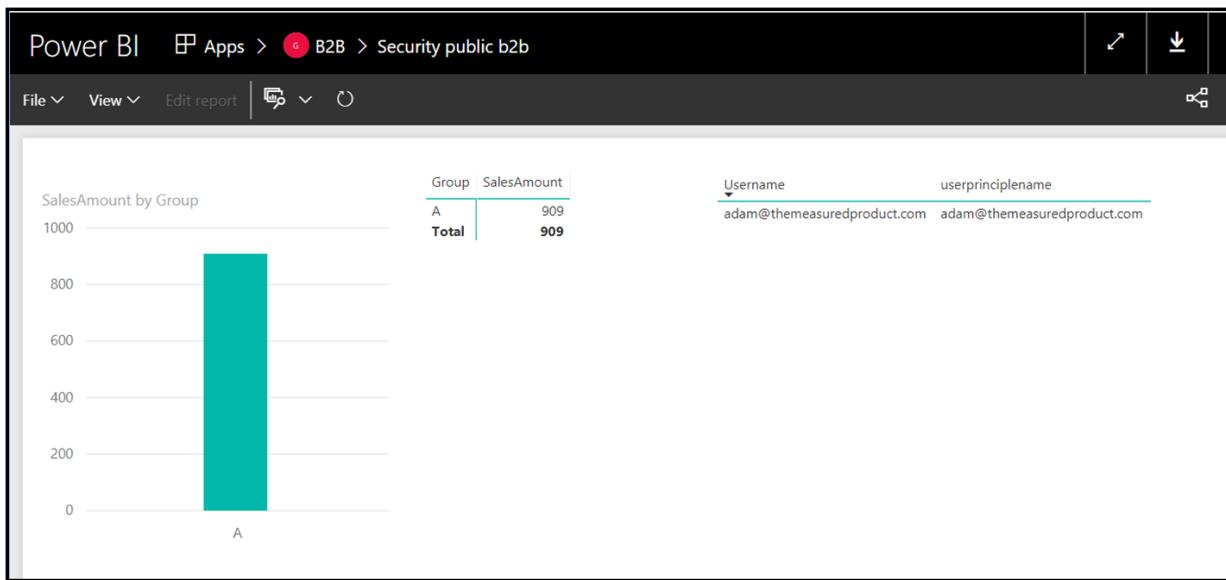
Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

[Save](#) [Cancel](#)

Contoso also creates another "AllRole" for its internal users who can see everything – this role does not have any security predicate.

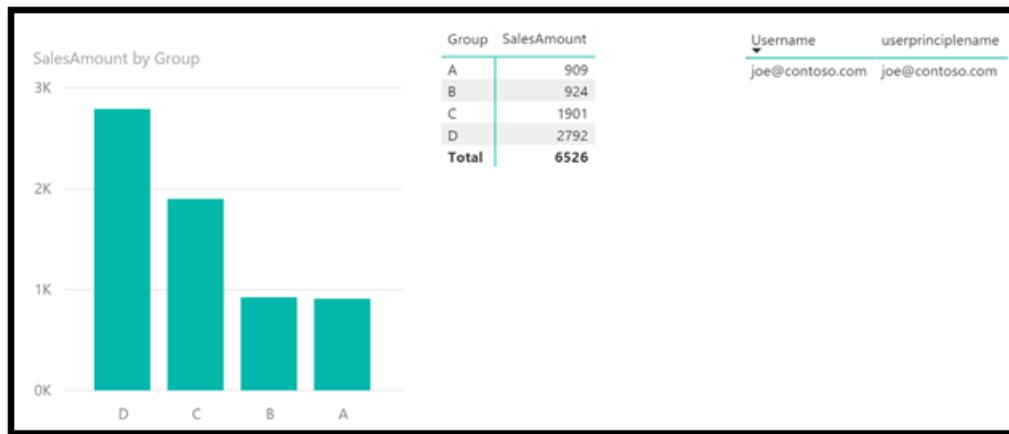
After uploading the Power BI desktop file to the service, Contoso can assign guest users to the "SecurityRole" and internal users to the "AllRole"

Now, when the guest users open the report, they only see sales from group A:



In the matrix to the right you can see the result of the `USERNAME()` and `USERPRINCIPALNAME()` function both return the guest users email address.

Now the internal user gets to see all the data:



As you can see, Dynamic RLS works with both internal or guest users.

NOTE

This scenario also works when using a model in Azure Analysis Services. Usually your Azure Analysis Service is connected to the same Azure AD as your Power BI - in that case, Azure Analysis Services also knows the guest users invited through Azure AD B2B.

Connecting to on premises data sources

Power BI offers the capability for Contoso to leverage on premises data sources like [SQL Server Analysis Services](#) or [SQL Server](#) directly thanks to the [On-Premises data gateway](#). It is even possible to sign on to those data sources with the same credentials as used with Power BI.

NOTE

When installing a gateway to connect to your Power BI tenant, you must use a user created within your tenant. External users cannot install a gateway and connect it to your tenant.

For external users, this might be more complicated as the external users are usually not known to the on-

premises AD. Power BI offers a workaround for this by allowing Contoso administrators to map the external usernames to internal usernames as described in [Manage your data source - Analysis Services](#). For example, lucy@supplier1.com can be mapped to lucy_supplier1_com#EXT@contoso.com.

Map user names

Create rules to map user names to Analysis Services server user names or associate custom data with user names. [Learn more](#)

Select the type of rule for this data source

Effective user names
 CustomData

Replace	With
1 lucy@supplier1.com	lucy_supplier1_com#EXT@contoso.com
2 *	external@contoso.com
3 Original name	New name

Add Delete ▼ ▲

Enter user name to see how the mapping rule will change it.

Original name

lucy@supplier1.com Test rule

After rule applied

lucy_supplier1_com#EXT@contoso.com

OK Cancel

This method is fine if Contoso only has a handful of users or if Contoso can map all the external users to a single internal account. For more complex scenarios where each user needs their own credentials, there is a more advanced approach that uses [custom AD attributes](#) to do the mapping as described in [Manage your data source - Analysis Services](#). This would allow the Contoso administrator to define a mapping for every user in your Azure AD (also external B2B users). These attributes can be set through the AD object model using scripts or code so Contoso can fully automate the mapping on invite or on a scheduled cadence.

Enabling external users to edit and manage content within Power BI

Contoso can allow external users to contribute content within the organization as described earlier in the cross-organization editing and management of Power BI content section.

NOTE

To edit and manage content within your organization's Power BI, the user must have a Power BI Pro license in a workspace other than My workspace. Users can obtain Pro licenses as covered in the *Licensing* section of this document.

The Power BI Admin Portal provides the **allow external guest users to edit and manage content in the organization** setting in Tenant settings. By default, the setting is set to disabled, meaning external users get a constrained read-only experience by default. The setting applies to users with UserType set to Guest in Azure AD. The table below describes the behaviors users experience depending on their UserType and how the settings are configured.

USER TYPE IN AZURE AD	ALLOW EXTERNAL GUEST USERS TO EDIT AND MANAGE CONTENT SETTING	BEHAVIOR
Guest	Disabled for the user (Default)	Per item consumption only view. Allows read-only access to reports, dashboards, and apps when viewed through a URL sent to the Guest user. Power BI Mobile apps provide a read-only view to the guest user.
Guest	Enabled for the user	The external user gets access to the full Power BI experience, though some features are not available to them. The external user must log in to Power BI using the Power BI Service URL with the tenant information included. The user gets the Home experience, a My Workspace, and based on permissions can browse, view, and create content. Power BI Mobile apps provide a read-only view to the guest user.

NOTE

External users in Azure AD can also be set to UserType Member. This is not currently supported in Power BI.

In the Power BI Admin portal, the setting is shown in the following image.

- Allow external guest users to edit and manage content in the organization
Enabled for a subset of the organization

The specified guest users in the organization can edit and manage content in workspaces in the organization. They receive the ability to browse content and request access to content. [Learn more.](#)



Apply to:

- The entire organization
 Specific security groups

High Privilege Guest Users Groups X Enter security groups

Except specific security groups

- i Only guest users who meet the criteria can edit and manage content in the organization

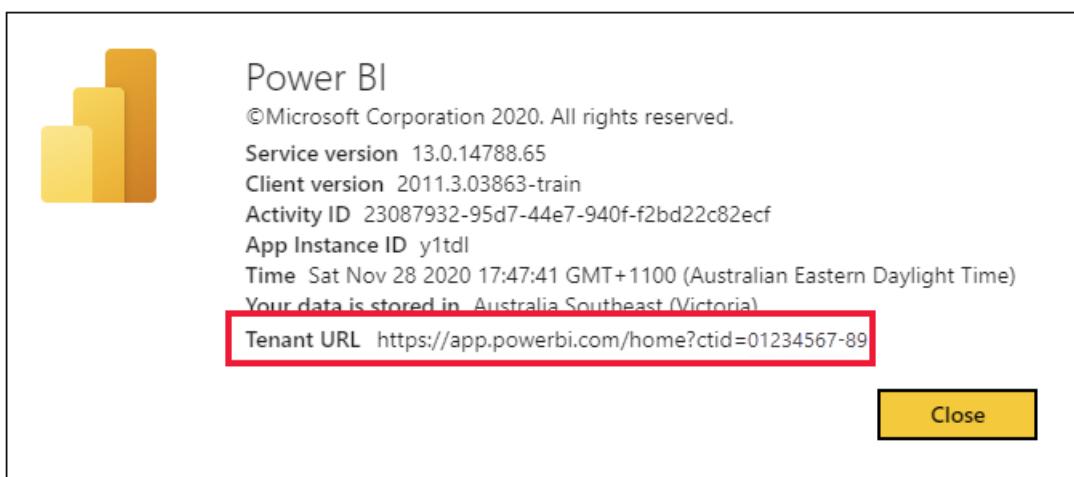
Apply

Cancel

Guest users get the read-only default experience and which can edit and manage content. The default is Disabled, meaning all Guest users have the read-only experience. The Power BI Admin can either enable the setting for all Guest users in the organization or for specific security groups defined in Azure AD. In the following image, the Contoso Power BI Admin created a security group in Azure AD to manage which external users can edit and manage content in the Contoso tenant.

To help these users to log in to Power BI, provide them with the Tenant URL. To find the tenant URL, follow these steps.

1. In the Power BI service, in the top menu, select help (?) then **About Power BI**.
2. Look for the value next to **Tenant URL**. This is the tenant URL you can share with your guest users.



When using the Allow external guest users to edit and manage content in the organization, the specified guest users get access to your organization's Power BI and see any content to which they have permission. They can

access Home, browse and contribute content to workspaces, install apps where they are on the access list, and have a My workspace. They can create or be an Admin of workspaces that use the new workspace experience.

NOTE

When using this option make sure to review the governance section of this document since default Azure AD settings prevent Guest users to use certain features like people pickers which can lead to a reduced experience.**

For guest users enabled through the Allow external guest users to edit and manage content in the organization tenant setting, some experiences are not available to them. To update or publish reports, guest users need to use the Power BI service web UI, including Get Data to upload Power BI Desktop files. The following experiences are not supported:

- Direct publishing from Power BI desktop to the Power BI service
- Guest users cannot use Power BI desktop to connect to service datasets in the Power BI service
- Classic workspaces tied to Microsoft 365 Groups: Guest user cannot create or be Admins of these workspaces. They can be members.
- Sending ad-hoc invites is not supported for workspace access lists
- Power BI Publisher for Excel is not supported for guest users
- Guest users cannot install a Power BI Gateway and connect it to your organization
- Guest users cannot install apps publish to the entire organization
- Guest users cannot use, create, update, or install template apps
- Guest users cannot use Analyze in Excel
- Guest users cannot be @mentioned in commenting (this functionality will be added in an upcoming release)
- Guest users cannot use subscriptions (this functionality will be added in an upcoming release)
- Guest users who use this capability should have a work or school account. Guest users using Personal accounts experience more limitations due to sign-in restrictions.

Governance

Additional Azure AD Settings that affect experiences in Power BI related to Azure AD B2B

When using Azure AD B2B sharing, the Azure Active Directory administrator controls aspects of the external user's experience. These are controlled on the External collaboration settings page within the Azure Active Directory settings for your Tenant.

Details on the settings are available here:

<https://docs.microsoft.com/azure/active-directory/b2b/delegate-invitations>

NOTE

By default, the Guest users permissions are limited option is set to Yes, so Guest users within Power BI have limited experiences especially surround sharing where people picker UIs do not work for those users. It is important to work with your Azure AD administrator to set it to No, as shown below to ensure a good experience.**

External collaboration settings

 Save  Discard

Guest users permissions are limited 

 Yes  No

Admins and users in the guest inviter role can invite 

 Yes  No

Members can invite 

 Yes  No

Guests can invite 

 Yes  No

Collaboration restrictions

- Allow invitations to be sent to any domain (most inclusive)
- Deny invitations to the specified domains
- Allow invitations only to the specified domains (most restrictive)

Control guest invites

Power BI administrators can control external sharing just for Power BI by visiting the Power BI admin portal. But admins can also control external sharing with various Azure AD policies. These policies allow admins to:

- Turn off invitations by end users
- Only admins and users in the Guest Inviter role can invite
- Admins, the Guest Inviter role, and members can invite
- All users, including guests, can invite

You can read more about these policies in [Delegate invitations for Azure Active Directory B2B collaboration](#).

All Power BI actions by external users are also [audited in our auditing portal](#).

Conditional Access policies for guest users

Contoso can enforce conditional access policies for guest users who access content from the Contoso tenant.

You can find detailed instructions in [Conditional access for B2B collaboration users](#).

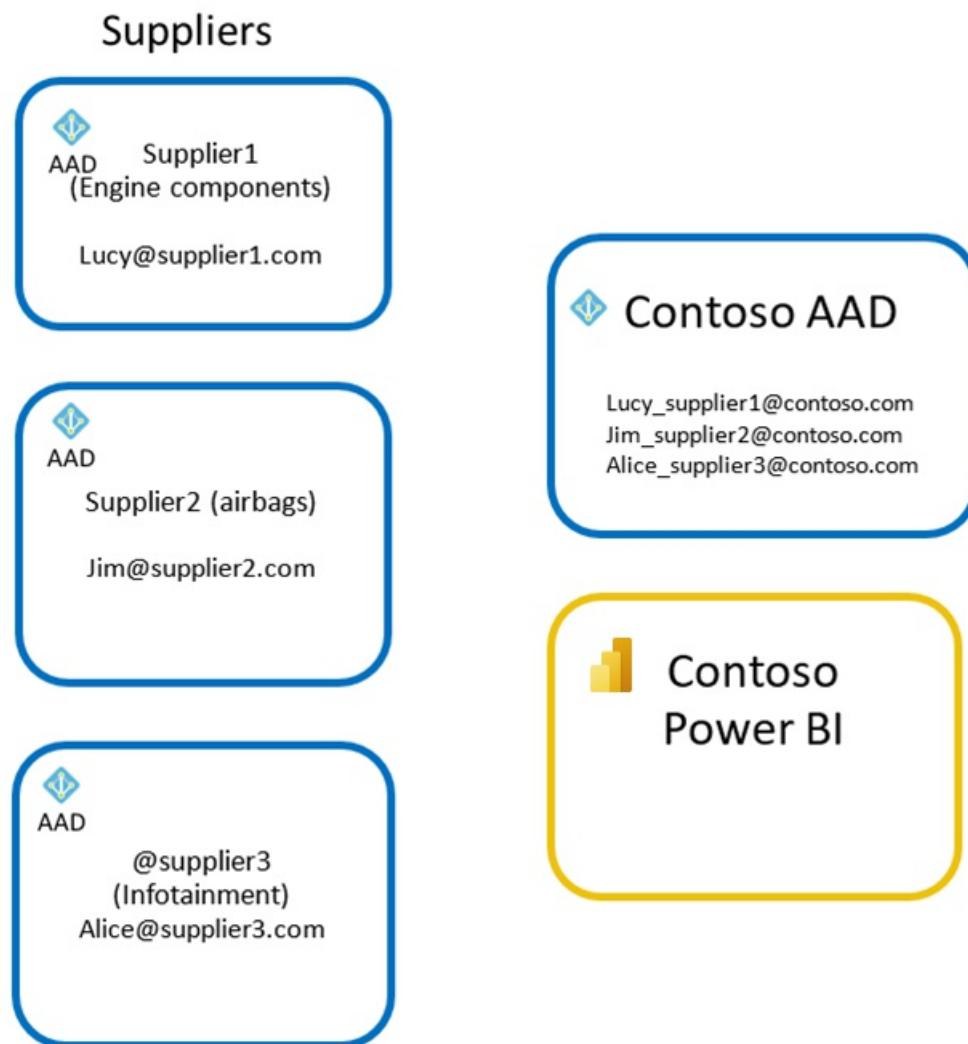
Common alternative approaches

While Azure AD B2B makes it easy to share data and reports across organizations, there are several other approaches that are commonly used and may be superior in certain cases.

Alternative Option 1: Create duplicate identities for partner users

With this option, Contoso had to manually create duplicate identities for each partner user in the Contoso

Tenant, as shown in the following image. Then within Power BI, Contoso can share to the assigned identities the appropriate reports, dashboards, or apps.



Reasons to choose this alternative:

- Since the user's identity is controlled by your organization, any related service such as email, SharePoint, etc. are also within the control of your organization. Your IT Administrators can reset passwords, disable access to accounts, or audit activities in these services.
- Users who use personal accounts for their business often are restricted from accessing certain services so may need an organizational account.
- Some services only work over your organization's users. For example, using Intune to manage content on the personal/mobile devices of external users using Azure B2B may not be possible.

Reasons not to choose this alternative:

- Users from partner organizations must remember two sets of credentials— one to access content from their own organization and the other to access content from Contoso. This is a hassle for these guest users and many guest users are confused by this experience.
- Contoso must purchase and assign per-user licenses to these users. If a user needs to receive email or use office applications, they need the appropriate licenses, including Power BI Pro to edit and share content in Power BI.
- Contoso might want to enforce more stringent authorization and governance policies for external users compared to internal users. To achieve this, Contoso needs to create an in-house nomenclature for external users and all Contoso users need to be educated about this nomenclature.
- When the user leaves their organization, they continue to have access to Contoso's resources until the

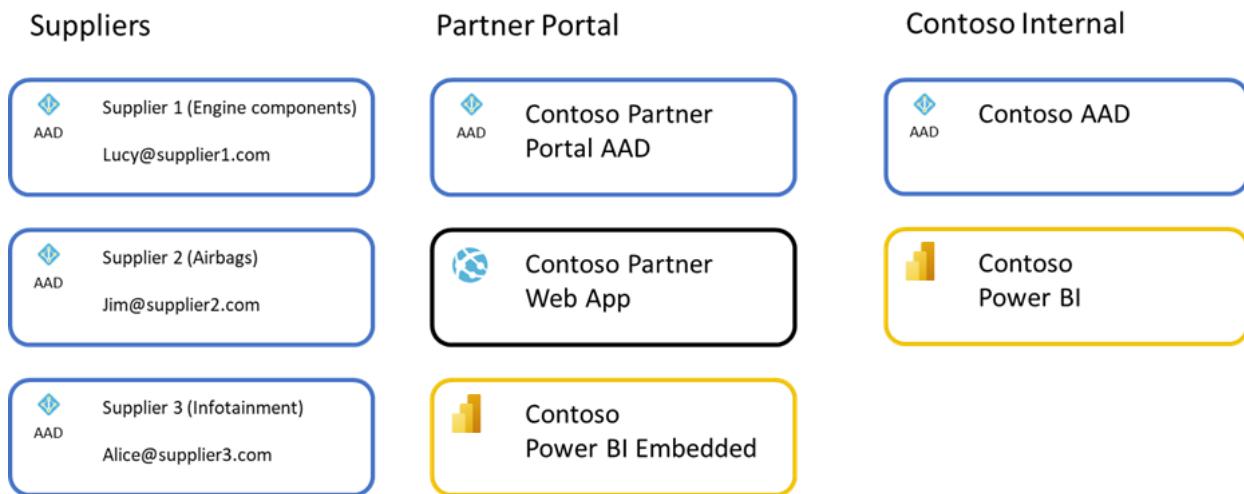
Contoso admin manually deletes their account

- Contoso admins have to manage the identity for the guest, including creation, password resets, etc.

Alternative Option 2: Create a custom Power BI Embedded application using custom authentication

Another option for Contoso is to build its own custom embedded Power BI application with custom authentication ('[App owns data](#)'). While many organizations do not have the time or resources to create a custom application to distribute Power BI content to their external partners, for some organizations this is the best approach and deserves serious consideration.

Often, organizations have existing partner portals that centralize access to all organizational resources for partners, provide isolation from internal organizational resources, and provide streamlined experiences for partners to support many partners and their individual users.



In the example above, users from each supplier login to Contoso's Partner Portal that uses AAD as an identity provider. It could use AAD B2B, Azure B2C, native identities, or federate with any number of other identity providers. The user would log in and access a partner portal build using Azure Web App or a similar infrastructure.

Within the web app, Power BI reports are embedded from a Power BI Embedded deployment. The web app would streamline access to the reports and any related services in a cohesive experience aimed to make it easy for suppliers to interact with Contoso. This portal environment would be isolated from the Contoso internal AAD and Contoso's internal Power BI environment to ensure suppliers could not access those resources. Typically, data would be stored in a separate Partner data warehouse to ensure isolation of data as well. This isolation has benefits since it limits the number of external users with direct access to your organization's data, limiting what data could potentially be available to the external user, and limiting accidental sharing with external users.

Using Power BI Embedded, the portal can leverage advantageous licensing, using app token or the master user plus premium capacity purchased in Azure model, which simplifies concerns about assigning licenses to end users, and can scale up/down based on expected usage. The portal can offer an overall higher quality and consistent experience since partners access a single portal designed with all of a Partner's needs in mind. Lastly, since Power BI Embedded based solutions are typically designed to be multi-tenant, it makes it easier to ensure isolation between partner organizations.

Reasons to choose this alternative:

- Easier to manage as the number of partner organizations grows. Since partners are added to a separate directory isolated from Contoso's internal AAD directory, it simplifies IT's governance duties and helps prevent accidental sharing of internal data to external users.
- Typical Partner Portals are highly branded experiences with consistent experiences across partners and streamlined to meet the needs of typical partners. Contoso can therefore offer a better overall experience to partners by integrating all required services into a single portal.

- Licensing costs for advanced scenarios like Editing content within the Power BI Embedded is covered by the Azure purchased Power BI Premium, and does not require assignment of Power BI Pro licenses to those users.
- Provides better isolation across partners if architected as a multi-tenant solution.
- The Partner Portal often includes other tools for partner beyond Power BI reports, dashboards, and apps.

Reasons not to choose this alternative:

- Significant effort is required to build, operate, and maintain such a portal making it a significant investment in resources and time.
- Time to solution is much longer than using B2B sharing since careful planning and execution across multiple workstreams is required.
- Where there are a smaller number of partners the effort required for this alternative is likely too high to justify.
- Collaboration with ad-hoc sharing is the primary scenario faced by your organization.
- The reports and dashboards are different for each partner. This alternative introduces management overhead beyond just sharing directly with Partners.

FAQ

Can Contoso send an invitation that is automatically redeemed, so that the user is just "ready to go"? Or does the user always have to click through to the redemption URL?

The end user must always click through the consent experience before they can access content.

If you will be inviting many guest users, we recommend that you delegate this from your core Azure AD admins by [adding a user to the guest inviter role in the resource organization](#). This user can invite other users in the partner organization by using the sign-in UI, PowerShell scripts, or APIs. This reduces the administrative burden on your Azure AD admins to invite or resend invites to users at the partner organization.

Can Contoso force multi-factor authentication for guest users if its partners don't have multi-factor authentication?

Yes. For more information, see [Conditional access for B2B collaboration users](#).

How does B2B collaboration work when the invited partner is using federation to add their own on-premises authentication?

If the partner has an Azure AD tenant that is federated to the on-premises authentication infrastructure, on-premises single sign-on (SSO) is automatically achieved. If the partner doesn't have an Azure AD tenant, an Azure AD account may be created for new users.

Can I invite guest users with consumer email accounts?

Inviting guest users with consumer email accounts is supported in Power BI. This includes domains such as hotmail.com, outlook.com and gmail.com. However, those users may experience limitations beyond what users with work or school accounts encounter.