

Connect to data in Power BI - documentation

Power BI documentation provides expert information for connecting to data with tools such as gateways, template apps, and data refresh.

Connect to data in Power BI

CONCEPT

[Data sources](#)

[Data view in Power BI Desktop](#)

TUTORIAL

[Import data from a web page](#)

Gateways

CONCEPT

[On-premises data gateways](#)

TUTORIAL

[Connect to on-premises SQL Server data](#)

HOW-TO GUIDE

[Use personal gateways](#)

Template apps

OVERVIEW

[What are template apps?](#)



CONCEPT

[Create template apps](#)

[Distribute template apps in your org](#)

Refresh data



CONCEPT

[Data refresh](#)

[Incremental refresh for datasets](#)



HOW-TO GUIDE

[Configure scheduled refresh](#)

Data sources for the Power BI service

Article • 07/24/2023

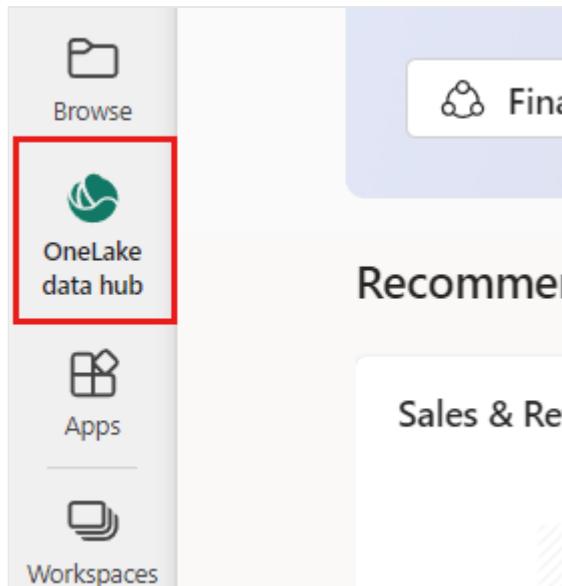
Data is the core of Power BI. You can explore data by creating charts and dashboards or by asking questions with **Q&A**. The visualizations and answers get their underlying data from a dataset, which comes from a data source.

This article focuses on data source types that you can connect to from the Power BI service. There are many other types of data sources. To use these other data sources in the Power BI service, you might first need to use Power BI Desktop or the advanced data query and modeling features in Excel. For more information, see [Databases](#) and [Other data sources](#).

Discover content

You can use the OneLake data hub to discover existing data and reports.

On your Power BI site, select **OneLake data hub** in the navigation pane:



The tiles at the top of the page show recommended data items. For example, data can be recommended because it's promoted by someone in your organization or because it was accessed recently.

OneLake data hub

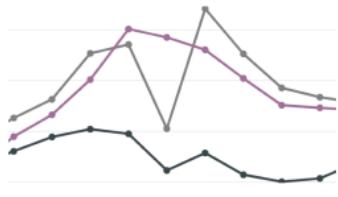
Discover, manage, and use data from across your org, and beyond. [Learn more about OneLake data hub](#)

All domains ▾

Recommended ⓘ

Product MAU, DAU, and NPS ⓘ

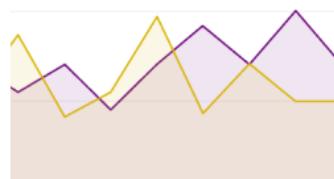
Certified



Details ▾

Marketing insights ⓘ

Promoted



Details ▾

Marketing

Certified



Details ▾

Below those tiles is a list of data that you have access to. You can filter to show all data, your own data, or data endorsed by someone in your organization:

All data	My data	Endorsed in your org	
Explorer	Name ↴	Type	Refreshed
	Sales FY21	Dataset	7m ago
	Marketing DB	Datamart	38m ago
	Client Logs Db	KQL Database	2h ago
	Top Campaigns	Dataset	7h ago
	Dataflow for triggers	Dataset	Yesterday at 11:12 AM
	Daily Sales	Lakehouse	June 18 at 9:02 AM
	Contoso DB	Warehouse	May 23 at 3:00 PM

You can select **Apps** on the navigation pane to discover apps published by other people in your organization. At the top right of that tab, select **Get apps** to choose apps from online services that you use:

The screenshot shows the Microsoft Power BI Apps interface. On the left, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, Apps, and Workspaces. The 'Create' icon is highlighted with a red box. The main area is titled 'Apps' and contains a search bar with 'Search' and a 'Get apps' button. Below the search bar, there are filters for 'View' (dropdown), 'Sort' (dropdown), and a 'Filter by keyword' input field with a placeholder 'Cloud Marketing Insights'. Three app cards are displayed: 'Power BI' (App for the Power BI Engineering Team), 'Cloud Marketing Insights', and 'Documentation Sync for Power BI'. Each card has a star icon for favoriting.

Many services have *template apps* for Power BI. Most services require an account. For more information, see [Connect to services you use with Power BI](#).

Create content

To create content, you can import or create files or databases.

Files

To import files:

1. Go to the workspace to which you want to import the files. Select **New** and then **Dataset**:

The screenshot shows the Microsoft Power BI workspace interface. The left sidebar includes icons for Home, Create (highlighted), Browse, OneLake data hub, Apps, and Workspaces. The main area is titled 'My workspace' and features a 'New' button with a dropdown menu. The 'Dataset' option is selected and highlighted with a red box. Other options in the dropdown include 'Report', 'Paginated report', 'Scorecard', 'Dashboard', and 'Dataset'. To the right, there's a table showing existing datasets: 'Retail Analysis Sample-no-PV' (Type: Workbook), 'PV' (Type: Workbook), and 'Dataset' (Type: Dataset). A search bar at the top right is also visible.

2. Select **Excel** or **CSV**. You can also paste or manually enter data.

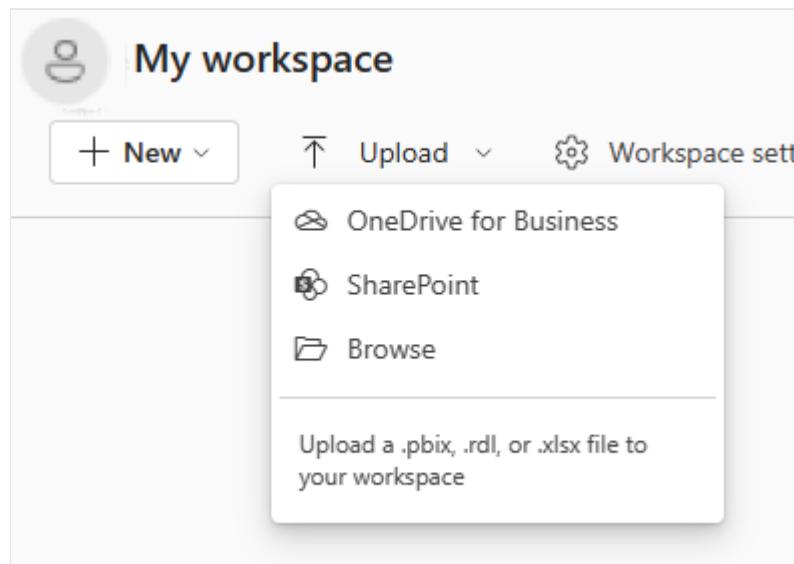
Add data to start building a report

Don't see the source you're looking for? [Download the desktop app.](#)

When you import Excel or CSV files or manually create a workbook, Power BI imports any supported data in tables and any data model into a new Power BI dataset.

You can also upload files. Use this method for .pbix files. When you upload Excel files from OneDrive for Business or SharePoint, Power BI creates a connection to the file. When you upload a local file, Power BI adds a copy of the file to the workspace.

To upload files, on the **My workspace** tab, select **Upload** to upload local files or files from SharePoint or OneDrive for Business:



Following are some types of files that you can add:

- **Excel** workbooks, or .xlsx and .xlsm files, can include different data types. For example, workbooks can include data that you enter into worksheets yourself, or data that you query and load from external data sources by using Power Query. Power Query is available via **Get & Transform Data** on the **Data** tab of Excel, or via **Get External Data** in Power Pivot. You can import data from tables in worksheets or import data from a data model. For more information, see [Get data from files for Power BI](#).

- **Power BI Desktop**, or **.pbix** report files, query and load data from external data sources to create reports. In Power BI Desktop, you can extend your data model by using measures and relationships, and publish the **.pbix** files to the Power BI service. Power BI Desktop is intended for advanced users who have a thorough understanding of their data sources, data querying and transformation, and data modeling. For more information, see [Connect to data in Power BI Desktop](#).
- **Comma-separated value**, or **.csv** files, are simple text files with rows of data that contain values separated by commas. For example, a **.csv** file that contains name and address data might have many rows, each with values for first name, last name, street address, city, and state. You can't import data into a **.csv** file, but many applications, like Excel, can save simple table data as **.csv** files.
- For other file types, like XML (**.xml**) or text (**.txt**), you can use Excel **Get & Transform Data** to query, transform, and load the data first. You can then import the Excel file into the Power BI service.

Where you store your files makes a significant difference. OneDrive provides the greatest flexibility and integration with Power BI. You can also keep your files on your local drive, but if you need to refresh the data, there are a few extra steps. For more information, see [Get data from files for Power BI](#).

Databases

You can connect Azure databases to Power BI to get analytics and reports that provide real-time insights. For example, you can connect to Azure SQL Database and explore data by creating reports in Power BI. Whenever you slice data or add a field to a visualization, Power BI queries the database directly.

For more information, see:

- [Azure and Power BI](#)
- [Azure SQL Database with DirectQuery](#)
- [Azure Synapse Analytics with DirectQuery](#)

You can also use Power BI Desktop or Excel to connect to, query, and load data into data models for a variety of other databases. You can then import the file into Power BI where a dataset exists. If you configure scheduled refresh, Power BI uses the configuration and connection information from the file to connect directly to the data source. Power BI queries for updates and loads the updates into the dataset. For more information, see [Connect to data in Power BI Desktop](#).

Other data sources

You can use hundreds of different data sources with Power BI. The data must be in a format consumable by the Power BI service. Power BI can then use the data to create reports and dashboards and answer questions with Q&A.

Some data sources already have data formatted for the Power BI service. These sources are similar to template apps from service providers like Google Analytics and Twilio. SQL Server Analysis Services tabular model databases are also ready to use.

In other cases, you might need to query and load the data you want into a file. For example, your organization might store logistics data in a data warehouse database on a server. But the Power BI service can connect to that database and explore its data only if it's a tabular model database. You can use Power BI Desktop or Excel to query and load the logistics data into a tabular data model that you then save as a file. You can import that file into Power BI where a dataset exists.

If the logistics data in the database changes every day, you can refresh the Power BI dataset. When you import the data into the dataset, you also import the connection information from Power BI Desktop or the Excel file.

If you configure a scheduled refresh or do a manual refresh on the dataset, Power BI uses the connection information with other settings to connect directly to the database. Power BI then queries for updates and loads those updates into the dataset. You probably need an [on-premises data gateway](#) to help secure any data transfer between an on-premises server and Power BI. When the transfer is complete, visualizations in reports and dashboards refresh automatically.

So even if you can't connect to your data source directly from the Power BI service, you can still get your data into Power BI. It just takes a few more steps and maybe some help from your IT department. For more information, see [Data sources in Power BI Desktop](#).

Datasets and data sources

You might see the terms *dataset* and *data source* used synonymously. But datasets and data sources are two different things, although they're related.

Power BI creates a dataset automatically when you connect to and import data from a file, template app, or live data source. A dataset contains information about the data source and data source credentials. The dataset also often includes a subset of data copied from the data source. When you create visualizations in reports and dashboards, you often look at data from the dataset.

The data in a dataset comes from a data source. For example, data could come from the following data sources:

- An online service like Google Analytics or QuickBooks
- A database in the cloud like Azure SQL Database
- A database or file on a local computer or a server in your organization

Data refresh

If you save your file on a local drive or a drive in your organization, you might need an on-premises gateway to be able to refresh the dataset in Power BI. The computer that stores the file must be running during the refresh. You can also reimport your file, or use **Publish** from Excel or Power BI Desktop, but those processes aren't automated.

If you save your files on OneDrive for work or school or on a SharePoint team site, your dataset, reports, and dashboard are always up to date. Because both OneDrive and Power BI are in the cloud, Power BI can connect directly to your files or import the files into Power BI. Power BI connects about once every hour and checks for updates. The dataset and any visualizations refresh automatically if there are any updates.

Template apps from services also automatically update, once a day in most cases. You can manually refresh these apps, but whether you see updated data depends on the service provider. Updates to template apps from people in your organization depend on the data sources they use and how the app creator configured the refresh.

Azure databases like SQL Database, Azure Synapse Analytics, and Spark in Azure HDInsight are cloud data sources. The Power BI service is also in the cloud, so Power BI can connect to those data sources live by using [DirectQuery](#). With DirectQuery, Power BI is always in sync, and you don't need to set up a scheduled refresh.

SQL Server Analysis Services is a live connection to Power BI just like an Azure cloud database. The difference is that the database is on a server in your organization. This type of connection requires an on-premises gateway, which your IT department can configure.

Data refresh is an important consideration when you use Power BI. For more information, see [Data refresh in Power BI](#).

Considerations and limitations

Data sources for the Power BI service have the following limitations. Other limitations apply to specific features, but the following list applies to the full Power BI service:

- **Dataset size limit.** Datasets stored in shared capacities in the Power BI service have a 1-GB size limit. For larger datasets, use [Power BI Premium](#).
- **Distinct values in a column.** When a Power BI dataset caches data in Import mode, it can store a limit of 1,999,999,997 distinct values in a column.
- **Row limit.** When you use DirectQuery, Power BI imposes a limit on the query results that it sends to your underlying data source. If the query sent to the data source returns more than one million rows, you see an error and the query fails. The underlying data can still contain more than one million rows. You're unlikely to reach this limit, because most reports aggregate the data into smaller sets of results.
- **Column limit.** The maximum number of columns allowed across all tables in a dataset is 16,000 columns. This limit applies to the Power BI service and to datasets Power BI Desktop uses. Power BI uses this limit to track the number of both columns and tables in the dataset, which means the maximum number of columns is 16,000 minus one for each table in the dataset.
- **Data source user limit.** The maximum number of data sources allowed per user is 1,000. This limit applies only to the Power BI service.

Next steps

- [Connect to services you use with Power BI](#)
- [Get data from files for Power BI](#)
- [Data refresh in Power BI](#)
- [DirectQuery in Power BI](#)
- [What is an on-premises data gateway?](#)
- [Data sources in Power BI Desktop](#)

What is an on-premises data gateway?

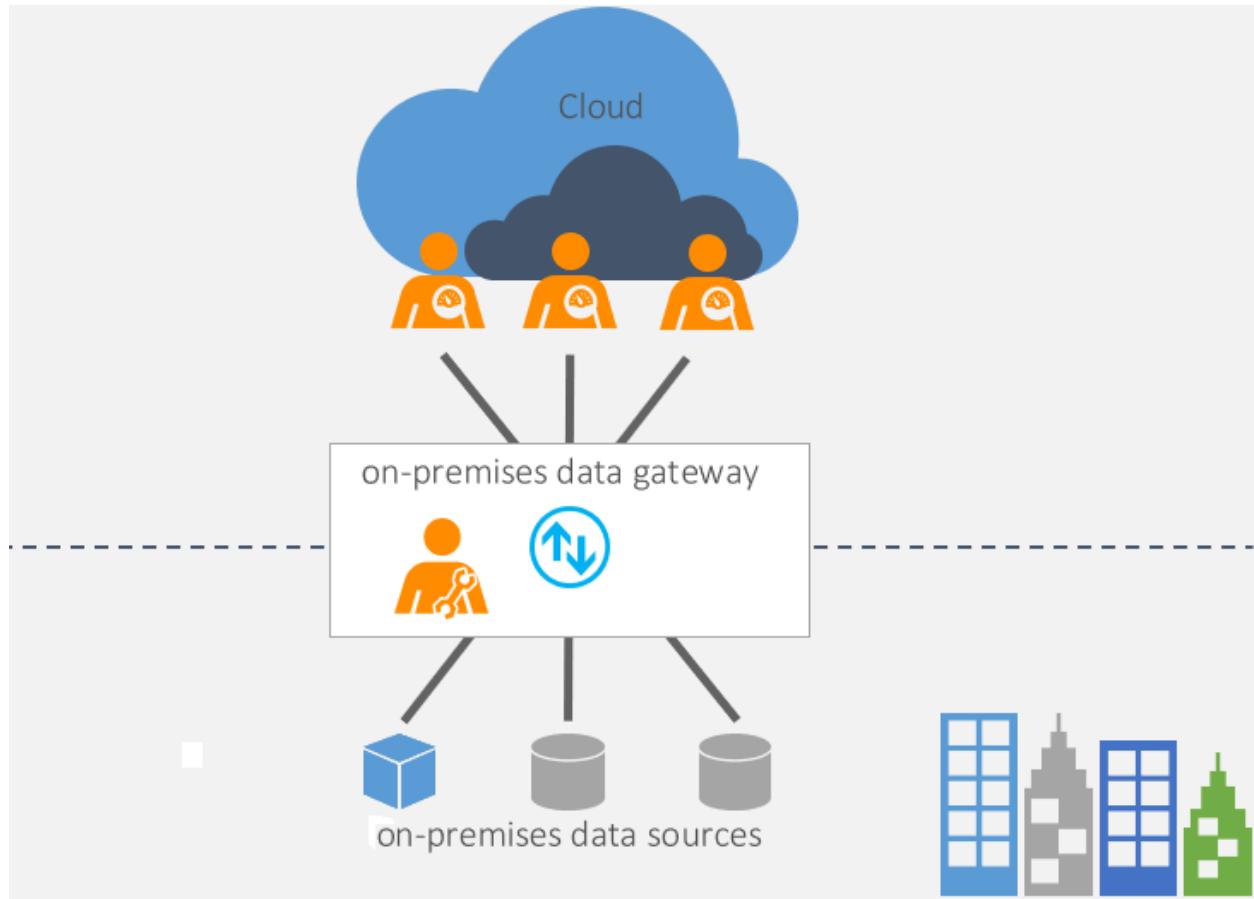
Article • 11/02/2022

ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

The on-premises data gateway acts as a bridge to provide quick and secure data transfer between on-premises data (data that isn't in the cloud) and several Microsoft cloud services. These cloud services include Power BI, PowerApps, Power Automate, Azure Analysis Services, and Azure Logic Apps. By using a gateway, organizations can keep databases and other data sources on their on-premises networks, yet securely use that on-premises data in cloud services.

How the gateway works



For more information on how the gateway works, see [On-premises data gateway architecture](#).

Types of gateways

There are three different types of gateways, each for a different scenario:

- **On-premises data gateway:** Allows multiple users to connect to multiple on-premises data sources. With a single gateway installation, you can use an on-premises data gateway with all supported services. This gateway is well-suited to complex scenarios in which multiple people access multiple data sources.
- **On-premises data gateway (personal mode):** Allows one user to connect to sources and can't be shared with others. An on-premises data gateway (personal mode) can only be used with Power BI. This gateway is well-suited to scenarios in which you're the only person who creates reports, and you don't need to share any data sources with others.
- **Virtual network data gateway:** Allows multiple users to connect to multiple data sources that are secured by virtual networks. No installation is required because it's a Microsoft managed service. This gateway is well-suited to complex scenarios in which multiple people access multiple data sources.

Use a gateway

There are five main steps for using a gateway:

1. [Download and install the gateway](#) on a local computer.
2. [Configure the gateway](#) based on your firewall and other network requirements.
3. [Add gateway admins](#) who can also manage and administer other network requirements.
4. [Use the gateway](#) to refresh an on-premises data source.
5. [Troubleshoot](#) issues with the gateway.

Next steps

- [Install the on-premises data gateway](#)

More questions? [Try the Power BI Community](#) ↗

Discover data items in the OneLake data hub

Article • 09/13/2023

The OneLake data hub makes it easy to find, explore, and use the Fabric data items in your organization that you have access to. It provides information about the items and entry points for working with them.

The data hub provides:

- A filterable list of all the data items you can access
- A gallery of recommended data items
- A way of finding data items by workspace
- A way to display only the data items of a selected domain
- An options menu of things you can do with the data item

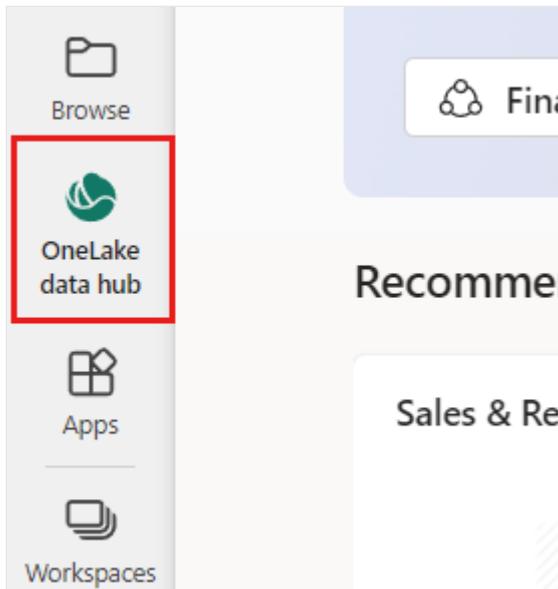
This article explains what you see on the data hub and describes how to use it.

The screenshot shows the Power BI OneLake data hub interface. At the top, there's a search bar and a navigation bar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, and My workspace. The main area has a title "OneLake data hub" with a subtitle "Discover, manage, and use data from across your org, and beyond." Below this is a section titled "Recommended" featuring six cards: "Product MAU, DAU, and NPS" (Certified), "Marketing insights" (Promoted), "Marketing" (Certified), "Top features" (Certified), "Customer feedback" (Certified), and "AppAccessSettings" (Promoted). Below the recommended section is a table titled "Explorer" with columns: Name, Type, Refreshed, Owner, Location, Endorsement, and Sensitivity. The table lists various data items like Sales FY21, Marketing DB, Client Logs Db, etc., with details such as owner names, location in Contoso workspace, and endorsement status. At the bottom right of the table is a magnifying glass icon for filtering.

Name	Type	Refreshed	Owner	Location	Endorsement	Sensitivity
Sales FY21	Dataset	7m ago	Tim Deboar	Contoso workspace	Certified	Highly Confidential\Contoso...
Marketing DB	Datamart	38m ago	Daichi Fukuda	New product insights	Certified	—
Client Logs Db	KQL Database	2h ago	Emiliano Ceballos	Azure data	Promoted	Confidential\Contoso FTE
Top Campaigns	Dataset	7h ago	Mikhail Kotov	Azure data	—	Public
Dataflow for triggers	Dataset	Yesterday at 11:12 AM	Marie Beaudouin	Contoso workspace	—	—
Daily Sales	Lakehouse	June 18 at 9:02 AM	Oscar Krogh	Contoso workspace	Certified	Non-Business
Contoso DB	Warehouse	May 23 at 3:00 PM	Marie Beaudouin	Big data	Promoted	—
Test datamart	Datamart	May 15 at 5:13 AM	Tim Deboar	Events	Certified	Public
Primary dataflow	KQL Database	April 29 at 8:45 PM	Oscar Krogh	Big data	—	—
Contoso DB	Lakehouse	April 11 at 11:56 AM	Ruth Bengtsson	Contoso workspace	Certified	Confidential\Contoso FTE

Open the data hub

To open the data hub, select the OneLake data hub icon in the navigation pane.



ⓘ Note

The OneLake data hub icon and label you see may differ slightly from that shown above, and may also differ slightly from that seen by other users. The data hub functionality is the same, however, no matter which icon/label appears. For more information, see [Considerations and limitations](#).

Find items in the data items list

The data items list displays all the data items you have access to. To shorten the list, you can filter by keyword or data-item type using the filters at the top of the list. If you select the name of an item, you'll get to the item's details page. If you hover over an item, you'll see three dots that open the [options menu](#) when you select them.

All data	My data	Endorsed in your org	Filter by keyword	Filter																																																
<table border="1"><thead><tr><th>Explorer</th><th>Name</th><th>Type</th><th>Refreshed</th><th>Owner</th><th>Location</th><th>Endorsement</th><th>Sensitivity</th></tr></thead><tbody><tr><td>Sales FY21</td><td>Dataset</td><td>7m ago</td><td>Tim Deboar</td><td>Contoso workspace</td><td>Certified</td><td>Highly Confidential</td><td>Contoso (1)</td></tr><tr><td>Marketing DB</td><td>Datamart</td><td>38m ago</td><td>Daichi Fukuda</td><td>New product insights</td><td>Certified</td><td>—</td><td>—</td></tr><tr><td>Client Logs Db</td><td>KQL Database</td><td>2h ago</td><td>Emiliano Ceballos</td><td>Azure data</td><td>Promoted</td><td>Confidential</td><td>Contoso (1)</td></tr><tr><td>Top Campaigns</td><td>Dataset</td><td>7h ago</td><td>Mikhail Kotov</td><td>Azure data</td><td>—</td><td>Public</td><td>(1)</td></tr><tr><td>Dataflow for finance</td><td>Dataset</td><td>Yesterday at 11:12 AM</td><td>Maria Beauregard</td><td>Contoso workspace</td><td>—</td><td>—</td><td>—</td></tr></tbody></table>	Explorer	Name	Type	Refreshed	Owner	Location	Endorsement	Sensitivity	Sales FY21	Dataset	7m ago	Tim Deboar	Contoso workspace	Certified	Highly Confidential	Contoso (1)	Marketing DB	Datamart	38m ago	Daichi Fukuda	New product insights	Certified	—	—	Client Logs Db	KQL Database	2h ago	Emiliano Ceballos	Azure data	Promoted	Confidential	Contoso (1)	Top Campaigns	Dataset	7h ago	Mikhail Kotov	Azure data	—	Public	(1)	Dataflow for finance	Dataset	Yesterday at 11:12 AM	Maria Beauregard	Contoso workspace	—	—	—				
Explorer	Name	Type	Refreshed	Owner	Location	Endorsement	Sensitivity																																													
Sales FY21	Dataset	7m ago	Tim Deboar	Contoso workspace	Certified	Highly Confidential	Contoso (1)																																													
Marketing DB	Datamart	38m ago	Daichi Fukuda	New product insights	Certified	—	—																																													
Client Logs Db	KQL Database	2h ago	Emiliano Ceballos	Azure data	Promoted	Confidential	Contoso (1)																																													
Top Campaigns	Dataset	7h ago	Mikhail Kotov	Azure data	—	Public	(1)																																													
Dataflow for finance	Dataset	Yesterday at 11:12 AM	Maria Beauregard	Contoso workspace	—	—	—																																													

The list has three tabs to narrow down the list of data items.

Tab	Description
All	Data items that you're allowed to find.
My data	Data items that you own.
Endorsed in your org	Endorsed data items in your organization that you're allowed to find. Certified data items are listed first, followed by promoted data items. For more information

Tab	Description
	about endorsement, see the Endorsement overview

The columns of the list are described below.

Column	Description
Name	The data item name. Select the name to open the item's details page.
Endorsement	Endorsement status.
Owner	Data item owner (listed in the <i>All</i> and <i>Endorsed in your org</i> tabs only).
Workspace	The workspace the data item is located in.
Refreshed	Last refresh time (rounded to hour, day, month, and year. See the details section on the item's detail page for the exact time of the last refresh).
Next refresh	The time of the next scheduled refresh (<i>My data</i> tab only).
Sensitivity	Sensitivity, if set. Select the info icon to view the sensitivity label description.

Find items by workspace

Related data items are often grouped together in a workspace. To see the data items by workspace, expand the **Explorer** pane and select the workspace you're interested in. The data items you're allowed to see in that workspace will be displayed in the data items list.

The screenshot shows the Power BI Data Hub interface. At the top, there are three navigation tiles: 'All data' (selected), 'My data', and 'Endorsed in your org'. Below these is the 'Explorer' pane, which includes a search bar and a list of workspaces: 'All workspaces' (selected), 'My Workspace', 'Contoso workspace', 'New product insights', 'RandomFolder', 'ShortcutLink Folder', and 'Azure data design'. To the right is the main content area, which displays a list of data items under the heading 'Name ↓': 'Sales FY21', 'Marketing DB', 'Client Logs Db', 'Top Campaigns', 'Daily Sales', 'Contoso DB', 'Test datamart', 'Primary dataflow', 'Contoso DB', and another 'Primary dataflow'. A red box highlights the back arrow icon in the Explorer pane header.

Name
Sales FY21
Marketing DB
Client Logs Db
Top Campaigns
Daily Sales
Contoso DB
Test datamart
Primary dataflow
Contoso DB
Primary dataflow

⚠ Note

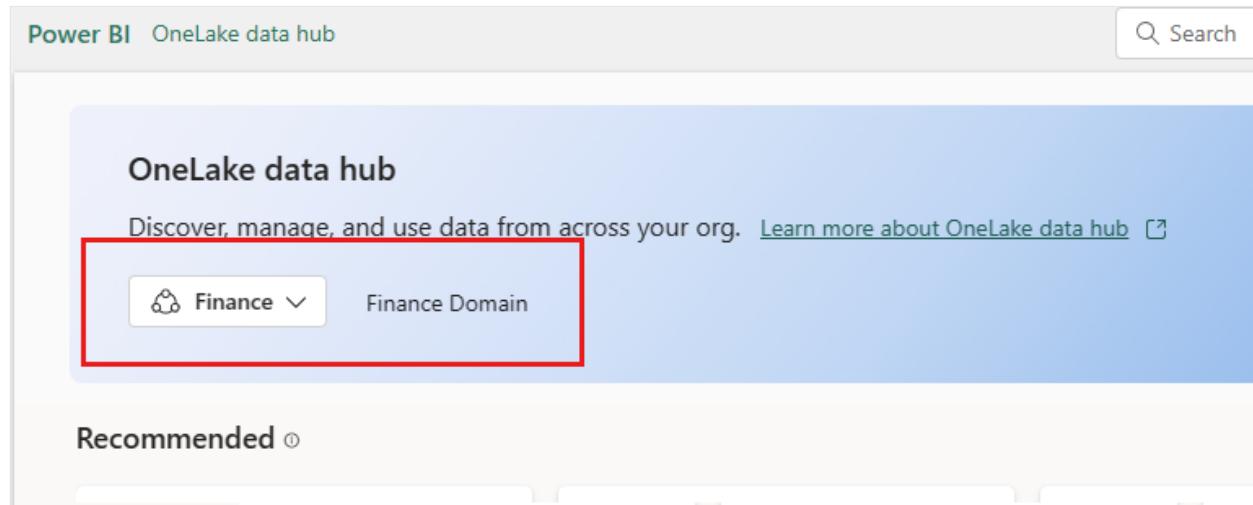
The Explorer pane may list workspaces that you don't have access to if the workspace contains items that you do have access to (through explicitly granted permissions, for example). If you select such a workspace, only the items you have access to will be displayed in the data items list.

Find recommended items

Use the tiles across the top of the data hub to find and explore recommended data items. Recommended data items are data items that have been [certified or promoted](#) by someone in your organization or have recently been refreshed or accessed. Each tile contains information about the item and an [options menu](#) for doing things with the item. When you select a recommended tile, you are taken to the item's details page.

Display only data items belonging to a particular domain

If [domains](#) have been defined in your organization, you can use the domain selector to select a domain so that only data items belonging to that domain will be displayed. If an image has been associated with the domain, you'll see that image on the data hub to remind you of the domain you're viewing.



For more information about domains, see the [Domains overview](#).

Open an item's options menu

Each item shown in the data hub has an options menu that enables you to do things, such as open the item's settings, manage item permissions, etc. The options available depend on the item and your permissions on the item.

To display the options menu, select **More options (...)** on one of the items shown in the data items list or a recommended item. In the data items list, you need to hover over the item to reveal **More options**.

The screenshot shows the Microsoft Fabric Home interface. On the left, the 'Explorer' pane is open, displaying a list of workspaces. One workspace, 'Sales & Returns', is selected and shown in more detail on the right. This detailed view includes a chart titled 'Product MAU, DAU, and NPS' with a 'Certified' badge, the owner's name 'Ina Powers', and a 'Details' button. A red arrow points from the text 'Open the options menu' to the three-dot menu icon at the bottom right of the detailed view.

ⓘ Note

The Explorer pane may list workspaces that you don't have access to if the workspace contains items that you do have access to (through explicitly granted permissions, for example). If you select such a workspace, only the items you have access to will be displayed in the data items list.

Considerations and limitations

The OneLake data hub's icon and label is currently undergoing evaluation, and its appearance may vary slightly for different users. However, data hub functionality is not affected and is the same no matter which icon/label variation appears. The icon/label variations you might encounter are shown in the following images.



Next steps

- Navigate to your items from Microsoft Fabric Home
- Endorsement

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#) | [Ask the community ↗](#)

Data details

Article • 01/10/2023

The data details pages help you explore, monitor, and apply the data in your organization. When you select a data item in the [data hub](#), a workspace, or other place in the Power BI service, the detail page for that data item opens.

Select a data item type to see the details documentation for that type.

- [Dataset](#)
- [Datamart](#)

Dataset details

Article • 05/02/2023

The dataset details page helps you explore, monitor, and leverage datasets. When you select a dataset in the [data hub](#), a workspace, or other place in Power BI, the details page for that dataset opens.

The screenshot shows the Power BI dataset details page for 'Rates by segment'. The top navigation bar includes 'Power BI', 'Search', and a user icon. The header displays the dataset name, owner ('Debra Berger'), and various actions like 'Refresh', 'Share', 'Create a report', 'Analyze in Excel', 'Lineage', 'Chat in Teams', and 'Show tables'. The main content area is divided into sections: 'Dataset details' (Workspace: FinanceCorp, Endorsement: Promoted, Description: Add a description), 'Visualize this data' (button to 'Create a report'), 'Share this data' (button to 'Share dataset'), and 'Tables view' (list of tables: Client Types, DailyReportsSales, DateDimView, Mobile_ActivitiesCDSDim, Mobile_ActivitiesDailyCDSDim, Mobile_DevicesCDSDim, Mobile_FeaturesCDSDim, Mobile_UsersCDSDim). Below these is an 'Explore related reports' section with three items: 'Rates by segment analysis' (Report, Promoted, Workspace: FinanceCorp, Unique viewers: 47, Views: 53), 'Contact info' (Report, -, Workspace: FinanceCorp, Unique viewers: 41, Views: 47), and 'Yearly Summary' (Report, -, Workspace: FinanceCorp, Unique viewers: 25, Views: 32).

The dataset details page:

- Shows you metadata about the dataset, including description, endorsement, and sensitivity.
- Provides actions that you can perform on the dataset, such as share, refresh, create new, Analyze in Excel, and more.
- Lists the reports and scorecards that are built on top of the dataset.

The page header displays the dataset name, endorsement (if any), and dataset owner. To send an email to the dataset owner or the dataset certifier (if any), select the header and then select the name of the owner.

Supported actions

The dataset details page enables you to perform a number of actions. The actions available vary from user to user depending on their permissions on the data item, and thus not all actions are available for all users.

Action	Description	On Action bar, choose:
--------	-------------	------------------------

Action	Description	On Action bar, choose:
Download this file	Downloads the .pbix file for this dataset.	File > Download this file
Manage permissions	Opens the manage dataset permissions page.	File > Manage permissions
Settings	Opens the dataset settings page.	File > Settings
Refresh now	Launches a refresh of the dataset.	Refresh > Refresh now
Schedule refresh	Opens the dataset settings page where you can set scheduled refresh.	Refresh > Schedule refresh
Share	Opens the Share dataset dialog.	Share, or use the Share this data tile .
Create a report from scratch	Opens the report editing canvas where you can create a new report based on the dataset.	Create a report > From scratch , or use the Visualize this data tile .
Create a report from template	Creates a copy of the template in <i>My Workspace</i> . This action is only available if a related report template exists.	Create a report > From template , or use the Visualize this data tile .
Create a report as formatted table	Opens the formatted table editing canvas.	Create a report > As formatted table , or use the Visualize this data tile .
Analyze in Excel	Launches Analyze in Excel using this dataset.	Analyze in Excel
Open lineage view	Opens the lineage view for the dataset.	Lineage > Open lineage view
Impact analysis	Opens the impact analysis side pane for this dataset.	Lineage > Impact analysis
Chat in Teams	Invite people to start chatting in Teams . People you invite will receive a Teams chat message from you with a link to this dataset details page. If they have access to the dataset, the link will open this dataset details page in Teams.	Chat in Teams

Action	Description	On Action bar, choose:
Show tables	Opens a side panel showing the dataset's tables. In the tables view you can create table previews by selecting desired columns.	Show tables

View dataset metadata

Dataset details

Workspace Refreshed Sensitivity

Sales & Marketing 8/8/19, 9:43:47 AM General

Description Add a description

The dataset details section shows:

- The name of the workspace where the item is located.
- The exact time of the last refresh.
- Endorsement status and certifier (if certified).
- Sensitivity (if set).
- Description (if any). You can create or edit the description from here.

Explore related reports

The explore related reports section shows you all the reports and scorecards that are built on the dataset. You can create a copy of an item by selecting the line the item is on and selecting the **Save a copy** icon that appears. This section also shows you usage metrics for the related items.

Explore related reports
Check out other reports and scorecards that already use this dataset..

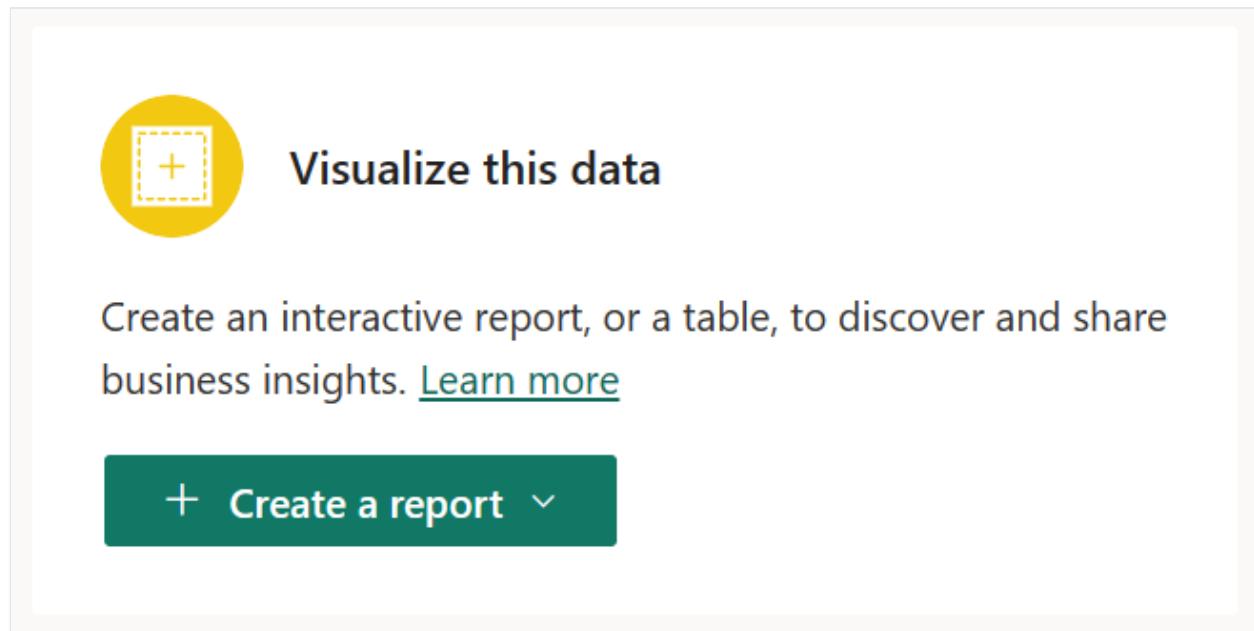
	Name	Type	Endorsement	Workspace	Unique viewers ⓘ	Views ⓘ
	Rates by segment analysis	Report	Promoted	FinanceCorp	57	147
	Contact info	Report	-	FinanceCorp	35	73
	Yearly summary	Report	-	FinanceCorp	46	89

The columns in the list of related reports are:

- **Name:** Report name. If the name ends with (template), it means that this report has been specially constructed to be used as a template. For example, "Sales (template)".
- **Type:** Item type, for example, report or scorecard.
- **Endorsement:** Endorsement status.
- **Workspace:** The name of the workspace where the related item is located.
- **Unique viewers:** Shows the total number of unique users who viewed the item at least once in the last 30 days, excluding the current day's views.
- **Views:** Shows the total number of times an item was viewed in the last 30 days, excluding the current day's views.

Visualize this data

To create a report based on the dataset, select the **Create report** button on this tile and choose the desired option.



- **Auto-create:** Creates an auto-generated report from the dataset.
- **From template:** Creates a copy of the template in *My workspace*.
- **From scratch:** Opens the report editing canvas to a new report built on the dataset. When you save your new report, it will be saved in the workspace that contains the dataset if you have write permissions on that workspace. If you don't have write permissions on the workspace, or if you are a free user and the dataset resides in a Premium-capacity workspace, the new report will be saved in *My workspace*.
- **As formatted table:** Opens the formatted table editing canvas.

Note

Only one template will be shown in the Create report drop-down, even if more than one report template exists for this dataset.

Share this data

You can share the dataset with other users in your organization. Selecting the **Share dataset** button opens the [Share dataset dialog](#), where you can choose which permissions to grant on the dataset.



Share this data

Give people access to the dataset and set their permissions to work with it. [Learn more](#)

Share dataset

Data preview

Data preview enables you to view a selected table or columns from the dataset. You can also export the data to supported file formats or create a [paginated report](#).

Prerequisites

- The dataset can be inside Premium or non-Premium workspaces. Classic workspaces aren't supported. [Read about new and classic workspaces](#).
- You need [Build permission](#) for the dataset.

Select data to preview

To preview a datasets's data from the dataset details page, select a table or columns on the **Tables** side panel.

The screenshot shows the 'Dataset details' page for 'As Azure Adventure Wo...'. It includes sections for 'Dataset details', 'Visualize this data', 'Share this data', and 'Explore related reports'. A red box highlights the 'Tables' panel on the right, which lists various tables and columns from the dataset.

Dataset details

- Workspace: Power BI Reach
- Refreshed: 6/14/21, 12:05:36 PM
- Sensitivity: Public

Description [Add a description](#)

Visualize this data

Create an interactive report, or a table, to discover and share business insights. [Learn more](#)

+ Create a report

Share this data

Give people access to the dataset and set their permissions to work with it. [Learn more](#)

Share dataset

Explore related reports

Check out other reports and scorecards that already use this dataset..

Name	Type	Endorsement	Workspace
As Azure Adventure Works	Report	-	Power BI Reach
Parameter Report From PBIDataset	Paginated Rep...	-	Power BI Reach
WeirdSlash	Paginated Rep...	-	Power BI Reach

Tables

Select tables and columns from this dataset to view and export the underlying data. [Learn more](#)

- [Alphabetical list of products](#)
- [Categories](#)
- [CustomerCustomerDemo](#)
- [Customers](#)
- [Employees](#)
- [EmployeeTerritories](#)
- [Invoices](#)
- [Order Subtotals](#)
- [Orders](#)
- [Orders Qry](#)
- [Products](#)
- [Table](#)

If you don't see the side panel, select **Show tables** on the action bar.

The screenshot shows the same dataset details page as above, but the 'Show tables' button in the top right corner of the main content area is highlighted with a red box. This indicates where the user should click to open the side panel.

An entirely filled parent checkbox on the dataset's table indicates that all its sub-tables and columns have been selected. A partially filled parent checkbox means that only a subset of them has been selected.

When you select a table or columns in a table, they will be displayed on the **Table preview** page that opens.

CategoryName	ProductName	QuantityPerUnit	UnitPrice
Beverages	Chai	10 boxes x 20 bags	\$18
Beverages	Chang	24 - 12 oz bottles	\$19
Beverages	Chartreuse verte	750 cc per bottle	\$18
Beverages	Côte de Blaye	12 - 75 cl bottles	\$263.5
Beverages	Ipoh Coffee	16 - 500 g tins	\$46
Beverages	Lakkalikööri	500 ml	\$18
Beverages	Laughing Lumberjack Lager	24 - 12 oz bottles	\$14
Beverages	Outback Lager	24 - 355 ml bottles	\$15
Beverages	Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75
Beverages	Sasquatch Ale	24 - 12 oz bottles	\$14
Beverages	Steeleye Stout	24 - 12 oz bottles	\$18
Condiments	Aniseed Syrup	12 - 550 ml bottles	\$10
Condiments	Chef Anton's Cajun Seasoning	48 - 6 oz jars	\$22
Condiments	Genen Shouyu	24 - 250 ml bottles	\$15.5
Condiments	Grandma's Boysenberry Spread	12 - 8 oz jars	\$25
Condiments	Gula Malacca	20 - 2 kg bags	\$19.45
Condiments	Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	\$21.05
Condiments	Louisiana Hot Spiced Okra	24 - 8 oz jars	\$17
Condiments	Northwoods Cranberry Sauce	12 - 12 oz jars	\$40
Condiments	Original Frankfurter grüne Soße	12 boxes	\$13
Condiments	Sirop d'éable	24 - 500 ml bottles	\$28.5
Condiments	Vegie-spread	15 - 625 g jars	\$43.9
Confections	Chocolade	10 pkgs.	\$12.75
Confections	Gumbär Gummibärchen	100 - 250 g bags	\$31.23
Confections	Maxilaku	24 - 50 g pkgs.	\$20
Confections	NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	\$14
Confections	Pavlova	32 - 500 g boxes	\$17.45
Confections	Schoggi Schokolade	100 - 100 g pieces	\$43.9
Confections	Scottish Longbreads	10 boxes x 8 pieces	\$12.5
Confections	Sir Rodney's Marmalade	30 gift boxes	\$81
Confections	Sir Rodney's Scones	24 pkgs. x 4 pieces	\$10
Confections	Tarte au sucre	48 pies	\$49.3
Confections	Teatime Chocolate Biscuits	10 boxes x 12 pieces	\$9.2

Table preview may not show all of the data you've selected. To see more, you can [export](#) or build a [paginated report](#).

You can resize column widths using a drag handle next to the column headers. Resizing columns can make the table preview more readable, especially for long column input values.

Show query

Show query enables you to copy the DAX query used to create the table preview to the clipboard. This makes it possible to reuse the query for future actions.

The screenshot shows the Power BI service interface with the 'Show query' feature highlighted. The top navigation bar includes 'Back', 'Show tables', and 'Show query'. A note below the navigation says 'Table previews only show up to 500 rows. Export your data or create a paginated report to see all rows.' The main area is titled 'Table preview' and shows a list of product data. Below the preview is a 'Query' pane containing the following DAX code:

```
DEFINE VAR __DS0Core =  
    SUMMARIZECOLUMNS(  
        'Alphabetical list of products'[ProductName],  
        'Alphabetical list of products'[QuantityPerUnit],  
        'Alphabetical list of products'[Discontinued],  
        'Alphabetical list of products'[CategoryName],  
        "SumUnitPrice", CALCULATE(SUM('Alphabetical list of products'[UnitPrice])),  
        "SumUnitsInStock", CALCULATE(SUM('Alphabetical list of products'[UnitsInStock])),  
        "SumUnitsOnOrder", CALCULATE(SUM('Alphabetical list of products'[UnitsOnOrder])),  
        "SumReorderLevel", CALCULATE(SUM('Alphabetical list of products'[ReorderLevel]))  
    )  
  
EVALUATE  
    __DS0Core  
  
ORDER BY  
    'Alphabetical list of products'[ProductName],  
    'Alphabetical list of products'[QuantityPerUnit],  
    'Alphabetical list of products'[Discontinued],  
    'Alphabetical list of products'[CategoryName]
```

At the bottom of the 'Query' pane are 'Copy to clipboard' and 'Cancel' buttons.

[Back](#)

At any time you can return to the dataset details page by selecting the **Back** button on the action bar. Selecting the Back button clears all your selections and brings you back to dataset details page.

ⓘ Note

Table preview is intended to quickly explore the underlying data of tables within your dataset. You cannot view measures or select more than one table or columns across tables. You can select **Create paginated report** for that.

Export data

Select the **Export** button on the Table preview page to export the data to one of the supported file formats.

The screenshot shows a dropdown menu from a 'Export' button. The menu includes options for Microsoft Excel (.xlsx), PDF (.pdf), Accessible PDF (.pdf), Comma Separated Values (.csv), Microsoft PowerPoint (.pptx), Microsoft Word (.docx), and XML (.xml). Below the menu, the value '\$22' is displayed. To the right of the menu, a table preview shows the 'UnitsOnOrder' column with values: 13, 123, 19, 42, 39, 17, 69, and 53.

UnitsOnOrder
13
123
19
42
39
17
69
53

Build a paginated report

Select the **Create paginated report** button to open the [editor](#).

ⓘ Note

Data will change from underlying data to summarized data. You can switch to underlying data using **More options**.

In the editor you can select multiple tables, measure, fields across tables, apply table styles, change aggregates, and so on.

The screenshot shows the Microsoft Power BI Editor interface. On the left is a table of product data with columns: CategoryName, ProductName, QuantityPerUnit, UnitPrice, and Unit. The table lists various items like Chai, Chang, Chartreuse verte, etc., categorized by Beverage, Condiments, and Confections. To the right of the table is a 'Build' pane which includes a 'Fields' section and a 'Values' section. The 'Fields' section contains a search bar and a list of checked and unchecked fields: CategoryID, CategoryName, Discontinued, ProductID, ProductName, QuantityPerUnit, ReorderLevel, SupplierID, and UnitPrice. The 'Values' section also contains a search bar and a list of checked and unchecked values: CategoryName, ProductName, QuantityPerUnit, Sum of UnitPrice, Sum of UnitsInStock, Sum of UnitsOnOrder, Sum of ReorderLevel, Count of OrderID, and Sum of Subtotal. At the bottom of the editor, there is a note: "When authoring, only 500 rows are shown. Go to reading view or export the report to see all rows".

You can then export the report to any of the supported file formats, and the file will be saved to your default downloads folder. Or you can save it as a [paginated report](#) to a workspace of your choice. Paginated reports fully preserve your report formatting.

Switch from summarized to underlying data in the editor

Select **More options (...)** to switch from **Summarized data** to **Underlying data**.

Fields >>

Search

simplereport

Summarized data

Underlying data

ProductName

SalesTable

The screenshot shows the 'Fields' pane in Power BI. At the top left is a search bar with a magnifying glass icon. To its right is a double-right arrow icon. Below the search bar is a list of datasets. The first dataset, 'simplereport', has a small icon next to it. A hand cursor icon is positioned over the 'simplereport' entry. Below 'simplereport' is a collapsed section indicated by a minus sign and a grid icon. This section contains two items: 'Summarized data' with a green checkmark and 'Underlying data'. Underneath these is another item, 'ProductName', also with a green checkmark. At the bottom of the list is another collapsed section indicated by a greater than sign and a grid icon. This section contains one item, 'SalesTable', preceded by a small square icon.

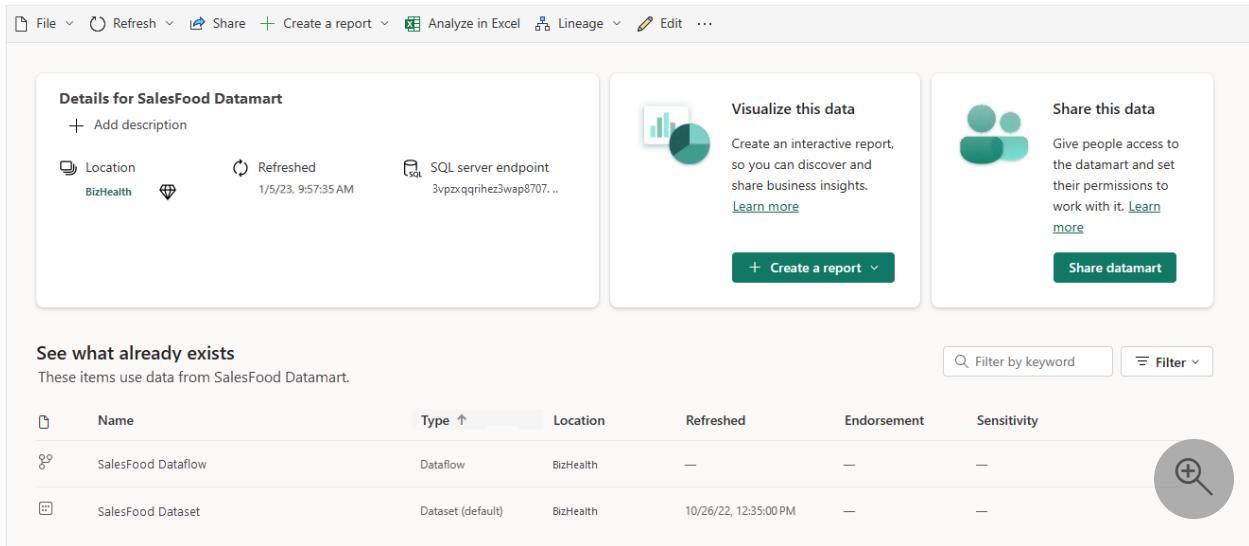
Next steps

- Use datasets across workspaces
- Create reports based on datasets from different workspaces
- Endorse your dataset
- Questions? Try asking the Power BI Community

Datamart details

Article • 03/13/2023

The datamart details page helps you explore, monitor, and use datamarts. When you select a datamart in the data hub, the details page for that datamart opens.



The screenshot shows the 'Datamart details' page for the 'SalesFood Datamart'. At the top, there's a navigation bar with options like File, Refresh, Share, Create a report, Analyze in Excel, Lineage, Edit, and more. Below the header, the datamart's name is displayed along with its location (BizHealth), refresh status (Refreshed 1/5/23, 9:57:35 AM), and connection string (SQL server endpoint). There are two main sections: 'Visualize this data' (with a 'Create a report' button) and 'Share this data' (with a 'Share datamart' button). A section titled 'See what already exists' lists existing items: 'SalesFood Dataflow' (Dataflow, BizHealth, Refreshed 1/5/23, 9:57:35 AM) and 'SalesFood Dataset' (Dataset (default), BizHealth, Refreshed 10/26/22, 12:35:00 PM). A search bar and filter button are also present.

The datamart details page:

- Shows you metadata about the datamart, including description, endorsement, and sensitivity, and connection string.
- Provides actions that you can perform on the datamart, such as share, refresh, create new, and Analyze in Excel.
- Lists the reports that are built on top of the datamart.

Supported actions

The datamart details page enables you to perform actions on the datamart. The following table lists all supported actions. On the datamart details page you may see only a subset of this list, as only actions you have permission to perform are listed.

Action	Description	On Action bar, choose:
Manage permissions	Opens the manage datamart permissions page.	File > Manage permissions
Settings	Opens the datamart settings page.	File > Settings
Refresh now	Launches a refresh of the dataset.	Refresh > Refresh now

Action	Description	On Action bar, choose:
Schedule refresh	Opens the dataset settings page where you can set scheduled refresh.	Refresh > Schedule refresh
Refresh history	Opens Refresh history window where you see the time, duration, and status of each refresh. You can download the history as a .csv file.	Refresh > Refresh history
Share	Opens the Share datamart dialog. Sharing a datamart allows recipients to build content based on the underlying dataset and query the corresponding SQL endpoint.	Share, or use the Share this data tile.
Create a report from scratch	Opens the report editing canvas where you can create a new report based on the datamart.	Create a report > From scratch, or use the Visualize this data tile.
Analyze in Excel	Launches Analyze in Excel using this datamart.	Analyze in Excel
Open lineage view	Opens the lineage view for the datamart.	Lineage > Open lineage view
Impact analysis	Opens the impact analysis side pane for this datamart.	Lineage > Impact analysis
Edit	Opens the datamart in the Datamart editor.	Edit

View datamart metadata

The screenshot shows the 'Details for Sales DM' section. It includes the following information:

- Details for Sales DM**: The name of the datamart.
- This a datamart for sales**: A descriptive text.
- Location**: Teams (with a diamond icon).
- Certified by**: Administrator (Debra...).
- Refreshed**: 3/9/23, 1:31:16 AM (with a red triangle icon).
- Sensitivity**: Public.
- +1**: A button to add a follower.

The datamart details section shows:

- The location of the datamart.
- Endorsement status (and certifier, if certified).
- The exact time of the last refresh.
- Sensitivity, if set.
- T-SQL connection string.

- Description, if any.

Explore related reports

The **See what already exists** section shows you reports that are built on top of the datamart's auto-generated dataset. You can create a copy of a report by selecting the line the item is on and clicking the **Save a copy** icon that appears. This section also shows you usage metrics for the related items.

See what already exists						
These items use data from SalesFood Datamart.						
	Name	Type	Location	Refreshed	Endorsement	Sensitivity
	SalesFood Dataflow	Dataflow	BizHealth	—	—	—
	SalesFood Dataset	Dataset (default)	BizHealth	10/26/22, 12:35:00 PM	—	—

Note

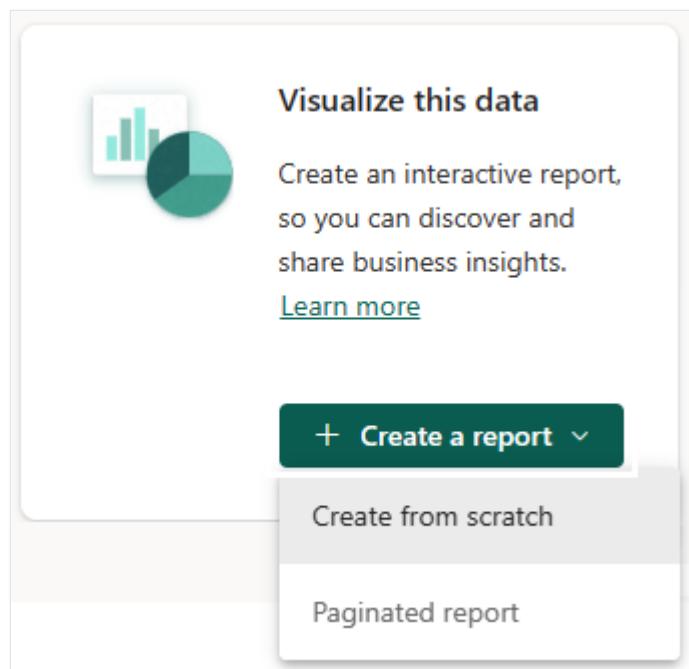
Reports build on top of other datasets created from the datamart aren't shown in this section.

The columns in the list of related reports are:

- **Name:** Report name. If the name ends with **(template)**, the report has been specially constructed to be used as a template. For example, "Sales (template)".
- **Type:** Item type, for example, report or scorecard.
- **Location:** The name of the workspace where the related item is located.
- **Refreshed:** Time of last refresh.
- **Endorsement:** Endorsement status.
- **Sensitivity:** Sensitivity.
- **Unique viewers:** Shows the total number of unique users who viewed the item at least once in the last 30 days, excluding the current day's views.
- **Views:** Shows the total number of times an item was viewed in the last 30 days, excluding the current day's views.

Visualize this data

To create a report based on the dataset, select the **Create a report** button on this tile and choose the desired option.



- The **Create from scratch** option opens the report editing canvas to a new report built on the dataset.

When you save your new report, it's saved in the workspace that contains the dataset, if you have write permissions on that workspace.

If you don't have write permissions on the workspace, the report is saved in My workspace. If you're a free user and the dataset resides in a Premium-capacity workspace, the report is saved in My workspace.

- The **Paginated report** option opens the paginated report online editor. For information about creating a paginated report using the online editor, see [Create exportable paginated reports in the Power BI service](#).

Share this data

You can share the datamart with other users in your organization. Selecting the **Share datamart** button opens the [Share datamart dialog](#). People you share the datamart with can build content based on the underlying dataset and query the corresponding SQL endpoint.



Share this data

Give people access to the datamart and set their permissions to work with it. [Learn more](#)

[Share datamart](#)

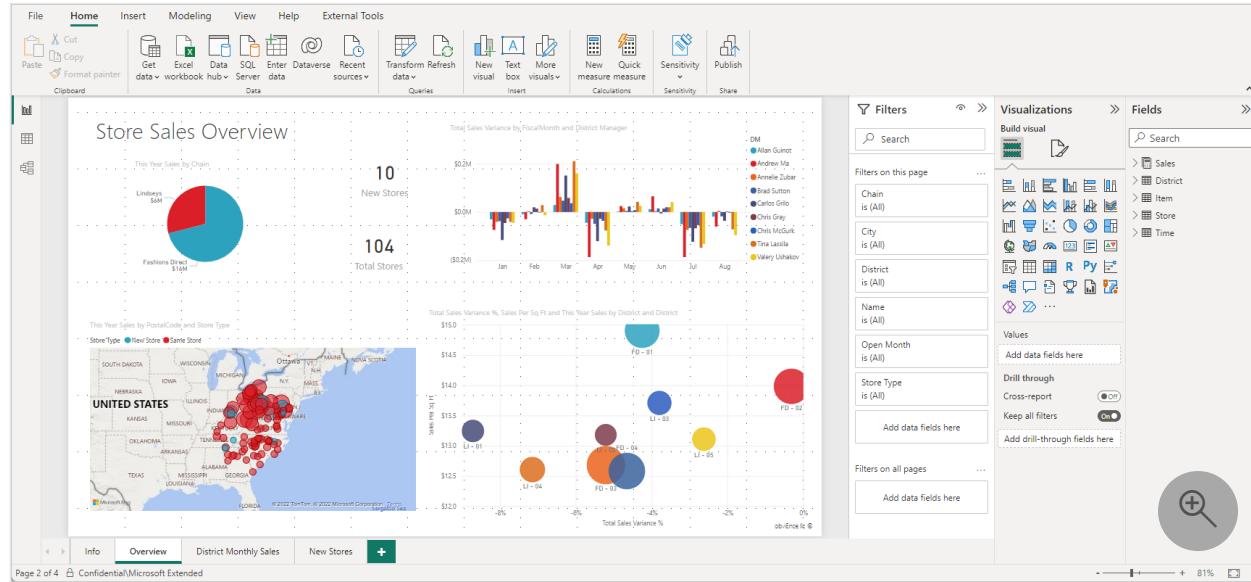
Next steps

- Create exportable paginated reports in the Power BI service
- Endorse your content
- Questions? [Try asking the Power BI Community](#) ↗

Quickstart: Connect to data in Power BI Desktop

Article • 01/12/2023

In this quickstart, you connect to data using Power BI Desktop, which is the first step in building data models and creating reports.



If you're not signed up for Power BI, [sign up for a free trial](#) before you begin.

Prerequisites

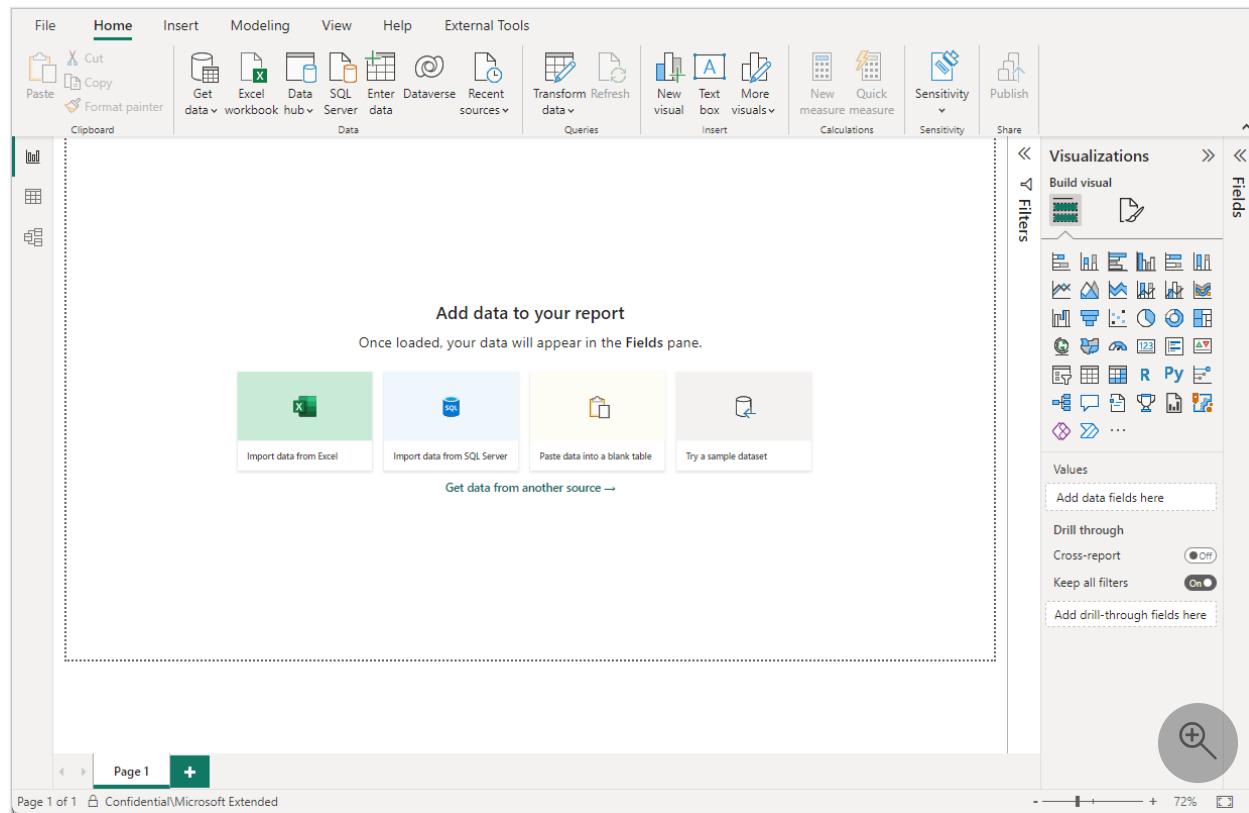
To complete the steps in this article, you need the following resources:

- Download and install Power BI Desktop, which is a free application that runs on your local computer. You can [download Power BI Desktop](#) directly, or you can get it from [the Microsoft Store](#).
- [Download this sample Excel workbook](#), and create a folder called `C:\PBID-qs` where you can store the Excel file. Later steps in this quickstart assume that is the file location for the downloaded Excel workbook.
- For many data connectors in Power BI Desktop, Internet Explorer 10 (or newer) is required for authentication.

Launch Power BI Desktop

Once you install Power BI Desktop, launch the application so it's running on your local computer. You're presented with a Power BI tutorial. Follow the tutorial or close the

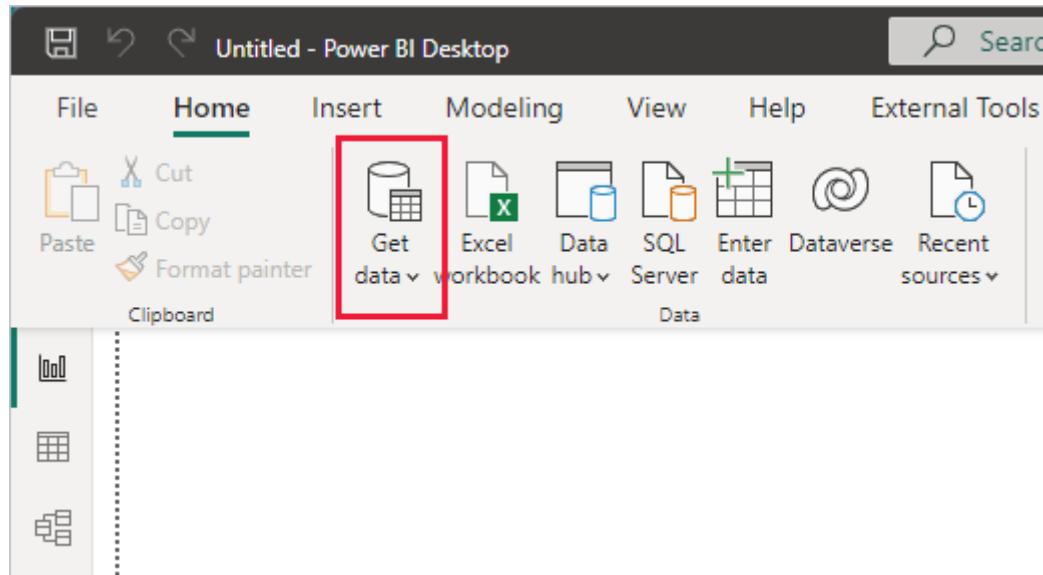
dialog to start with a blank canvas. The canvas is where you create visuals and reports from your data.



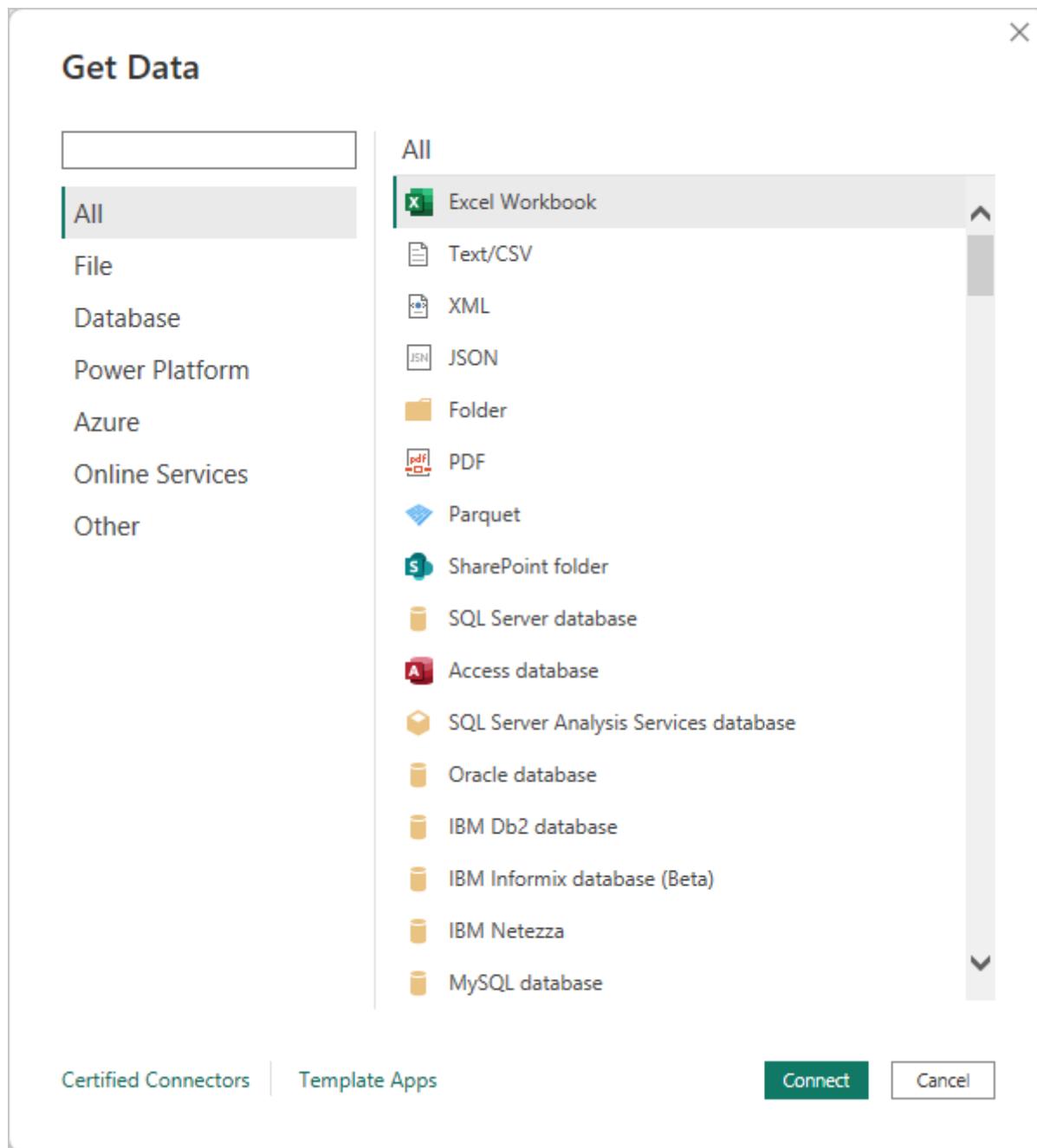
Connect to data

With Power BI Desktop, you can connect to many different types of data. These sources include basic data sources, such as a Microsoft Excel file. You can connect to online services that contain all sorts of data, such as Salesforce, Microsoft Dynamics, Azure Blob Storage, and many more.

To connect to data, from the Home ribbon select Get data.

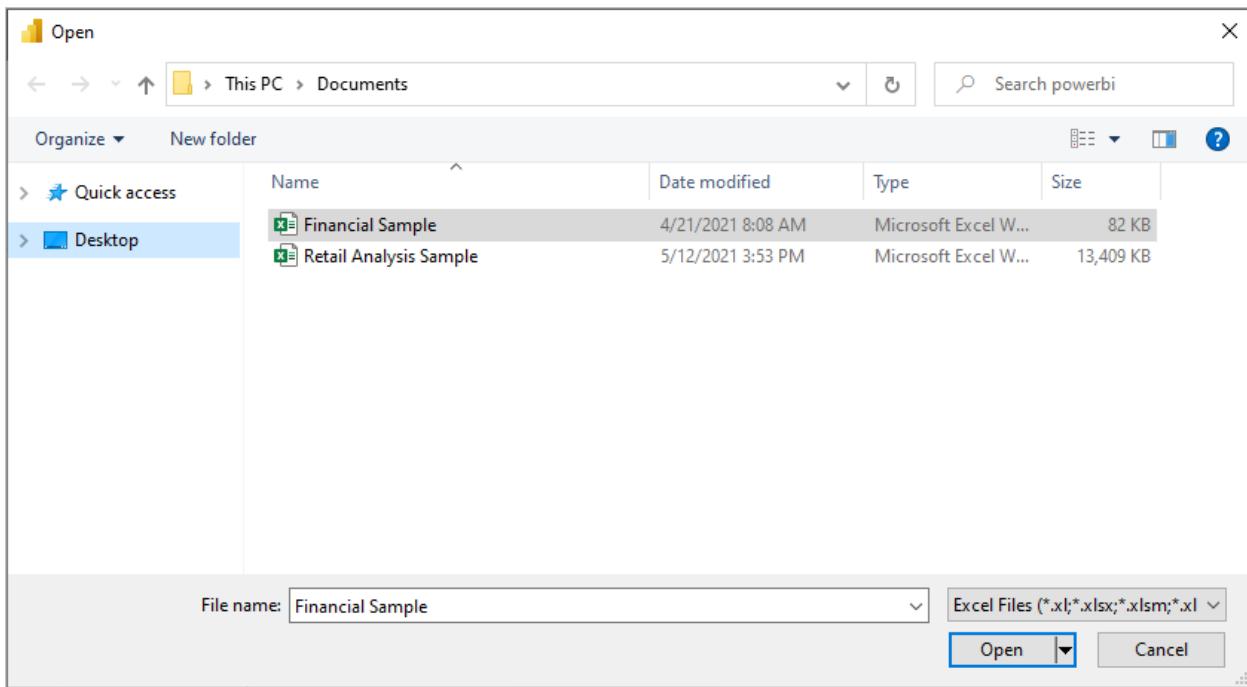


The **Get Data** window appears. You can choose from the many different data sources to which Power BI Desktop can connect. In this quickstart, use the Excel workbook that you downloaded in [Prerequisites](#).



Since this data source is an Excel file, select **Excel** from the **Get Data** window, then select the **Connect** button.

Power BI prompts you to provide the location of the Excel file to which to connect. The downloaded file is called *Financial Sample*. Select that file, and then select **Open**.



Power BI Desktop then loads the workbook and reads its contents, and shows you the available data in the file using the **Navigator** window. In that window, you can choose which data you would like to load into Power BI Desktop. Select the tables by marking the checkboxes beside each table you want to import. Import both available tables.

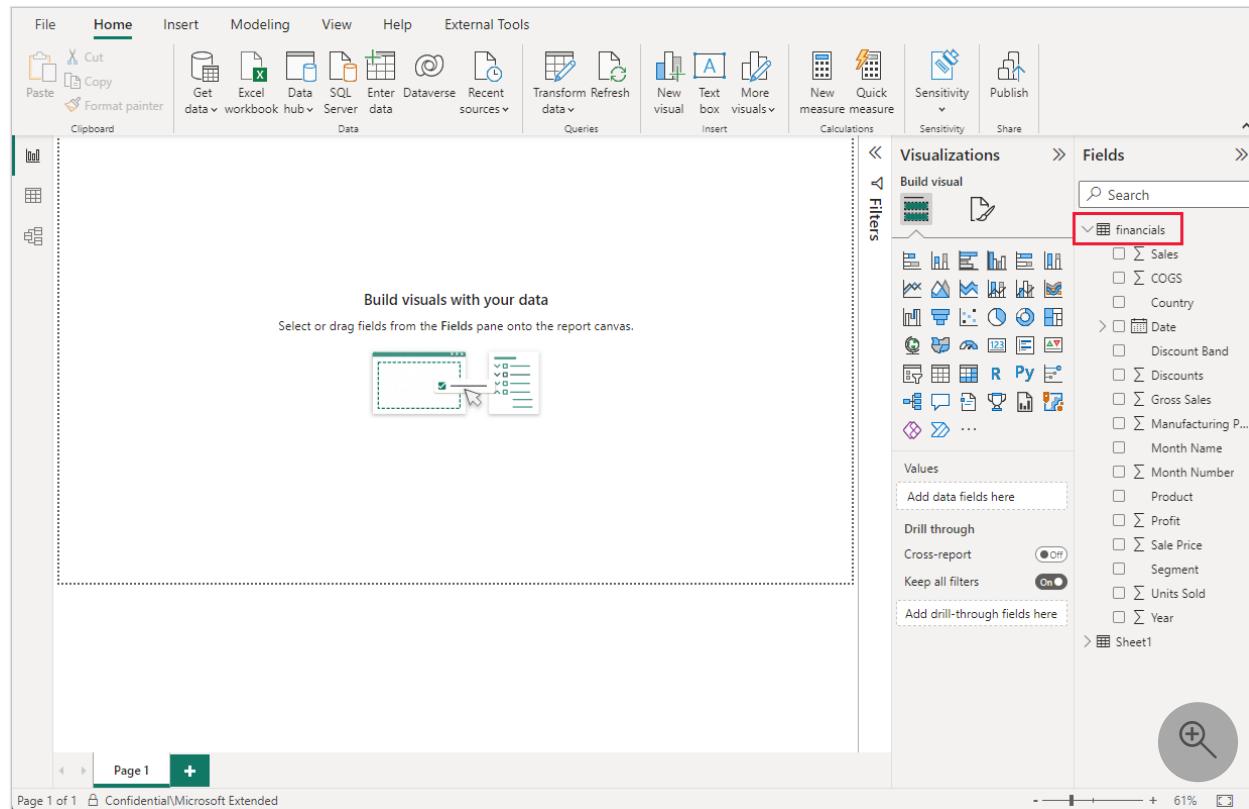
The screenshot shows the Power BI Desktop Navigator window. On the left, there is a tree view showing 'Financial Sample.xlsx [2]' with 'financials' and 'Sheet1' selected. On the right, the 'Sheet1' data is displayed in a table format. The table has columns: Segment, Country, Product, and Discount Band. The data includes various segments like Government, Midmarket, and Enterprise across countries like Canada, Germany, France, Mexico, and United States of America, with products like Carretera, Paseo, and Montana, and discount bands like None and None. At the bottom right, there are buttons for 'Load', 'Transform Data', and 'Cancel'.

Segment	Country	Product	Discount Band
Government	Canada	Carretera	None
Government	Germany	Carretera	None
Midmarket	France	Carretera	None
Midmarket	Germany	Carretera	None
Midmarket	Mexico	Carretera	None
Government	Germany	Carretera	None
Midmarket	Germany	Montana	None
Channel Partners	Canada	Montana	None
Government	France	Montana	None
Channel Partners	Germany	Montana	None
Midmarket	Mexico	Montana	None
Enterprise	Canada	Montana	None
Small Business	Mexico	Montana	None
Government	Germany	Montana	None
Enterprise	Canada	Montana	None
Midmarket	United States of America	Montana	None
Government	Canada	Paseo	None
Midmarket	Mexico	Paseo	None
Channel Partners	Canada	Paseo	None
Government	Germany	Paseo	None
Channel Partners	Germany	Paseo	None
Government	Mexico	Paseo	None
Midmarket	France	Paseo	None

Once you've made your selections, select **Load** to import the data into Power BI Desktop.

View data in the Fields pane

Once you've loaded the tables, the **Fields** pane shows you the data. You can expand each table by selecting the arrow beside its name. In the following image, the *financials* table is expanded, showing each of its fields.



And that's it! You've connected to data in Power BI Desktop, loaded that data, and now you can see all the available fields within those tables.

Next steps

There are all sorts of things you can do with Power BI Desktop once you've connected to data. You can create visuals and reports. Take a look at the following resource to get you going:

- [Get started with Power BI Desktop](#)

Tutorial: Shape and combine data in Power BI Desktop

Article • 01/12/2023

With Power BI Desktop, you can connect to many different types of data sources, then shape the data to meet your needs, enabling you to create visual reports to share with others. *Shaping* data means transforming the data: renaming columns or tables, changing text to numbers, removing rows, setting the first row as headers, and so on. *Combining* data means connecting to two or more data sources, shaping them as needed, then consolidating them into a single query.

In this tutorial, you'll learn how to:

- Shape data by using Power Query Editor.
- Connect to different data sources.
- Combine those data sources, and create a data model to use in reports.

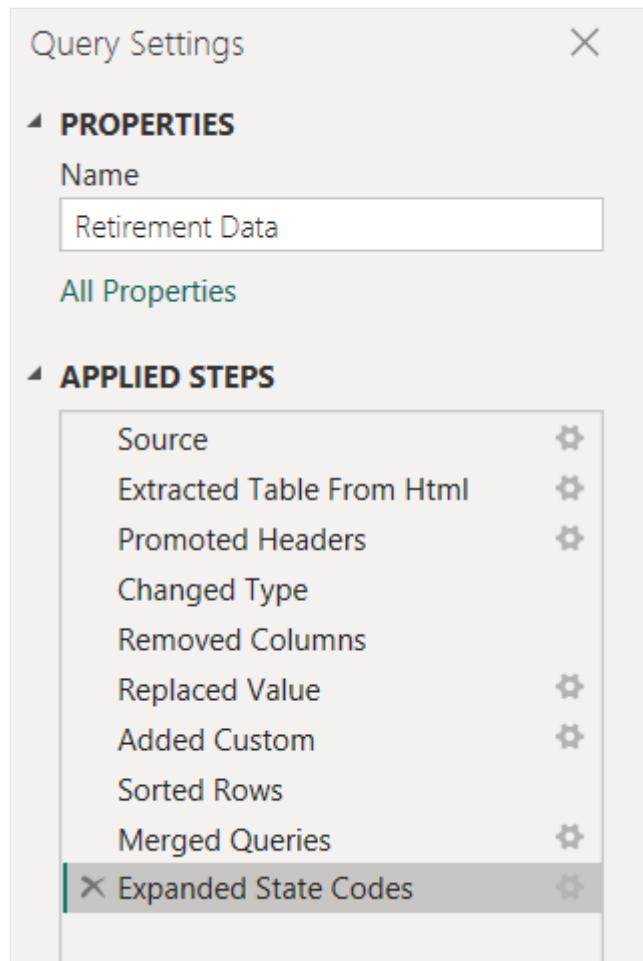
This tutorial demonstrates how to shape a query by using Power BI Desktop, highlighting the most common tasks. The query used here is described in more detail, including how to create the query from scratch, in [Getting Started with Power BI Desktop](#).

Power Query Editor in Power BI Desktop uses the right-click menus, and the **Transform** ribbon. Most of what you can select in the ribbon is also available by right-clicking an item, such as a column, and choosing from the menu that appears.

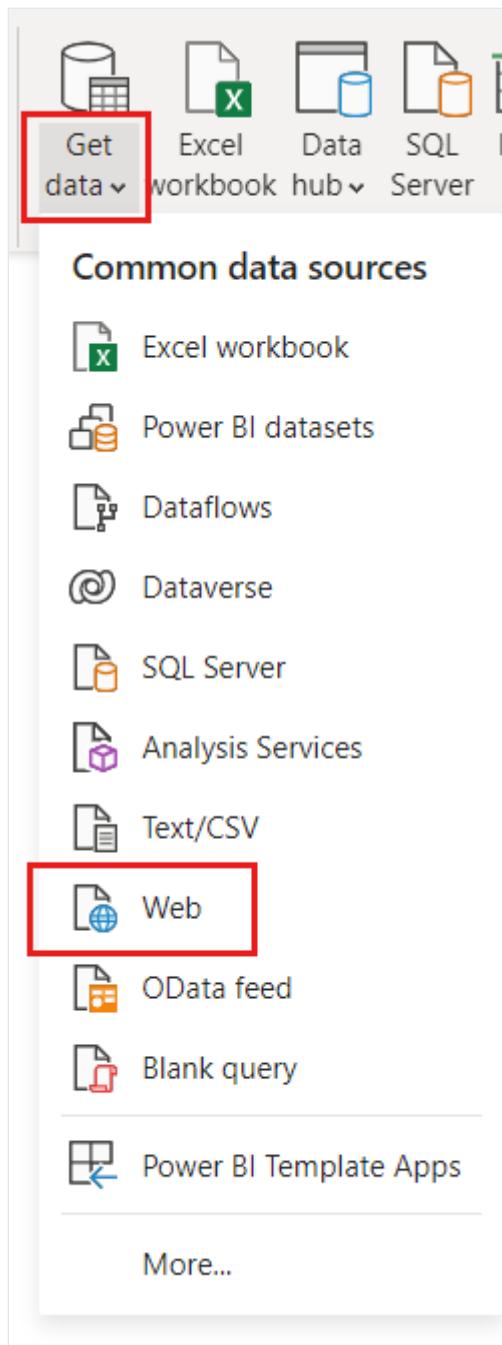
Shape data

To shape data in Power Query Editor, you provide step-by-step instructions for Power Query Editor to adjust the data as it loads and presents the data. The original data source isn't affected; only this particular view of the data is adjusted, or *shaped*.

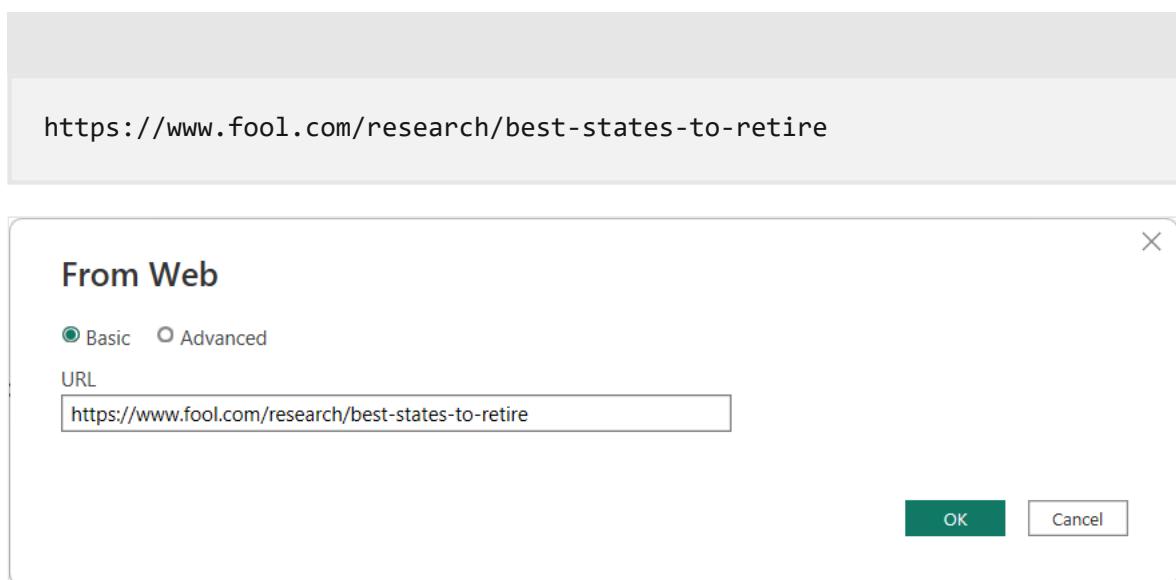
The steps you specify (such as rename a table, transform a data type, or delete a column) are recorded by Power Query Editor. Each time this query connects to the data source, Power Query Editor carries out those steps so that the data is always shaped the way you specify. This process occurs whenever you use Power Query Editor, or for anyone who uses your shared query, such as on the Power BI service. Those steps are captured, sequentially, in the **Query Settings** pane, under **APPLIED STEPS**. We'll go through each of those steps in this article.



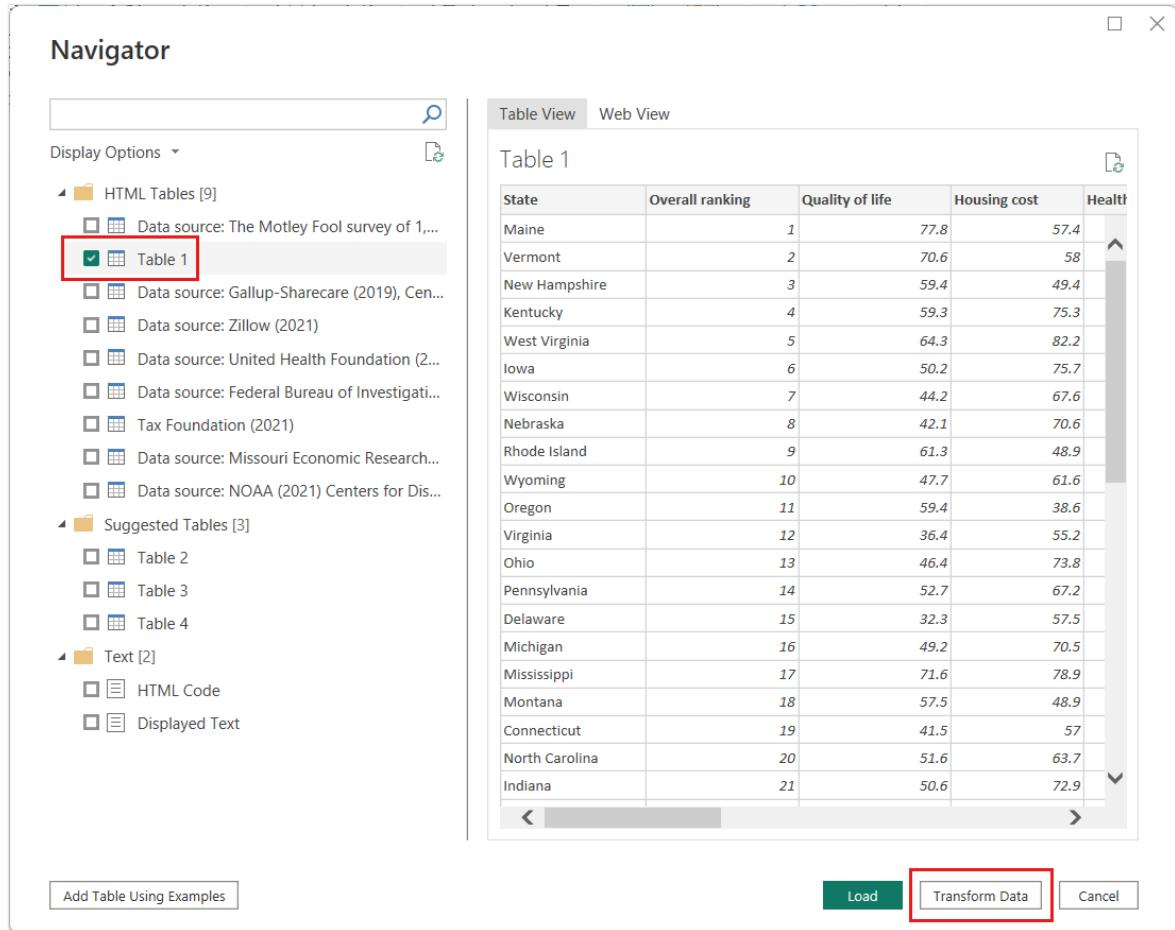
1. Import the data from a web source. Select the **Get data** dropdown, then choose **Web**.



2. Paste this URL into the **From Web** dialog and select **OK**.



3. In the Navigator dialog, select Table 1, then choose Transform Data.



4. The Power Query Editor window opens. You can see the default steps applied so far, in the **Query Settings** pane under **APPLIED STEPS**.

- **Source:** Connecting to the website.
- **Extracted Table from Html:** Selecting the table.
- **Promoted Headers:** Changing the top row of data into column headers.
- **Changed Type:** Changing the column types, which are imported as text, to their inferred types.

Table 1

Overall ranking

Quality of life

Housing cost

Healthcare cost and quality

Crime rate rate

Pub.

Retirement Data

Properties

Applied Steps

Changed Type

5. Change the table name from the default `Table 1` to `Retirement Data`, then press

Enter.

Query Settings

Properties

Name

Retirement Data

All Properties

Applied Steps

Source

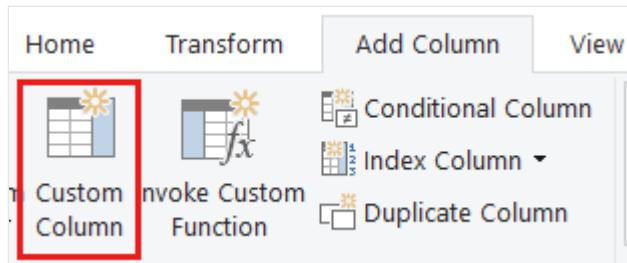
Extracted Table From Html

Promoted Headers

Changed Type

6. The existing data is ordered by a weighted score, as described on the source web page under [Methodology](#). Let's add a custom column to calculate a different score. We'll then sort the table on this column to compare the custom score's ranking to the existing Rank.

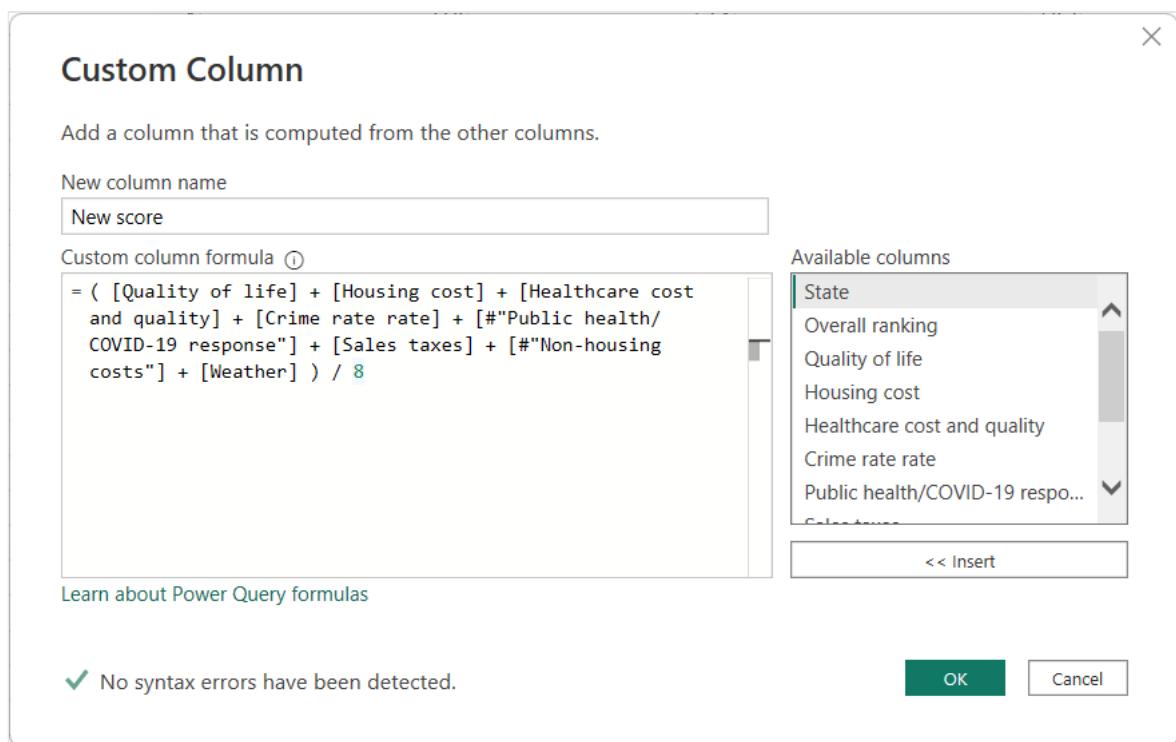
7. From the Add Column ribbon, select Custom Column.



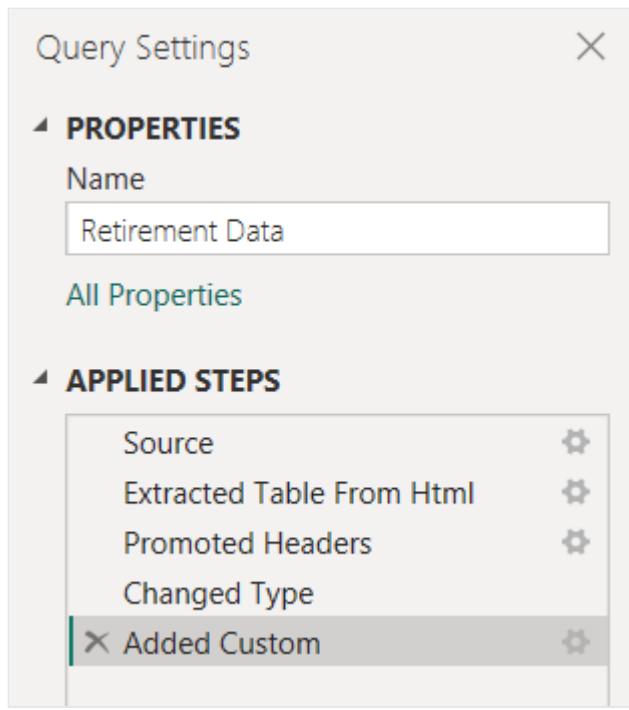
8. In the **Custom Column** dialog, in **New column name**, enter *New score*. For the **Custom column formula**, enter the following data:

```
( [Quality of life] + [Housing cost] + [Healthcare cost and quality] +
[Crime rate rate] + [#"Public health/COVID-19 response"] + [Sales
taxes] + [#"Non-housing costs"] + [Weather] ) / 8
```

9. Make sure the status message is *No syntax errors have been detected*, and select **OK**.



10. In **Query Settings**, the **APPLIED STEPS** list now shows the new **Added Custom** step we just defined.



Adjust the data

Before we work with this query, let's make a few changes to adjust its data:

- Adjust the rankings by removing a column.

For example, assume **Weather** isn't a factor in our results. Removing this column from the query doesn't affect the other data.

- Fix any errors.

Because we removed a column, we need to adjust our calculations in the **New score** column by changing its formula.

- Sort the data.

Sort the data based on the **New score** column, and compare to the existing **Rank** column.

- Replace the data.

We'll highlight how to replace a specific value and how to insert an applied step.

These changes are described in the following steps.

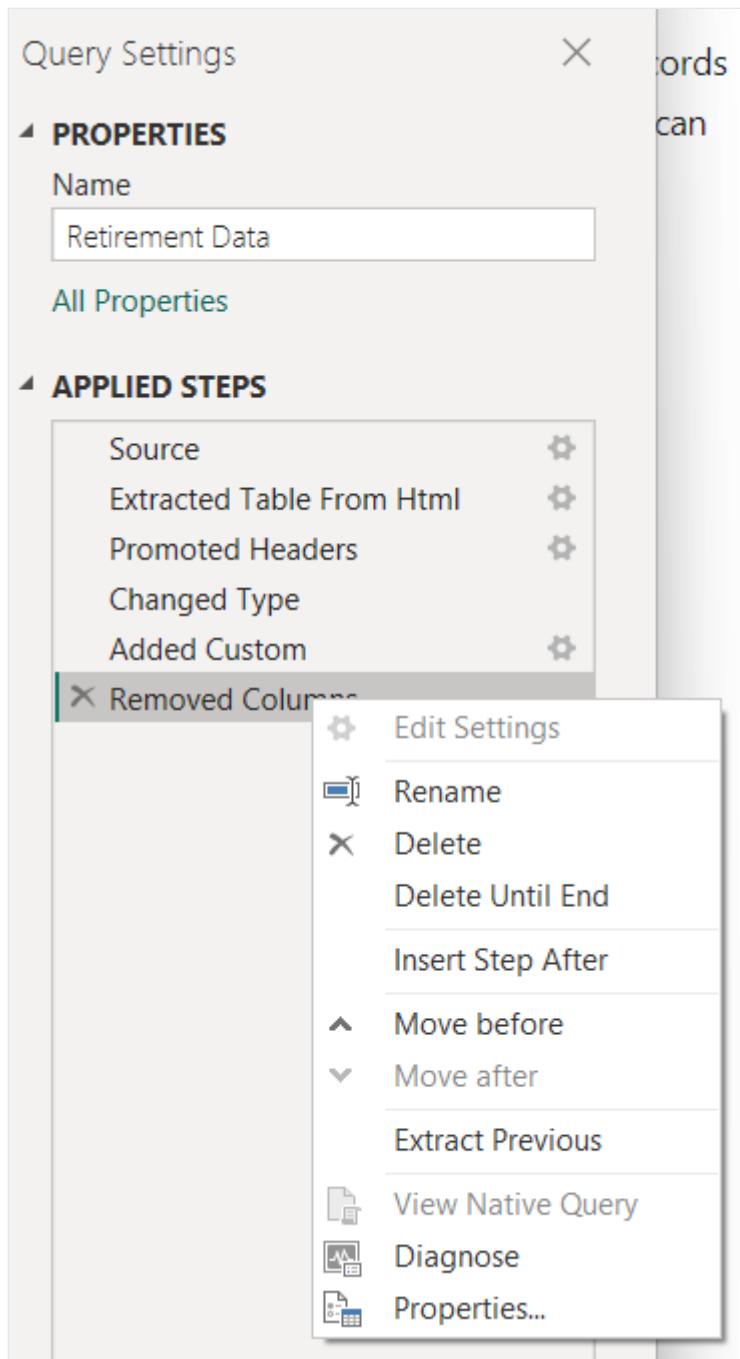
1. To remove the **Weather** column, select the column, choose the **Home** tab from the ribbon, and then choose **Remove Columns**.

The screenshot shows the Power Query Editor ribbon with the 'Transform' tab selected. The 'Transform' tab has several icons: 'Choose Columns', 'Remove Columns' (which is highlighted with a red box), 'Keep Rows', 'Remove Rows', 'Reduce Rows', 'Sort', 'Split Column', 'Group By', and 'Replace Value'. Below the ribbon, there is a preview area showing a table with three columns: '1.2 Non-housing costs', '1.2 Weather', and '1.3 Housing cost'. The first row contains values '42.6', '55.8', and '12.1'. Above the preview area, there is a formula bar with the text: 'Type", "New score", each ([Quality of life] + [Housing cost])'.

① Note

The **New score** values haven't changed, due to the ordering of the steps. Power Query Editor records the steps sequentially, yet independently, of each other. To apply actions in a different sequence, you can move each applied step up or down.

2. Right-click a step to see its context menu.



3. Move up the last step, **Removed Columns**, to just above the **Added Custom** step.

Query Settings X

PROPERTIES

Name
Retirement Data

[All Properties](#)

APPLIED STEPS

- Source ⚙️
- Extracted Table From Html ⚙️
- Promoted Headers ⚙️
- Changed Type ⚙️
- Removed Columns** ✖️
- Added Custom** ⚙️

4. Select the **Added Custom** step.

Notice the **New score** column now shows *Error* rather than the calculated value.

ABC	123	New score	▼
		Error	

There are several ways to get more information about each error. If you select the cell without clicking on the word *Error*, Power Query Editor displays the error information.

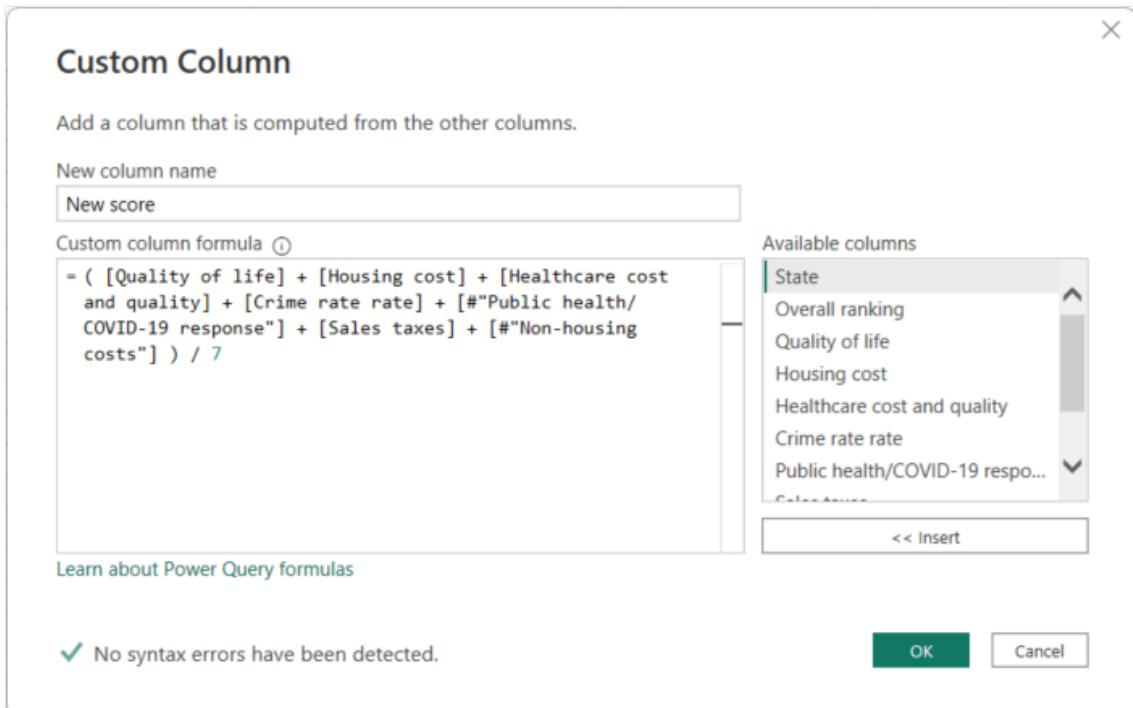
The screenshot shows a Power Query Editor window with a table containing three columns: "1.2 Non-housing costs", "1.2 Total score", and "New score". The "New score" column displays numerical values followed by the word "Error". A yellow error bar at the bottom provides details about the error, stating "Expression.Error: The field 'Weather' of the record wasn't found." It also lists "Details: State=Maine Overall ranking=1".

	1.2 Non-housing costs	1.2 Total score	New score
1	42.6	55.8	56.6 Error
2	38.9	52.9	55.04 Error
3	68.7	54.5	54.62 Error
4	39.4	68.5	51.48 Error
5	38.1	71.3	50.66 Error
6	39.1	71.2	50.21 Error
7	44	67.3	49.32 Error
8	39	68.7	48.61 Error
9	36.2	52.1	48.58 Error
10	45.9	68.7	48.43 Error
..			

If you select the word *Error* directly, Power Query Editor creates an **Applied Step** in the **Query Settings** pane and displays information about the error. Because we don't need to display error information anywhere else, select **Cancel**.

5. To fix the errors, there are two changes needed, removing the *Weather* column name and changing the divisor from 8 to 7. You can make these changes in two ways:

- a. Right-click the **Custom Column** step and select **Edit Settings**. This brings up the **Custom Column** dialog you used to create the **New score** column. Edit the formula as described previously, until it looks like this:



- b. Select the **New score** column, then display the column's data formula by enabling the **Formula Bar** checkbox from the **View** tab.

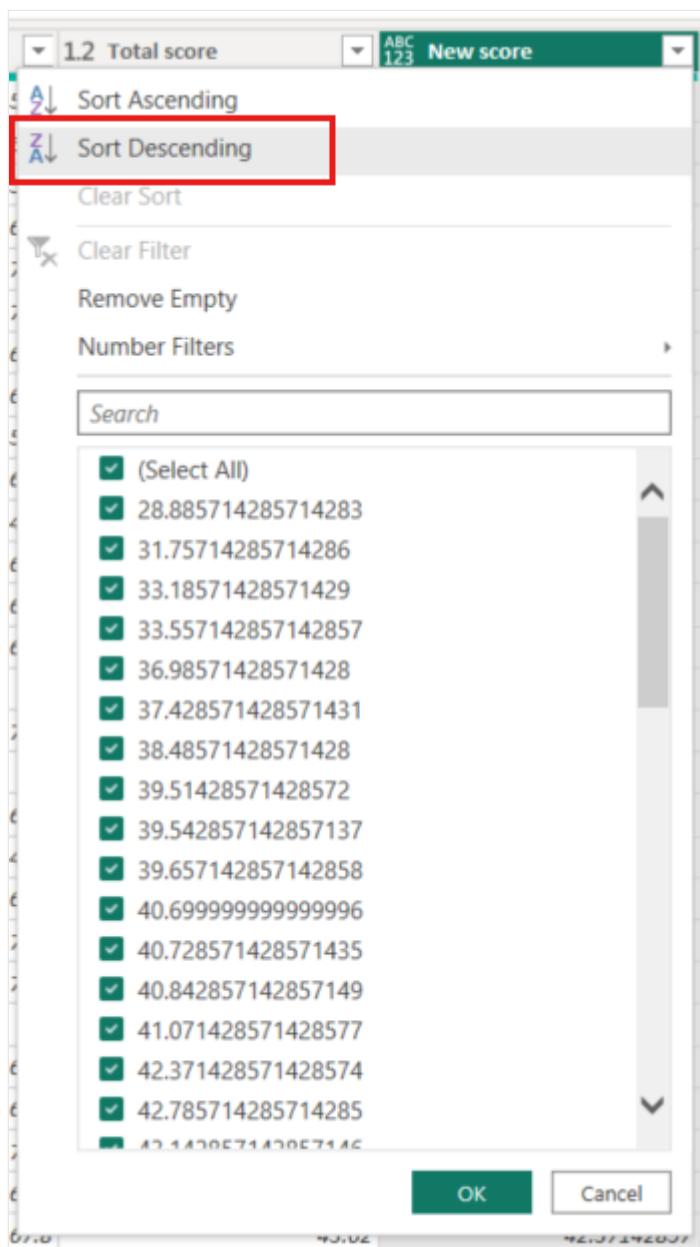
Edit the formula as described previously, until it looks like this, then press **Enter**.

Power Query Editor replaces the data with the revised values and the **Added Custom** step completes with no errors.

! Note

You can also select **Remove Errors**, by using the ribbon or the right-click menu, which removes any rows that have errors. However, in this tutorial we want to preserve all the data in the table.

6. Sort the data based on the **New score** column. First, select the last applied step, **Added Custom** to display the most recent data. Then, select the drop-down located next to the **New score** column header and choose **Sort Descending**.



The data is now sorted according to **New score**. You can select an applied step anywhere in the list, and continue shaping the data at that point in the sequence. Power Query Editor automatically inserts a new step directly after the currently selected applied step.

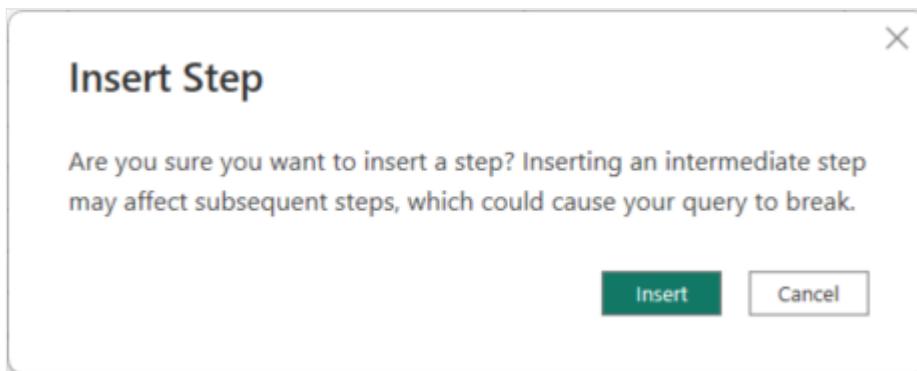
7. In **APPLIED STEPS**, select the step preceding the custom column, which is the **Removed Columns** step. Here we'll replace the value of the **Housing cost** ranking in Oregon. Right-click the appropriate cell that contains Oregon's **Housing cost** value, and then select **Replace Values**. Note which **Applied Step** is currently selected.

A screenshot of the Power Query Editor interface. A context menu is open over a row in a table. The menu items are: Copy, Number Filters, Replace Values..., Drill Down, Add as New Query. The 'Replace Values...' option is highlighted with a red box.

	State	Overall ranking	1.2 Quality of life	1.2 Housing cost	1.2 Healthcare cost an
1	Maine	1	77.8	57.4	
2	Vermont	2	70.6	58	
3	New Hampshire	3	59.4	49.4	
4	Kentucky	4	59.3	75.3	
5	West Virginia	5	64.3	82.2	
6	Iowa	6	50.2	75.7	
7	Wisconsin	7	44.2	67.6	
8	Nebraska	8	42.1	70.6	
9	Rhode Island	9	61.3	48.9	
10	Wyoming	10	47.7	61.6	
11	Oregon	11	59.4	29.6	
12	Virginia	12	36.4		
13	Ohio	13	46.4		
14	Pennsylvania	14	52.7		
15	Delaware	15	32.3		
16	Michigan	16	49.2		
17	Mississippi	17	71.6	78.9	

8. Select Insert.

Because we're inserting a step, Power Query Editor reminds us that subsequent steps could cause the query to break.



9. Change the data value to 100.0.

Power Query Editor replaces the data for Oregon. When you create a new applied step, Power Query Editor names it based on the action, in this case, **Replaced Value**. If you have more than one step with the same name in your query, Power Query Editor appends an increasing number to each subsequent applied step's name.

10. Select the last Applied Step, Sorted Rows.

Notice the data has changed regarding Oregon's new ranking. This change occurs because we inserted the **Replaced Value** step in the correct location, before the **Added Custom** step.

We've now shaped our data to the extent we need to. Next let's connect to another data source, and combine data.

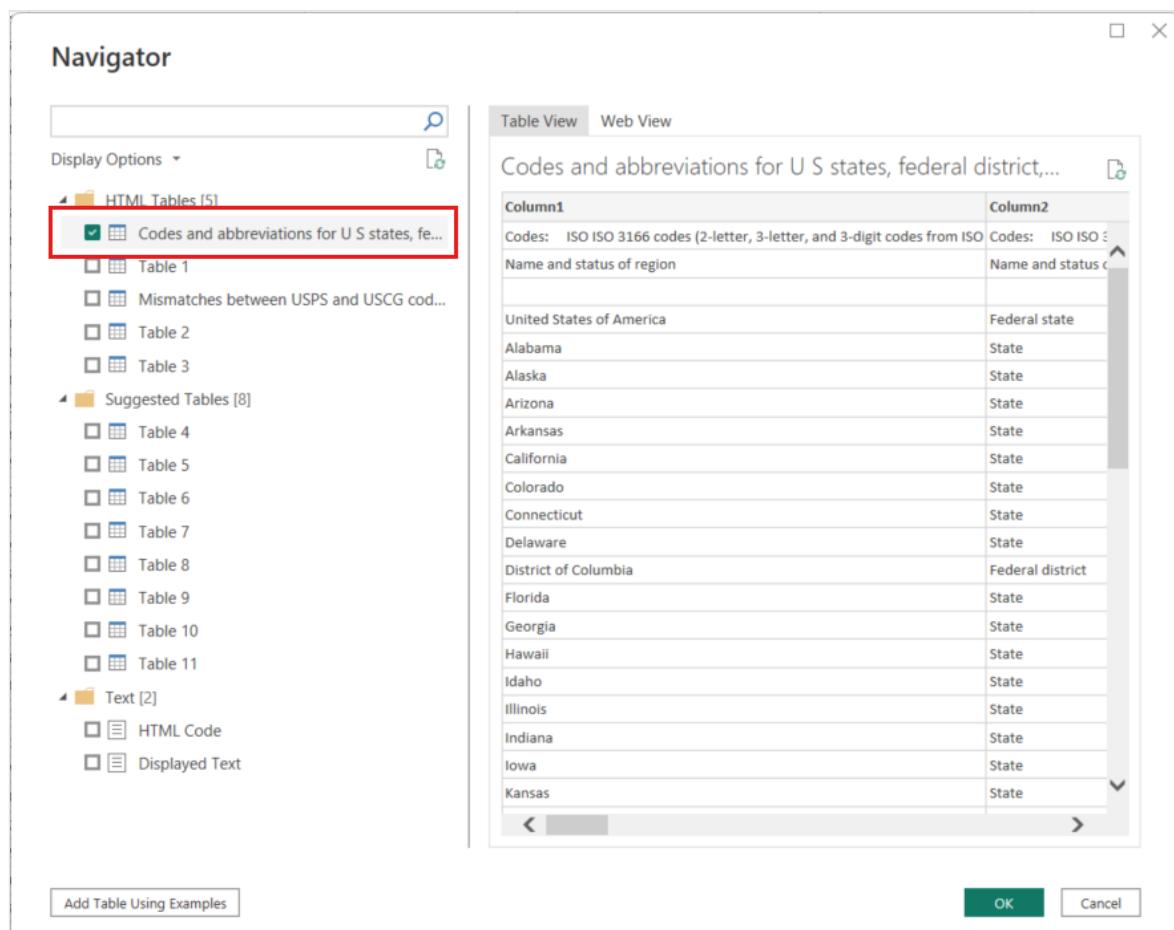
Combine data

The data about various states is interesting, and will be useful for building further analysis efforts and queries. However, most data about states uses a two-letter abbreviation for state codes, not the full name of the state. We need a way to associate state names with their abbreviations.

There's another public data source that does provides that association, but it needs a fair amount of shaping before we can connect it to our retirement table. To shape the data, follow these steps:

1. From the **Home** ribbon in Power Query Editor, select **New Source > Web**.
2. Enter the address of the website for state abbreviations, https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations, and then select **Connect**.

The Navigator displays the content of the website.



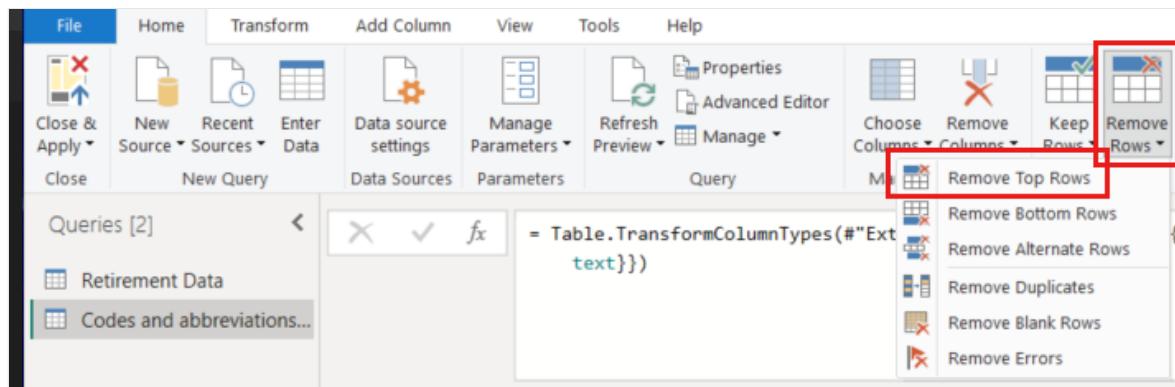
3. Select **Codes and abbreviations for U.S. states, federal district, territories, and other regions**.

Tip

It will take a bit of shaping to pare this table's data down to what we want. Is there a faster or easier way to accomplish the following steps? Yes, we could create a *relationship* between the two tables, and shape the data based on that relationship. The following example steps are helpful to learn for working with tables. However, relationships can help you quickly use data from multiple tables.

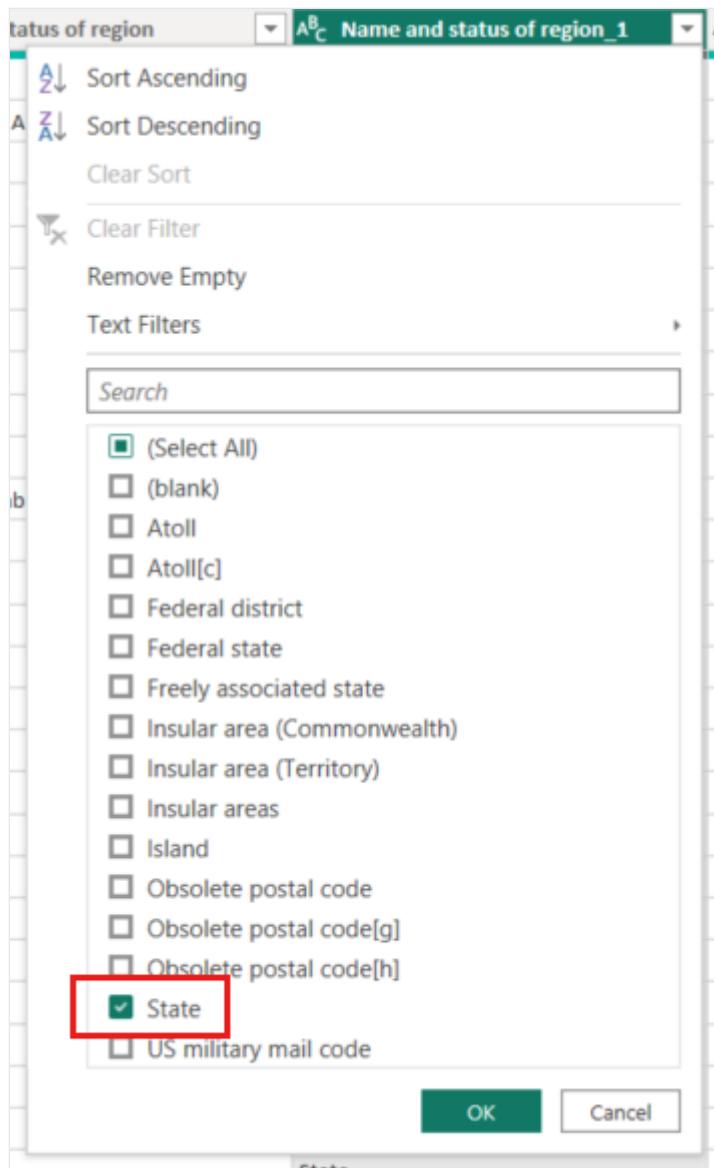
To get the data into shape, follow these steps:

1. Remove the top row. Because it's a result of the way that the web page's table was created, we don't need it. From the **Home** ribbon, select **Remove Rows > Remove Top Rows**.



The **Remove Top Rows** dialog appears. Specify 1 row to remove.

2. Promote the new top row to headers with **Use First Row As Headers** from the **Home** tab, or from the **Transform** tab in the ribbon.
3. Because the **Retirement Data** table doesn't have information for Washington DC or territories, we need to filter them from our list. Select the **Name and status of region_1** column's drop-down, then clear all checkboxes except **State**.



4. Remove all unneeded columns. Because we need only the mapping of each state to its official two-letter abbreviation (**Name and status of region** and **ANSI** columns), we can remove the other columns. First select the **Name and status of region** column, then hold down the **CTRL** key and select the **ANSI** column. From the **Home** tab on the ribbon, select **Remove Columns > Remove Other Columns**.

A_B_C_Name and status of region	A_B_C_Name and status of region_1	A_B_C_ISO	A_B_C_ANSI	A_B_C_ANSI_2	A_B_C_USPS	A_B_C_USCG
1 Alabama	State	US-AL	AL	01	AL	AL
2 Alaska	State	US-AK	AK	02	AK	AK
3 Arizona	State	US-AZ	AZ	04	AZ	AZ
4 Arkansas	State	US-AR	AR	05	AR	AR
5 California	State	US-CA	CA	06	CA	CF
Colorado	State	US-CO	CO	08	CO	CL

! Note

The *sequence* of applied steps in Power Query Editor is important, and affects how the data is shaped. It's also important to consider how one step might

impact another subsequent step. For example, if you remove a step from the applied steps, subsequent steps might not behave as originally intended.

① Note

When you resize the Power Query Editor window to make the width smaller, some ribbon items are condensed to make the best use of visible space.

When you increase the width of the Power Query Editor window, the ribbon items expand to make the most use of the increased ribbon area.

5. Rename the columns and the table. There are a few ways to rename a column: First select the column, then either select **Rename** from the **Transform** tab on the ribbon, or right-click and select **Rename**. The following image shows both options, but you only need to choose one.

The screenshot shows the Microsoft Power Query Editor interface. At the top, the ribbon has tabs for Transform, Add Column, View, Tools, and Help. Under the Transform tab, there are several icons: Transpose, Reverse Rows, Count Rows, Data Type: Text (with a dropdown), Replace Values (with a dropdown), Unpivot Columns (with a dropdown), Detect Data Type, Fill (with a dropdown), Move (with a dropdown), Pivot Column, Convert to List, and a 'More' button. A red box highlights the 'Rename' icon. Below the ribbon, a status bar says 'Any Column'. In the main area, there's a table titled 'Name and status of region' with columns labeled 'A' and 'B'. The table contains 23 rows of state names from Alabama to Minnesota. A context menu is open over the first row, with a red box highlighting the 'Rename...' option. Other options in the menu include Copy, Remove, Remove Other Columns, Duplicate Column, Add Column From Examples..., Remove Duplicates, Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., Split Column, Group By..., Fill, Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, Move, Drill Down, and Add as New Query.

	A	B
1	Alabama	Name and status of region
2	Alaska	A B ANSI
3	Arizona	
4	Arkansas	
5	California	
6	Colorado	
7	Connecticut	
8	Delaware	
9	Florida	
10	Georgia	
11	Hawaii	
12	Idaho	
13	Illinois	
14	Indiana	
15	Iowa	
16	Kansas	
17	Kentucky	
18	Louisiana	
19	Maine	Rename...
20	Maryland	
21	Massachusetts	
22	Michigan	
23	Minnesota	

6. Rename the columns to *State Name* and *State Code*. To rename the table, enter the **Name State Codes** in the **Query Settings** pane.

The screenshot shows the Power Query Editor interface. In the center, there's a preview of a table with two columns: 'State Name' and 'State Code'. The rows list US states from Alabama to Louisiana. To the right, the 'Query Settings' pane is open, showing the 'PROPERTIES' section with 'Name' set to 'State Codes' and the 'APPLIED STEPS' section which includes 'Renamed Columns'. At the bottom, it says 'PREVIEW DOWNLOADED AT 4:55 PM'.

Combine queries

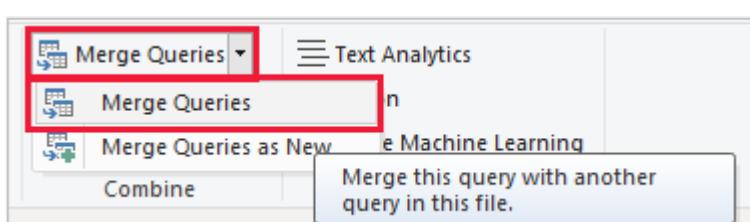
Now that we've shaped the *State Codes* table the way we want, let's combine these two tables, or queries, into one. Because the tables we now have are a result of the queries we applied to the data, they're often referred to as *queries*.

There are two primary ways of combining queries: *merging* and *appending*.

- For one or more *columns* that you'd like to add to another query, you *merge* the queries.
- For one or more *rows* of data that you'd like to add to an existing query, you *append* the query.

In this case, we want to merge the queries:

1. From the left pane of Power Query Editor, select the query *into which* you want the other query to merge. In this case, it's **Retirement Data**.
2. Select **Merge Queries > Merge Queries** from the **Home** tab on the ribbon.

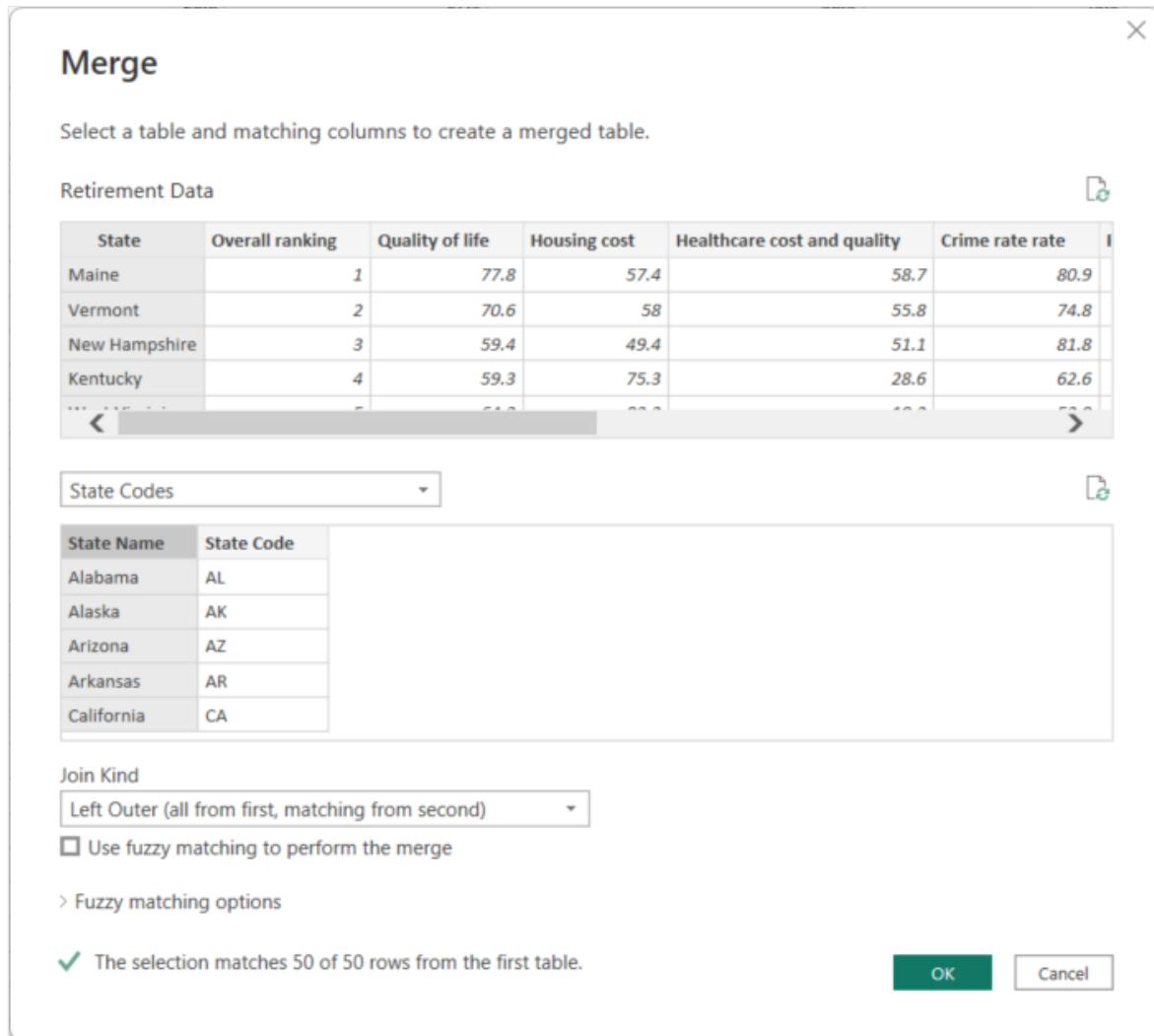


You might be prompted to set the privacy levels, to ensure the data is combined without including or transferring data you don't want transferred.

The **Merge** window appears. It prompts you to select which table you'd like merged into the selected table, and the matching columns to use for the merge.

3. Select **State** from the *Retirement Data* table, then select the **State Codes** query.

When you select a matching columns, the **OK** button is enabled.

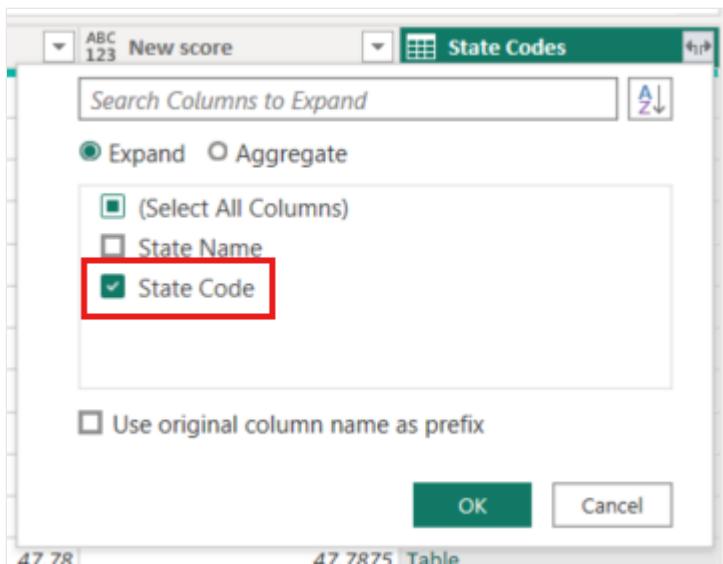


4. Select **OK**.

Power Query Editor creates a new column at the end of the query, which contains the contents of the table (query) that was merged with the existing query. All columns from the merged query are condensed into the column, but you can **Expand** the table and include whichever columns you want.

5. To expand the merged table, and select which columns to include, select the expand icon ().

The **Expand** window appears.



6. In this case, we want only the **State Code** column. Select that column, clear **Use original column name as prefix**, and then select **OK**.

If we had left the checkbox selected for **Use original column name as prefix**, the merged column would be named **State Codes.State Code**.

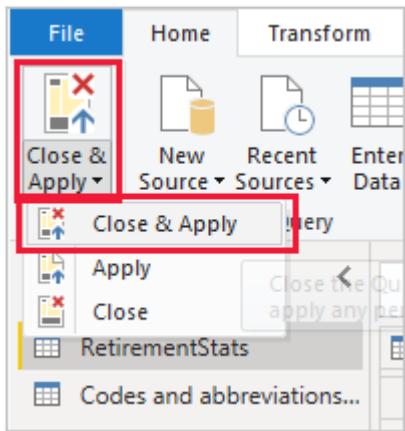
! **Note**

If you want to explore how to bring in the **State Codes** table, you can experiment a bit. If you don't like the results, just delete that step from the **APPLIED STEPS** list in the **Query Settings** pane, and your query returns to the state prior to applying that **Expand** step. You can do this as many times as you like until the expand process looks the way you want it.

We now have a single query (table) that combines two data sources, each of which was shaped to meet our needs. This query can be a basis for interesting data connections, such as housing cost statistics, quality of life, or crime rate in any state.

7. To apply your changes and close Power Query Editor, select **Close & Apply** from the **Home** ribbon tab.

The transformed dataset appears in Power BI Desktop, ready to be used for creating reports.



Next steps

For more information on Power BI Desktop and its capabilities, see the following resources:

- [What is Power BI Desktop?](#)
- [Query overview in Power BI Desktop](#)
- [Data sources in Power BI Desktop](#)
- [Connect to data in Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)

Tutorial: Analyze webpage data by using Power BI Desktop

Article • 01/12/2023

As a long-time soccer fan, you want to report on the UEFA European Championship (Euro Cup) winners over the years. With Power BI Desktop, you can import this data from a web page into a report and create visualizations that show the data. In this tutorial, you learn how to use Power BI Desktop to:

- Connect to a web data source and navigate across its available tables.
- Shape and transform data in the Power Query Editor.
- Name a query and import it into a Power BI Desktop report.
- Create and customize a map and a pie chart visualization.

Connect to a web data source

You can get the UEFA winners data from the Results table on the UEFA European Football Championship Wikipedia page at

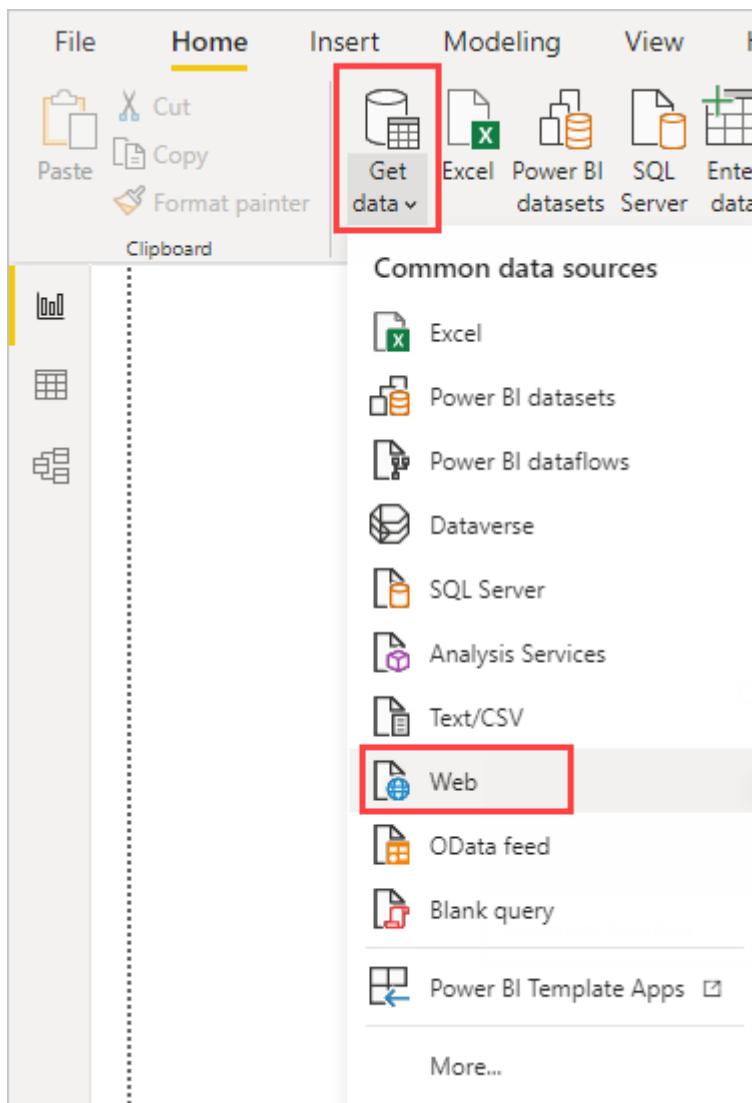
https://en.wikipedia.org/wiki/UEFA_European_Football_Championship.

Results		[edit]							
		See also: List of UEFA European Championship finals							
Year	Host	Final			Third place match			Number of teams	
		Winner	Score	Runner-up	Third place	Score	Fourth place		
1960 <small>Details</small>	France	Soviet Union	2–1 (a.e.t.)	Yugoslavia	Czechoslovakia	2–0	France	4	
1964 <small>Details</small>	Spain	Spain	2–1	Soviet Union	Hungary	3–1 (a.e.t.)	Denmark	4	
1968 <small>Details</small>	Italy	Italy	1–1 (a.e.t.) 2–0 (replay)	Yugoslavia	England	2–0	Soviet Union	4	
1972 <small>Details</small>	Belgium	West Germany	3–0	Soviet Union	Belgium	2–1	Hungary	4	
1976 <small>Details</small>	Yugoslavia	Czechoslovakia	2–2 (a.e.t.) (5–3 p)	West Germany	Netherlands	3–2 (a.e.t.)	Yugoslavia	4	
1980 <small>Details</small>	Italy	West Germany	2–1	Belgium	Czechoslovakia	1–1 ^[4] (9–8 p)	Italy	8	
Year		Final			Losing semi-finalists ^[5]			Number of teams	
1984 <small>Details</small>	France	France	2–0	Spain	Denmark and Portugal				
		Netherlands	2–0	Soviet Union	Italy and West Germany				
1988 <small>Details</small>	West Germany	Denmark	2–0	Germany	Netherlands and Sweden			8	
1992 <small>Details</small>	Sweden	Germany	2–1 (a.g.e.t.)	Czech Republic	England and France			16	
1996 <small>Details</small>	England	France	2–1 (a.g.e.t.)	Italy	Netherlands and Portugal			16	
2000 <small>Details</small>	Belgium & Netherlands	Greece	1–0	Portugal	Czech Republic and Netherlands			16	
2004 <small>Details</small>	Portugal	Spain	1–0	Germany	Russia and Turkey			16	
2008 <small>Details</small>	Austria & Switzerland	Spain	4–0	Italy	Germany and Portugal			16	
2012 <small>Details</small>	Poland & Ukraine	Portugal	1–0 (a.e.t.)	France	Germany and Wales			24	
2016 <small>Details</small>	France							24	
2020 <small>Details</small>	Pan-European							24	

Web connections are only established using basic authentication. Web sites requiring authentication might not work properly with the Web connector.

To import the data:

1. In the Power BI Desktop **Home** ribbon tab, drop down the arrow next to **Get Data**, and then select **Web**.



ⓘ Note

You can also select the **Get Data** item itself, or select **Get Data** from the Power BI Desktop get started dialog, then select **Web** from the **All** or **Other** section of the **Get Data** dialog, and then select **Connect**.

2. In the **From Web** dialog, paste the URL

https://en.wikipedia.org/wiki/UEFA_European_Football_Championship into the **URL** text box, and then select **OK**.



After you connect to the Wikipedia web page, the **Navigator** dialog shows a list of available tables on the page. You can select any of the table names to preview its data. The **Results[edit]** table has the data you want, although it's not exactly in the shape you want. You'll reshape and clean up the data before loading it into your report.

Column1	Column2	Column3	Column4	Column5
Year	Host		Final	Final
		null	null	Winners Score
1960Details	France		null	Soviet Union 2–1 (a.)
1964Details	Spain		null	Spain 2–1
1968Details	Italy		null	Italy 1–1 (a.)
1972Details	Belgium		null	West Germany 3–0
1976Details	Yugoslavia		null	Czechoslovakia 2–2 (a.)
1980Details	Italy		null	West Germany 2–1
Year	Host(s)		Final	Final
		null	null	Winner Score
1984Details	France		null	France 2–0
1988Details	West Germany		null	Netherlands 2–0
1992Details	Sweden		null	Denmark 2–0
1996Details	England		null	Germany 2–1 (g.)
2000Details	Belgium Netherlands		null	France 2–1 (g.)
2004Details	Portugal		null	Greece 1–0
2008Details	Austria Switzerland		null	Spain 1–0
2012Details	Poland Ukraine		null	Spain 4–0
2016Details	France		null	Portugal 1–0 (a.)
2020[c]Details	Pan-European		null	
2024Details	Germany		null	

➊ Note

The Preview pane shows the most recent table selected, but all selected tables load into the Power Query Editor when you select **Transform Data** or **Load**.

3. Select the **Results[edit]** table in the Navigator list, and then select **Transform Data**.

A preview of the table opens in **Power Query Editor**, where you can apply transformations to clean up the data.

The screenshot shows the Power Query Editor interface with a table containing historical data. The table has 12 columns and 21 rows. The columns are labeled Column1 through Column5. The 'APPLIED STEPS' pane on the right shows a step named 'Changed Type'. The status bar at the bottom indicates 'PREVIEW DOWNLOADED AT 8:26 AM'.

Shape data in Power Query Editor

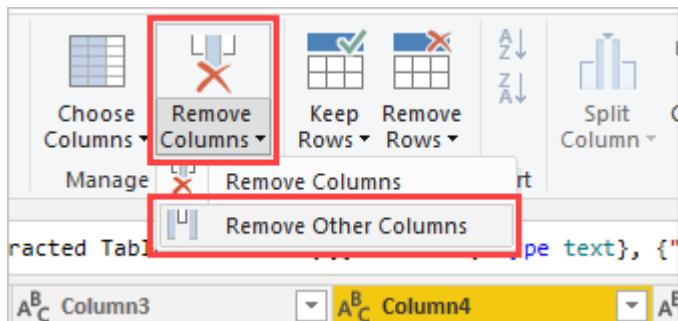
You want to make the data easier to scan by displaying only the years and the countries/regions that won. You can use the Power Query Editor to perform these data shaping and cleansing steps.

First, remove all the columns except for two from the table. Rename these columns as *Year* and *CountryRegion* later in the process.

1. In the **Power Query Editor** grid, select the columns. Select **Ctrl** to select multiple items.
2. Right-click and select **Remove Other Columns**, or select **Remove Columns > Remove Other Columns** from the **Manage Columns** group in the **Home** ribbon tab, to remove all other columns from the table.

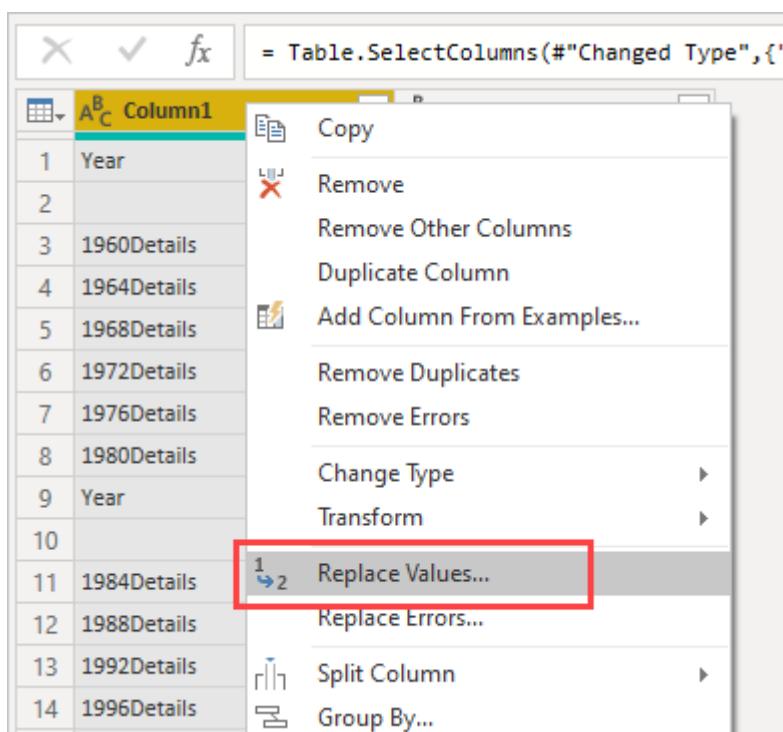
The screenshot shows the Power Query Editor with a context menu open over the table. The 'Remove Other Columns' option is highlighted. The menu also includes options like Copy, Remove Columns, Add Column From Examples..., Remove Duplicates, Remove Errors, Replace Values..., Fill, Change Type, Transform, and Merge Columns.

or

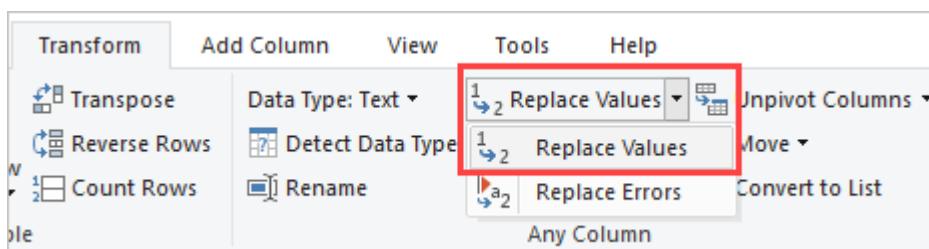


This version of the imported data has the word *Details* appended to the year. You can remove the extra word *Details* from the first column cells.

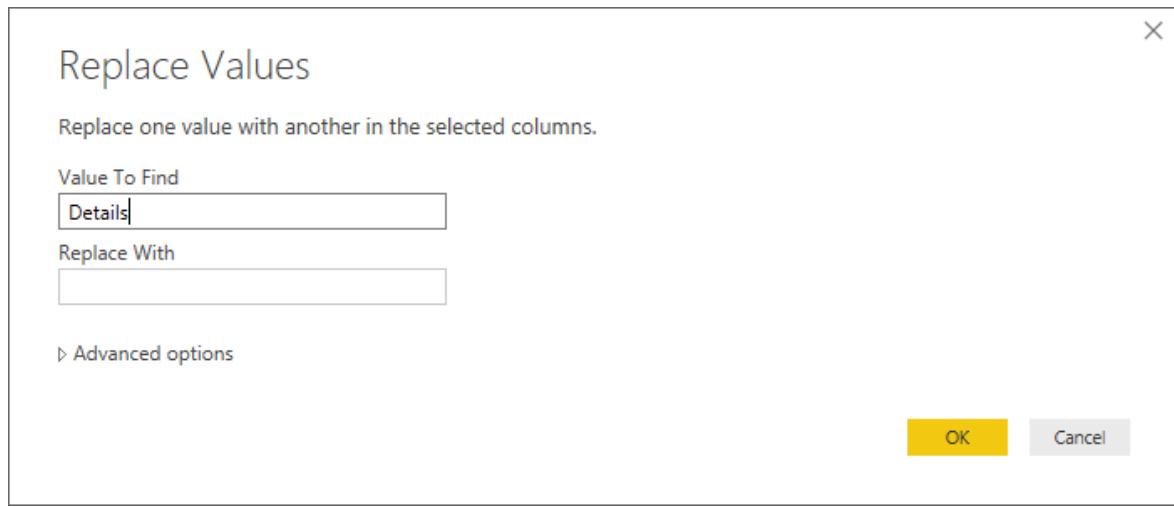
1. Select the first column.
2. Right-click, and select **Replace Values**, or select **Replace Values** from the **Transform** group in the **Home** tab of the ribbon. This option is also found in the **Any Column** group in the **Transform** tab.



or

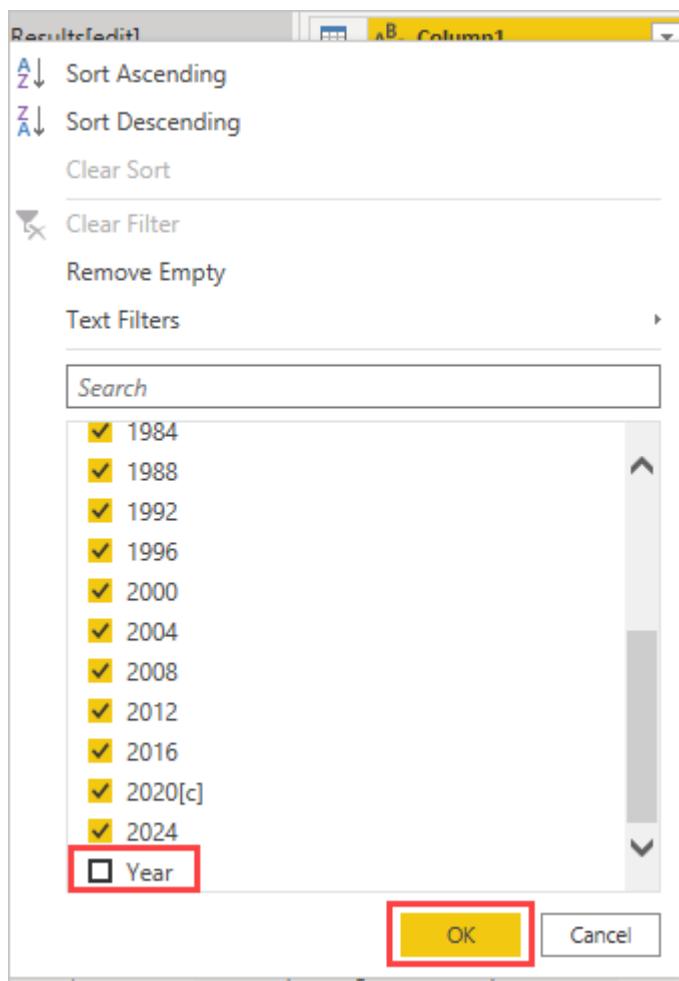


3. In the **Replace Values** dialog, type **Details** in the **Value To Find** text box, leave the **Replace With** text box empty, and then select **OK** to delete the word *Details* from this column.



Some cells contain only the word "Year" rather than year values. You can filter the column to only display rows that don't contain the word "Year".

1. Select the filter drop-down arrow on the column.
2. In the drop-down menu, scroll down and clear the checkbox next to the **Year** option, and then select **OK**.



Since you're only looking at the final winners data now, you can rename the second column to **CountryRegion**. To rename the column:

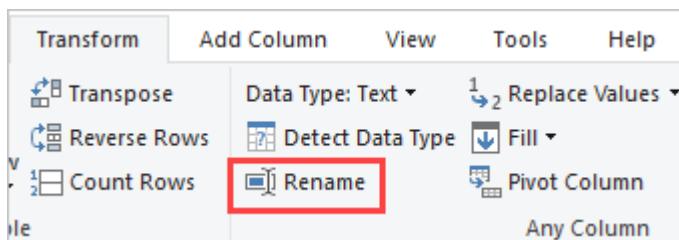
1. Double-click or tap and hold in the second column header, or

- Right-click the column header, and select **Rename**, or
- Select the *column and select **Rename** from the **Any Column** group in the **Transform** tab of the ribbon.

The screenshot shows a data grid with two columns: 'Column1' and 'Column4'. The 'Column1' header is selected. A context menu is open, listing options like Copy, Remove, Duplicate Column, etc., with 'Rename...' highlighted by a red box. The 'Transform' tab is selected in the ribbon.

Column1	Column4
1960	Soviet Union
1964	Spain
1968	Italy
1972	West Germany
1976	Czechoslovakia
1980	West Germany
1984	France
1988	Netherlands
1992	Denmark
1996	Germany
2000	France
2004	Greece
2008	Spain
2012	Spain
2016	Portugal
2020[c]	
2024	

or



2. Type **CountryRegion** in the header and press **Enter** to rename the column.

You also want to filter out rows that have `null` values in the **CountryRegion** column.

You could use the filter menu as you did with the **Year** values, or you can:

1. Right-click on the **CountryRegion** cell in the 2020 row, which has the value `null`.
2. Select **Text Filters > Does not Equal** in the context menu to remove any rows that contain that cell's value.

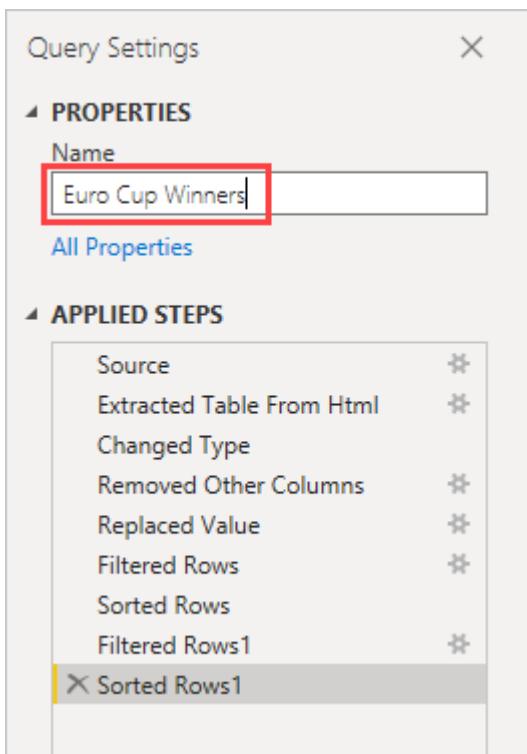
A screenshot of a Power BI interface showing a table with two columns: 'Column1' and 'Country'. The 'Country' column contains a list of countries. A context menu is open over the 'Country' column, specifically over the cell 'Spain'. The menu has several options: 'Copy', 'Text Filters', 'Replace Values...', 'Drill Down', and 'Add as New Query'. The 'Text Filters' option is highlighted with a red box. A secondary menu for 'Text Filters' is open, listing various comparison operators: 'Equals', 'Does Not Equal', 'Begins With', 'Does Not Begin With', 'Ends With', 'Does Not End With', 'Contains', and 'Does Not Contain'. The 'Does Not Equal' option is also highlighted with a red box.

	Column1	Country
1	2024	Portugal
2	2020[c]	Spain
3	2016	Spain
4	2012	Spain
5	2008	Spain
6	2004	Greece
7	2000	France
8	1996	Germany
9	1992	Denmark
10	1988	Netherlands
11	1984	France
12	1980	West Germany
13	1976	Germany

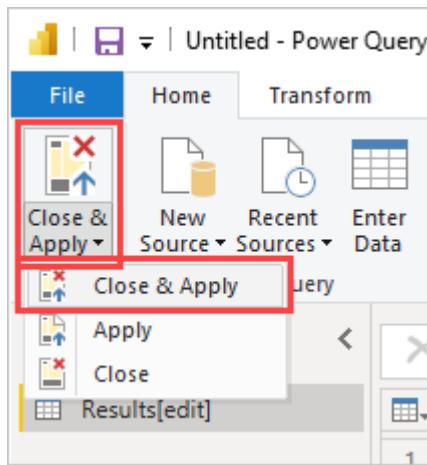
Import the query into Report View

Now that you've shaped the data the way you want, you're ready to name your query "Euro Cup Winners" and import it into your report.

1. In the **Query Settings** pane, in the **Name** text box, enter **Euro Cup Winners**.



2. Select **Close & Apply** > **Close & Apply** from the **Home** tab of the ribbon.



The query loads into the Power BI Desktop *Report* view, where you can see it in the **Fields** pane.

Field	Type
Country	Text
Year	Text

Tip

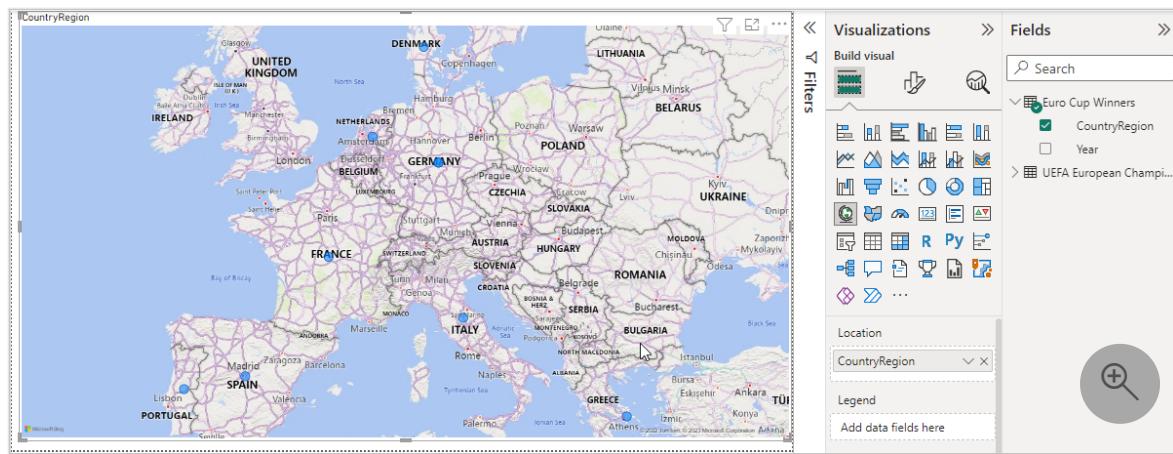
You can always get back to the Power Query Editor to edit and refine your query by:

- Selecting the **More options** ellipsis (...) next to **Euro Cup Winners** in the **Fields** pane, and selecting **Edit query**, or
- Selecting **Transform data** in the **Queries** group of the **Home** ribbon tab in Report view.

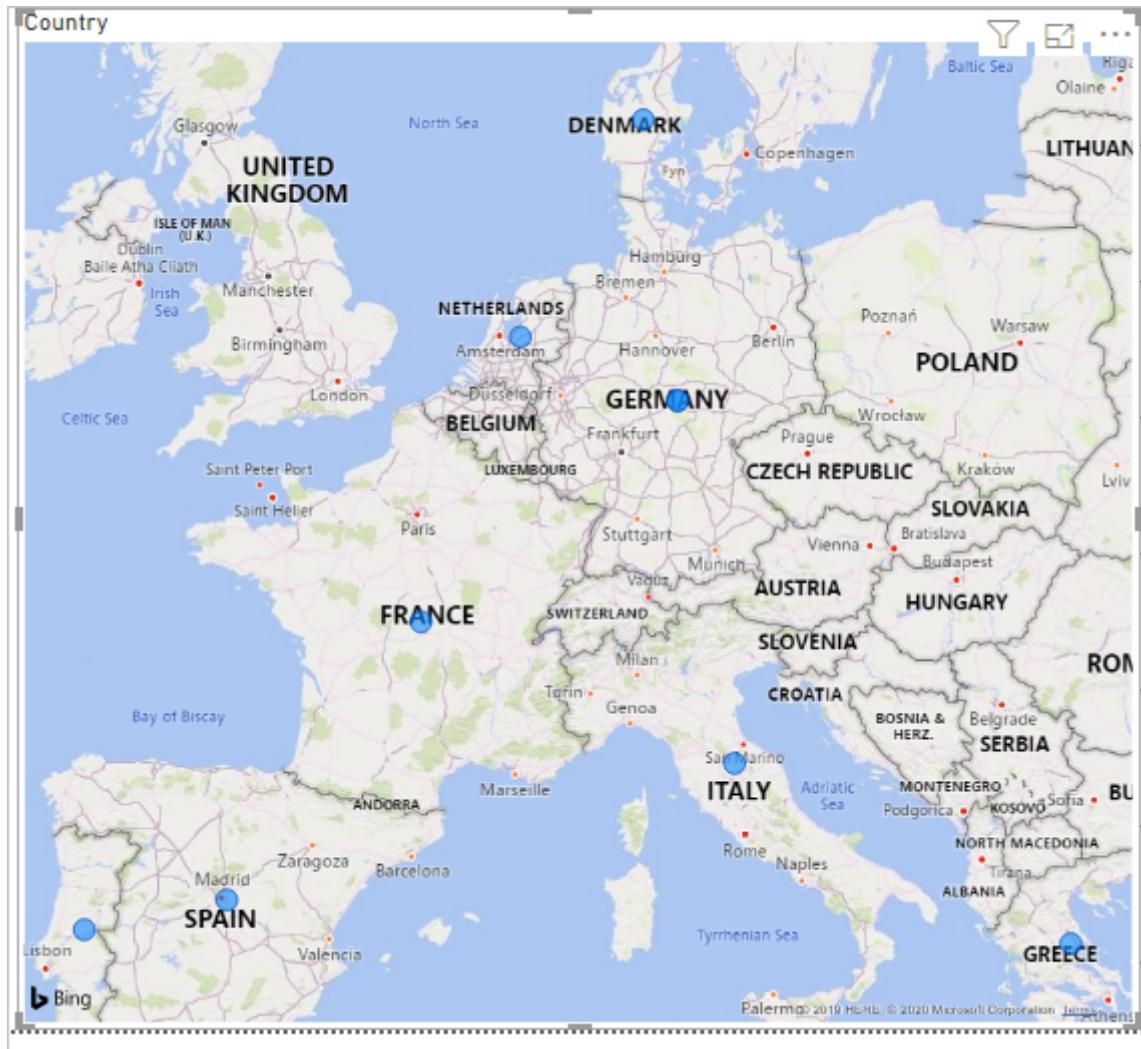
Create a visualization

To create a visualization based on your data:

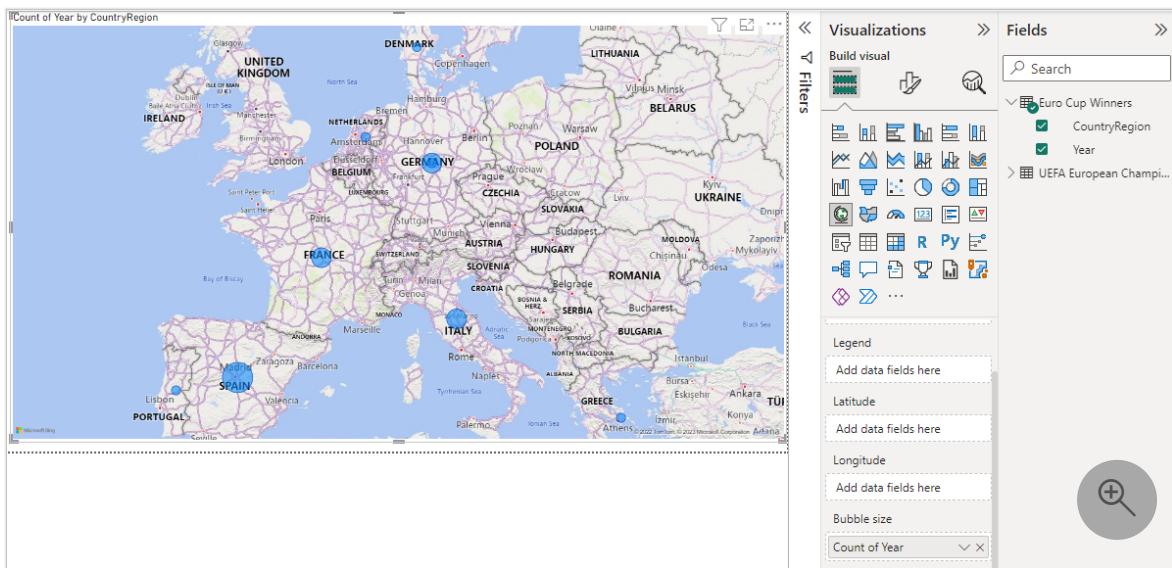
1. Select the **CountryRegion** field in the **Fields** pane, or drag it to the report canvas. Power BI Desktop recognizes the data as country/region names, and automatically creates a **Map** visualization.



2. Enlarge the map by dragging the handles in the corners so all the winning country/region names are visible.



3. The map shows identical data points for every country/region that won a Euro Cup tournament. To make the size of each data point reflect how often the country/region has won, drag the Year field to Drag data fields here under Bubble size in the lower part of the Visualizations pane. The field automatically changes to a Count of Year measure, and the map visualization now shows larger data points for countries/regions that have won more tournaments.



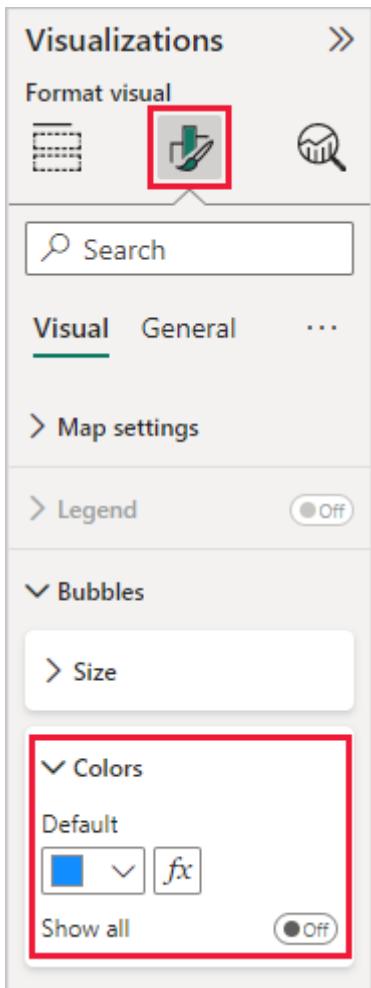
Customize the visualization

As you can see, it's very easy to create visualizations based on your data. It's also easy to customize your visualizations to better present the data in ways that you want.

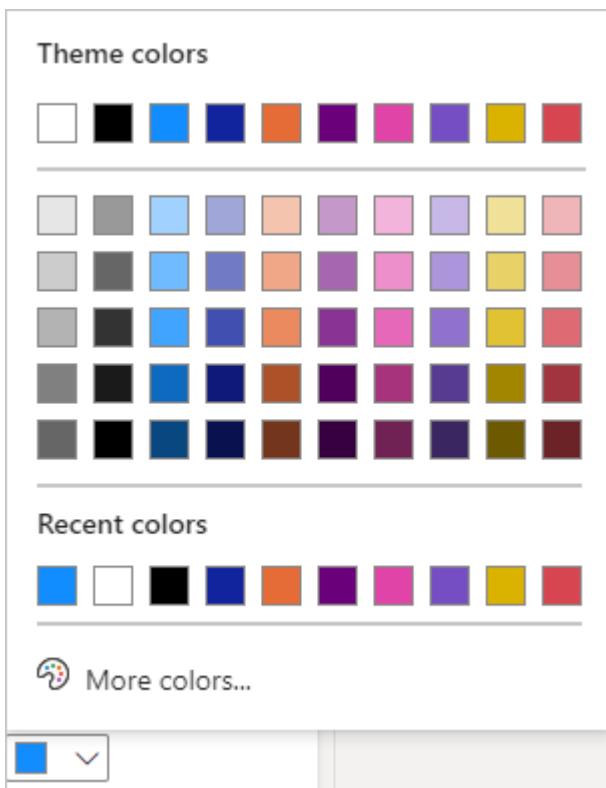
Format the map

You can change the appearance of a visualization by selecting it and then selecting the **Format** (paint brush) icon in the **Visualizations** pane. For example, the "Germany" data points in your visualization could be misleading, because West Germany won two tournaments and Germany won one. The map superimposes the two points rather than separating or adding them together. You can color these two points differently to highlight this fact. You can also give the map a more descriptive and attractive title.

1. With the visualization selected, select the **Format** icon, and then select **Visual > Bubbles > Colors** to expand the data color options.

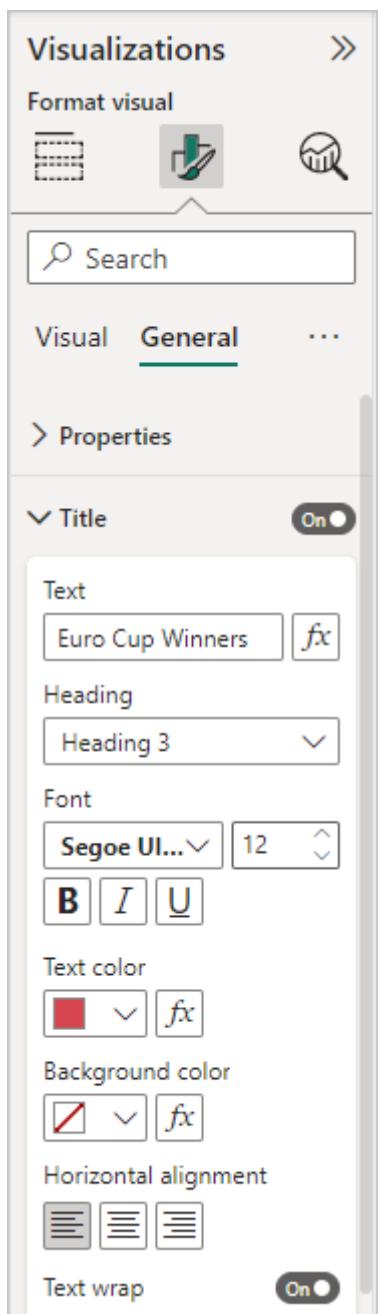


2. Turn **Show all** to **On**, and then select the drop-down menu next to **West Germany** and choose a yellow color.

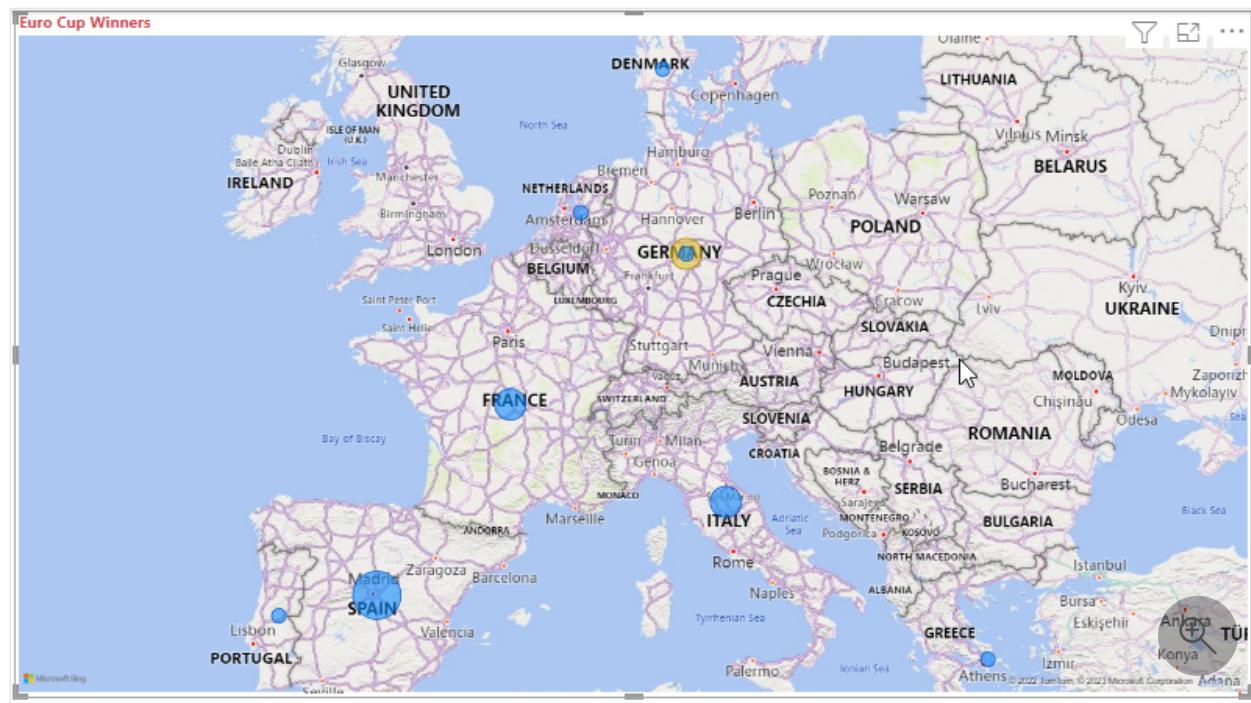


3. Select **General > Title** to expand the title options, and in the **Text** field, type **Euro Cup Winners** in place of the current title.

4. Change Text color to red, size to 12, and Font to Segoe UI (Bold).



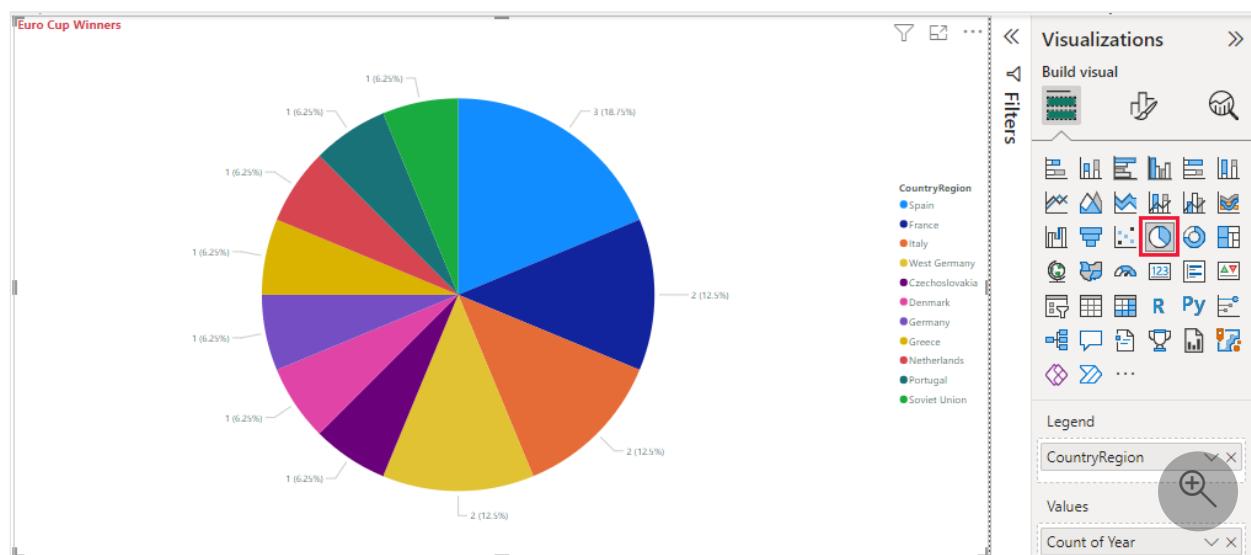
Your map visualization now looks like this example:



Change the visualization type

You can change the type of a visualization by selecting it and then selecting a different icon at the top of the **Visualizations** pane. For example, your map visualization is missing the data for the Soviet Union, because that country/region no longer exists on the world map. Another type of visualization like a *treemap* or *pie chart* might be more accurate, because it shows all the values.

To change the map to a pie chart, select the map and then choose the **Pie chart** icon in the **Visualizations** pane.



Tip

- You can use the **Data colors** formatting options to make "Germany" and "West Germany" the same color.
- To group the countries/regions with the most wins together on the pie chart, select the ellipsis (...) at the upper right of the visualization, and then select **Sort by Count of Year**.

Power BI Desktop provides a seamless end-to-end experience, from getting data from a wide range of data sources and shaping it to meet your analysis needs, to visualizing this data in rich and interactive ways. Once your report is ready, you can [upload it to Power BI](#) and create dashboards based on it, which you can share with other Power BI users.

See also

- [Microsoft Learn training for Power BI](#)
- [Watch Power BI videos](#)
- [Visit the Power BI Forum ↗](#)
- [Read the Power BI Blog ↗](#)

Tutorial: Analyze sales data from Excel and an OData feed

Article • 02/15/2023

It's common to have data in multiple data sources. For example, you could have two databases, one for product information, and another for sales information. With *Power BI Desktop*, you can combine data from different sources to create interesting, compelling data analyses and visualizations.

In this tutorial, you combine data from two data sources:

- An Excel workbook with product information
- An OData feed containing orders data

You're going to import each dataset and do transformation and aggregation operations. Then, you can use the two source's data to produce a sales analysis report with interactive visualizations. Later, apply these techniques to SQL Server queries, CSV files, and other data sources in Power BI Desktop.

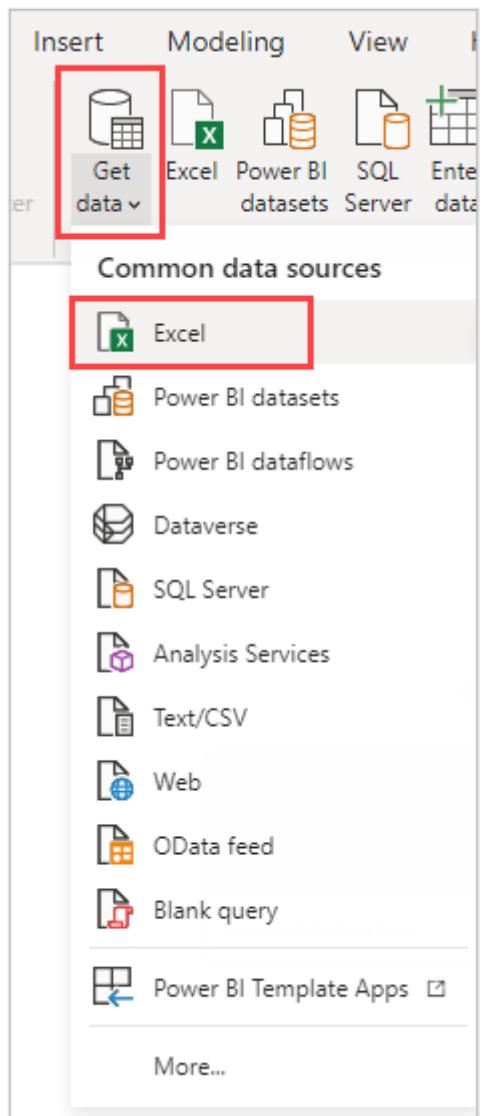
Note

In Power BI Desktop, there are often a few ways to accomplish a task. For example, you can right-click or use a **More options** menu on a column or cell to see more ribbon selections. Several alternate methods are described in the following steps.

Import Excel product data

First, import product data from the *Products.xlsx* Excel workbook into Power BI Desktop.

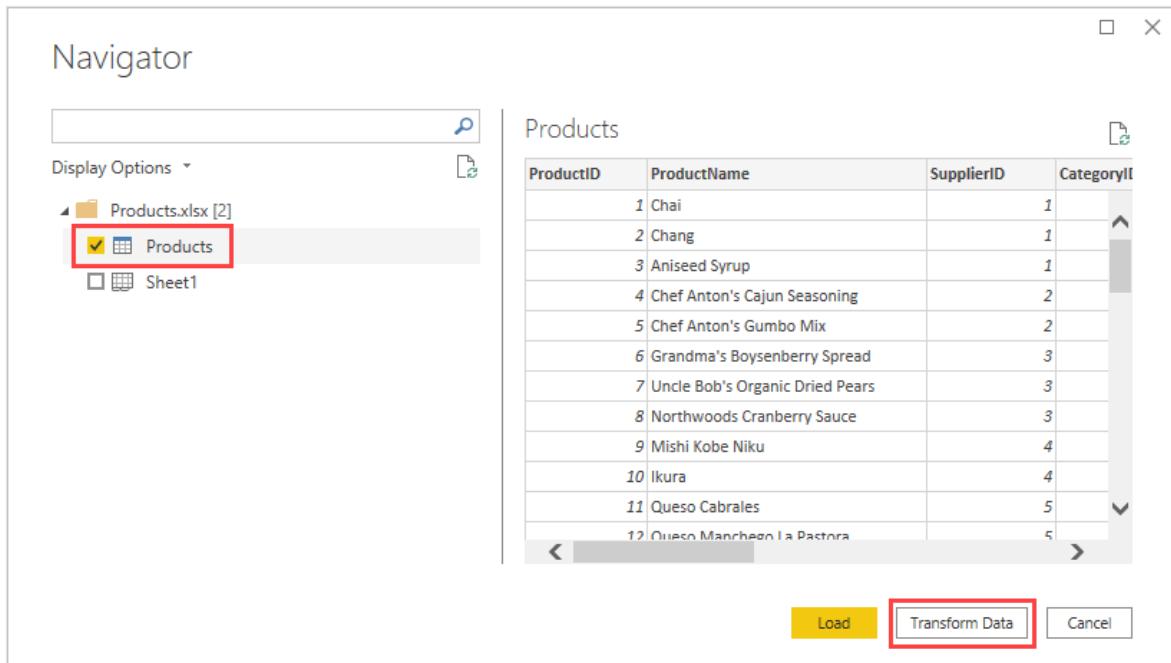
1. [Download the Products.xlsx Excel workbook](#) and save it as *Products.xlsx*.
2. Select the arrow next to **Get data** in the Power BI Desktop ribbon's **Home** tab, and then select **Excel** from the **Common data sources** menu.



Note

You can also select the **Get data** item itself, or select **Get data** from the Power BI **Get started** dialog box, then select **Excel** or **File > Excel** in the **Get Data** dialog box, and then select **Connect**.

3. In the **Open** dialog box, navigate to and select the **Products.xlsx** file, and then select **Open**.
4. In the **Navigator**, select the **Products** table and then select **Transform Data**.



A table preview opens in the Power Query Editor, where you can apply transformations to clean up the data.

! Note

You can also open the Power Query Editor by selecting **Transform data** from the **Home** ribbon in Power BI Desktop, or by right-clicking or choosing **More options** next to any query in the **Report** view, and selecting **Transform data**.

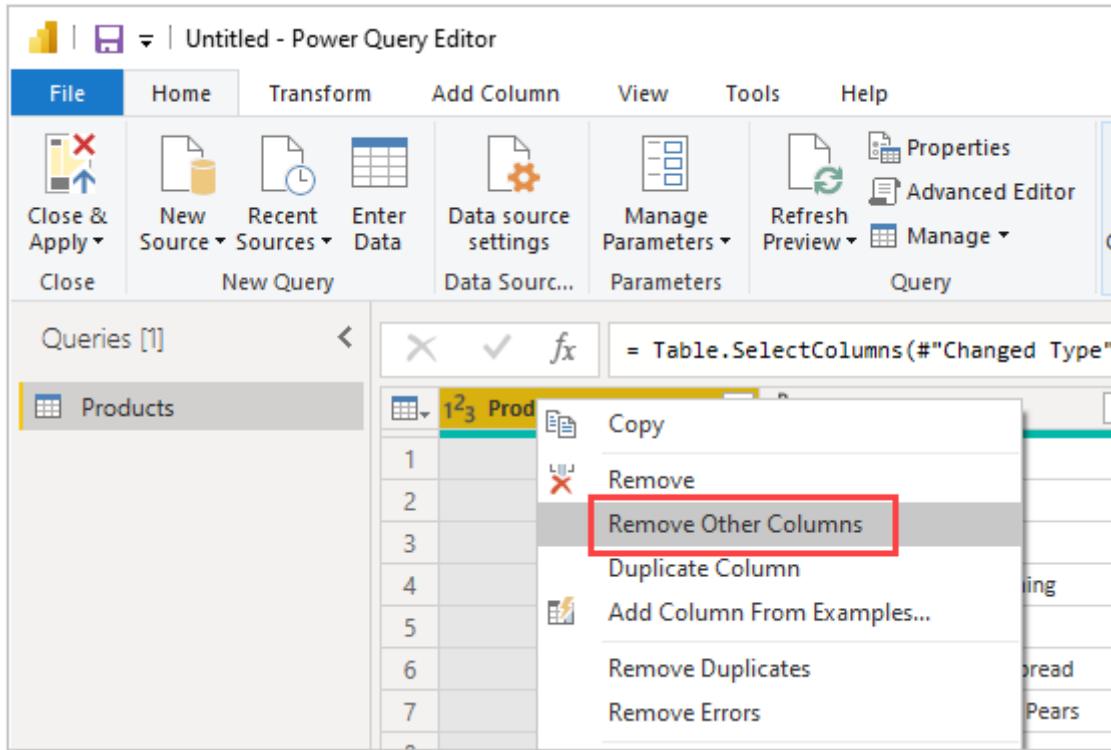
Clean up the columns

Your combined report uses the Excel workbook's **ProductID**, **ProductName**, **QuantityPerUnit**, and **UnitsInStock** columns. You can remove the other columns.

1. In Power Query Editor, select the **ProductID**, **ProductName**, **QuantityPerUnit**, and **UnitsInStock** columns. You can use Ctrl to select more than one column, or Shift to

select columns next to each other.

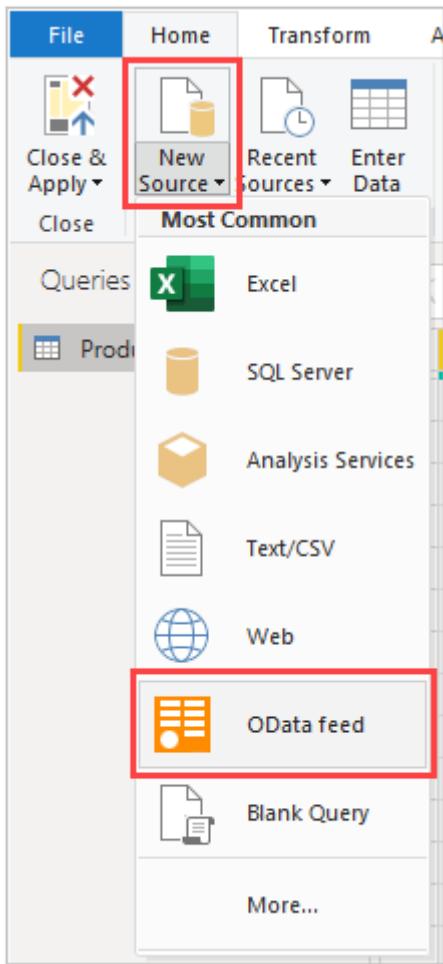
2. Right-click any of the selected headers. Select **Remove Other Columns** from the dropdown menu. You can also select **Remove Columns > Remove Other Columns** from the **Manage Columns** group in the **Home** ribbon tab.



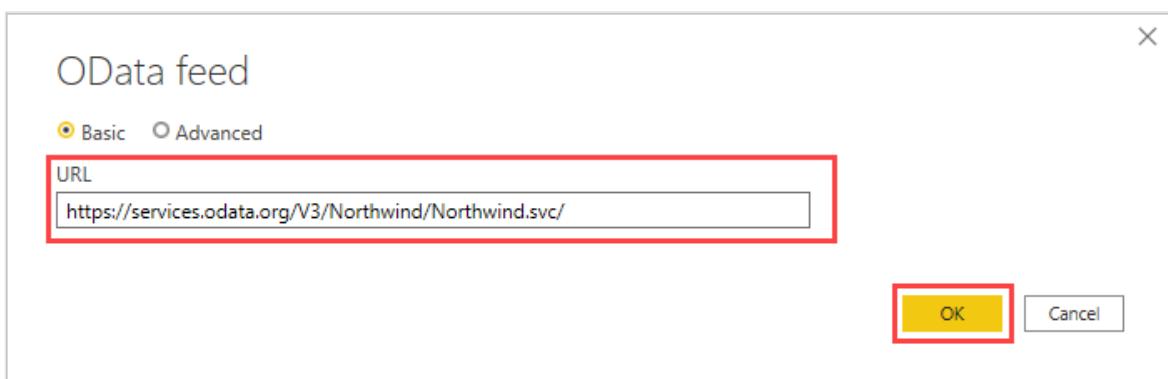
Import the OData feed's order data

Next, import the order data from the sample Northwind sales system OData feed.

1. In Power Query Editor, select **New Source** and then, from the **Most Common** menu, select **OData feed**.



2. In the **OData feed** dialog box, paste the Northwind OData feed URL,
`https://services.odata.org/V3/Northwind/Northwind.svc/`. Select **OK**.



3. In **Navigator**, select the **Orders** table, and then select **OK** to load the data into Power Query Editor.

The screenshot shows the Power BI Navigator window. On the left, a tree view lists various tables from the Northwind OData feed. The 'Orders' table is selected, indicated by a red box around its icon and name. On the right, a preview of the 'Orders' table is displayed in a grid format. The table has columns: OrderID, CustomerID, EmployeeID, OrderDate, and RequiredDate. The data shows several rows of order information. At the bottom right of the preview area, there is an 'OK' button highlighted with a red box.

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/199
10249	TOMSP	6	7/5/1996 12:00:00 AM	8/16/199
10250	HANAR	4	7/8/1996 12:00:00 AM	8/5/199
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/199
10252	SUPRD	4	7/9/1996 12:00:00 AM	8/6/199
10253	HANAR	3	7/10/1996 12:00:00 AM	7/24/199
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/199
10255	RICSU	9	7/12/1996 12:00:00 AM	8/9/199
10256	WELLI	3	7/15/1996 12:00:00 AM	8/12/199
10257	HILAA	4	7/16/1996 12:00:00 AM	8/13/199
10258	ERNSH	1	7/17/1996 12:00:00 AM	8/14/199
10259	CENTC	4	7/18/1996 12:00:00 AM	8/15/199
10260	OTTIK	4	7/19/1996 12:00:00 AM	8/16/199
10261	QUEDE	4	7/19/1996 12:00:00 AM	8/16/199
10262	RATTC	8	7/22/1996 12:00:00 AM	8/19/199
10263	ERNSH	9	7/23/1996 12:00:00 AM	8/20/199
10264	FOLKO	6	7/24/1996 12:00:00 AM	8/21/199
10265	BLONP	2	7/25/1996 12:00:00 AM	8/22/199
10266	WARTH	3	7/26/1996 12:00:00 AM	9/6/199
10267	FRANK	4	7/29/1996 12:00:00 AM	8/26/199
10268	GROSR	8	7/30/1996 12:00:00 AM	8/27/199
10269	WHITC	5	7/31/1996 12:00:00 AM	8/14/199

ⓘ Note

In Navigator, you can select any table name, without selecting the checkbox, to see a preview.

Expand the order data

You can use table references to build queries when connecting to data sources with multiple tables, such as relational databases or the Northwind OData feed. The **Orders** table contains references to several related tables. You can use the expand operation to add the **ProductID**, **UnitPrice**, and **Quantity** columns from the related **Order_Details** table into the subject (**Orders**) table.

1. Scroll to the right in the **Orders** table until you see the **Order_Details** column. It contains references to another table and not data.

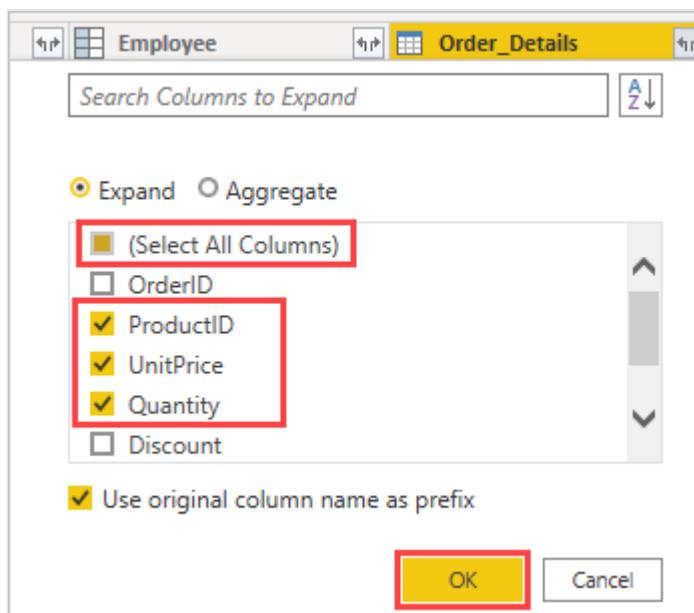
The screenshot shows the Power Query Editor interface. The 'File' tab is selected. In the 'Queries [2]' pane, 'Orders' is selected. The main preview area shows a table with three columns: 'Employee' (Table), 'Order_Details' (Table), and 'Shipper' (Table). The 'Order_Details' column is highlighted with a red box. The 'Applied Steps' pane on the right shows a single step named 'Navigation'. The status bar at the bottom indicates '18 COLUMNS, 830 ROWS' and 'Column profiling based on top 1000 rows'.

2. Select the **Expand icon** (↗) in the **Order_Details** column header.

3. In the dropdown menu:

a. Select **(Select All Columns)** to clear all columns.

b. Select **ProductID**, **UnitPrice**, and **Quantity**, and then select **OK**.



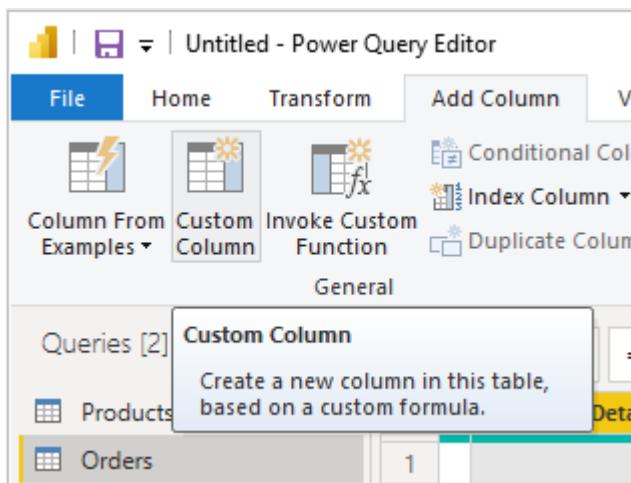
After you expand the **Order_Details** table, three new nested table columns replace the **Order_Details** column. There are new rows in the table for each order's added data.

	1.2 Order_Details.ProductID	1.2 Order_Details.UnitPrice	1.2 Order_Details.Quantity
1		11	14
2		42	9.8
3		72	34.8
4		14	18.6
5		51	42.4
6		41	7.7
7		51	42.4
8		65	16.8
9		22	16.8
10		57	15.6
11		65	16.8
12		20	16.8

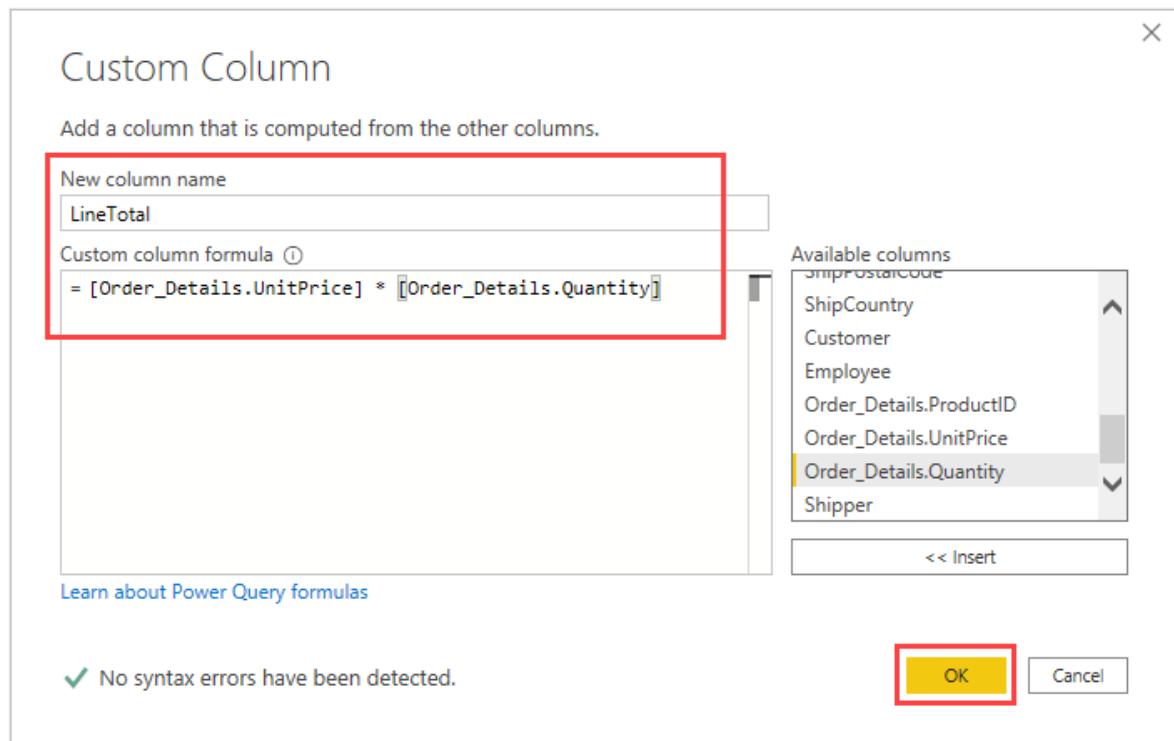
Create a custom calculated column

Power Query Editor lets you create calculations and custom fields to enrich your data. You can create a custom column that multiplies the unit price by item quantity to calculate the total price for each order's line item.

1. In the Power Query Editor's Add Column ribbon tab, select **Custom Column**.



2. In the **Custom Column** dialog box, type **LineTotal** in the **New column name** field.
3. In the **Custom column formula** field after the **=**, enter **[Order_Details.UnitPrice] * [Order_Details.Quantity]**. You can also select the field names from the **Available columns** scroll box and select **<< Insert**, instead of typing them.
4. Select **OK**.

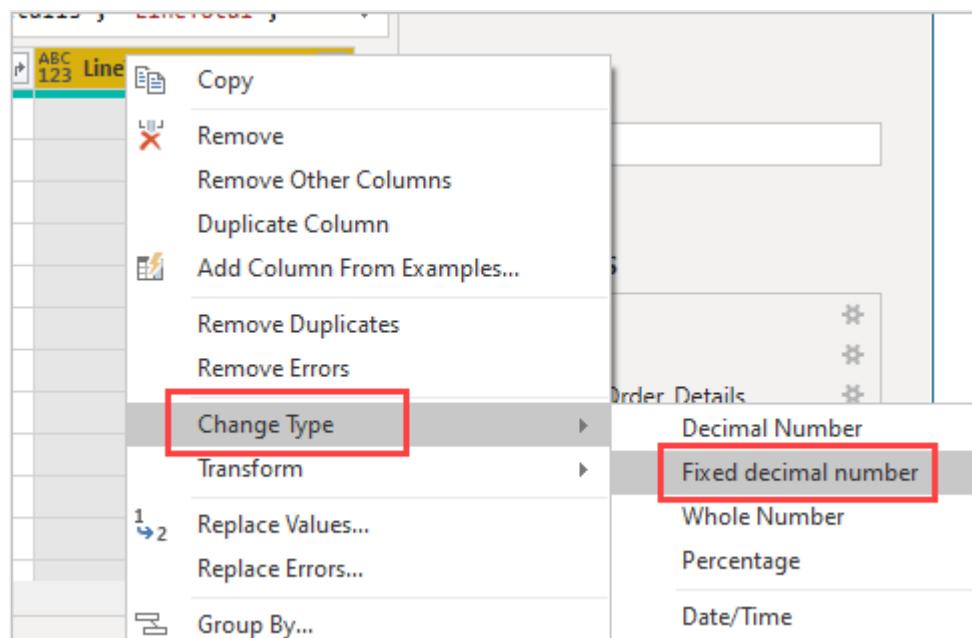


The new **LineTotal** field appears as the last column in the **Orders** table.

Set the new field's data type

When Power Query Editor connects to data, it makes a best guess as to each field's data type for display purposes. A header icon indicates each field's assigned data type. You can also look under **Data Type** in the **Home** ribbon tab's **Transform** group.

Your new **LineTotal** column has an **Any** data type, but it has currency values. To assign a data type, right-click the **LineTotal** column header, select **Change Type** from the dropdown menu, and then select **Fixed decimal number**.



Note

You can also select the **LineTotal** column, then select the arrow next to **Data Type** in the **Transform** area of the **Home** ribbon tab, and then select **Fixed decimal number**.

Clean up the orders columns

To make your model easier to work with in reports, you can delete, rename, and reorder some columns.

Your report is going to use the following columns:

- **OrderDate**
- **ShipCity**
- **ShipCountry**
- **Order_Details.ProductID**
- **Order_Details.UnitPrice**
- **Order_Details.Quantity**
- **LineTotal**

Select these columns and use **Remove Other Columns** as you did with the Excel data. Or, you can select the non-listed columns, right-click on one of them, and select **Remove Columns**.

You can rename the columns prefixed with "**Order_Details.**" to make them easier to read:

1. Double-click or tap and hold each column header, or right-click the column header, and select **Rename** from the dropdown menu.
2. Delete the **Order_Details.** prefix from each name.

Finally, to make the **LineTotal** column easier to access, drag and drop it to the left, just to the right of the **ShipCountry** column.

	A ^B _C ShipCountry	\$ LineTotal	1 ² ₃ ProductID
1	France	168.00	1
2	France	98.00	4
3	France	174.00	7
4	Germany	167.40	1
5	Germany	1,696.00	5
6	Brazil	77.00	4
7	Brazil	1,484.00	5
8	Brazil	252.00	6
9	France	100.80	2
10	France	234.00	5
11	France	336.00	6
12	Pakistan	2,502.00	

Review the query steps

Your Power Query Editor actions to shape and transform data are recorded. Each action appears on the right in the **Query Settings** pane under **APPLIED STEPS**. You can step back through the **APPLIED STEPS** to review your steps, and edit, delete, or rearrange them if necessary. However, changing preceding steps is risky as that can break later steps.

Select each of your queries in the **Queries** list on the left side of Power Query Editor, and review the **APPLIED STEPS** in **Query Settings**. After you apply the previous data transformations, the **APPLIED STEPS** for your two queries should look like this:

Products query

Query Settings

PROPERTIES

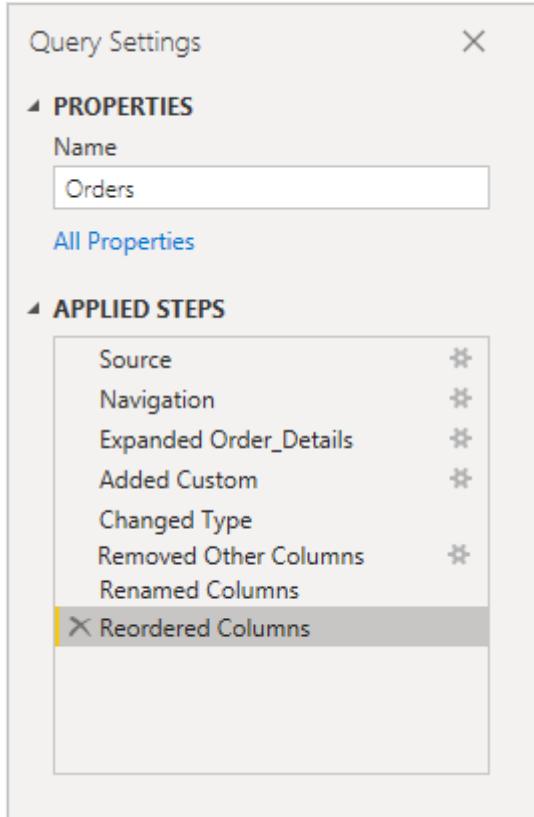
Name: Products

[All Properties](#)

APPLIED STEPS

- Source
- Navigation
- Changed Type

Orders query

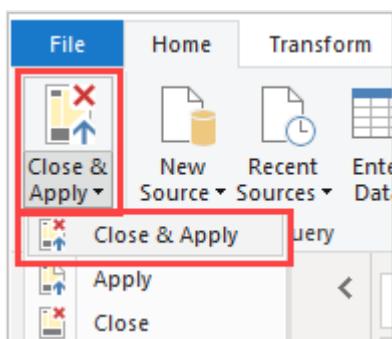


Tip

Underlying the applied steps are formulas written in the *Power Query Language*, also known as the **M language**. To see and edit the formulas, select **Advanced Editor** in the **Query** group of the **Home** tab of the ribbon.

Import the transformed queries

When you're satisfied with your transformed data and ready to import it into Power BI Desktop **Report** view, select **Close & Apply** > **Close & Apply** in the **Home** ribbon tab's **Close** group.



Once the data is loaded, the queries appear in the **Fields** list in the Power BI Desktop **Report** view.

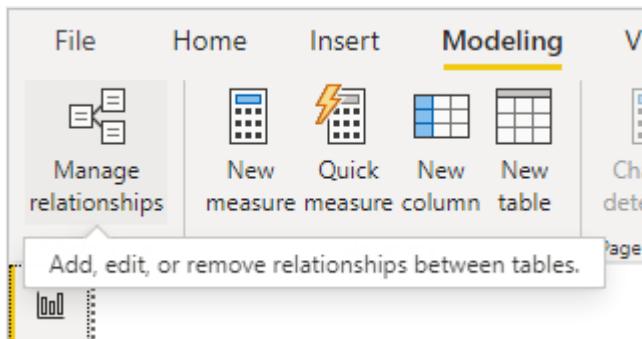
The screenshot shows the 'Fields' pane in Power BI Desktop. At the top is a search bar with the placeholder 'Search'. Below it is a tree view of fields. The 'Orders' dataset is expanded, showing fields like LineTotal, Order_Details..., OrderDate, ShipCity, and ShipCountry. The 'Products' dataset is also expanded, showing CategoryID, Discontinued, ProductID, and ProductName.

Manage the relationship between the datasets

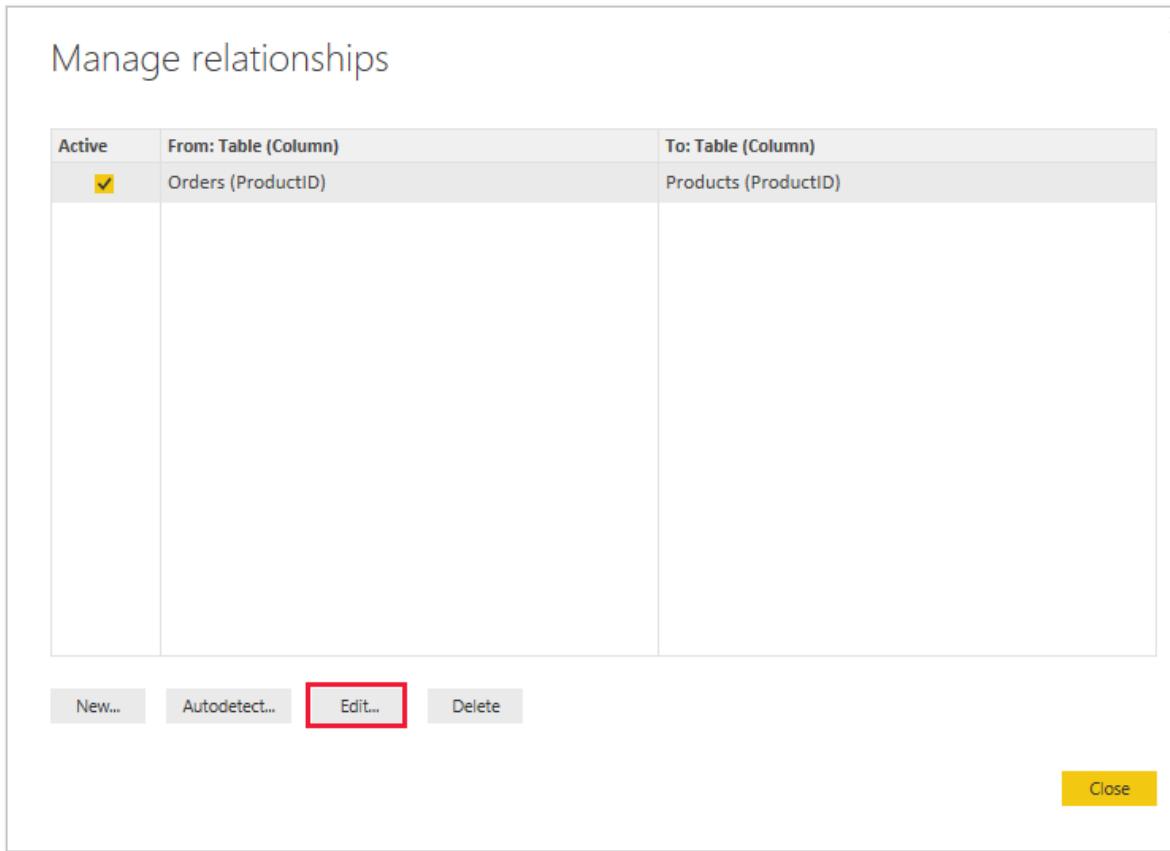
Power BI Desktop doesn't require you to combine queries to report on them. However, you can use the relationships between datasets, based on common fields, to extend, and enrich your reports. Power BI Desktop may detect relationships automatically, or you can create them in the Power BI Desktop **Manage Relationships** dialog box. For more information, see [Create and manage relationships in Power BI Desktop](#).

The shared `ProductID` field creates a relationship between this tutorial's `Orders` and `Products` datasets.

1. In Power BI Desktop Report view, select **Manage relationships** in the Modeling ribbon tab's Relationships area.



2. In the **Manage relationships** dialog box, you can see that Power BI Desktop has already detected and listed an active relationship between the **Products** and **Orders** tables. To view the relationship, select **Edit**.



Edit relationship opens, showing details about the relationship.

Edit relationship

Select tables and columns that are related.

Orders						
OrderDate	ShipCity	ShipCountry	LineTotal	ProductID	UnitPrice	Quantity
10/8/1996 12:00:00 AM	Boise	USA	\$291.9	16	13.9	21
10/8/1996 12:00:00 AM	Boise	USA	\$1,008	35	14.4	70
10/8/1996 12:00:00 AM	Boise	USA	\$288	46	9.6	30

Products								
ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel
1	Chai	1	1	10 boxes x 20 bags	18	39	0	10
2	Chang	1	1	24 - 12 oz bottles	19	17	0	10
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13	0	15

Cardinality

Many to one (*:1)

Cross filter direction

Single

Make this relationship active

Apply security filter in both directions

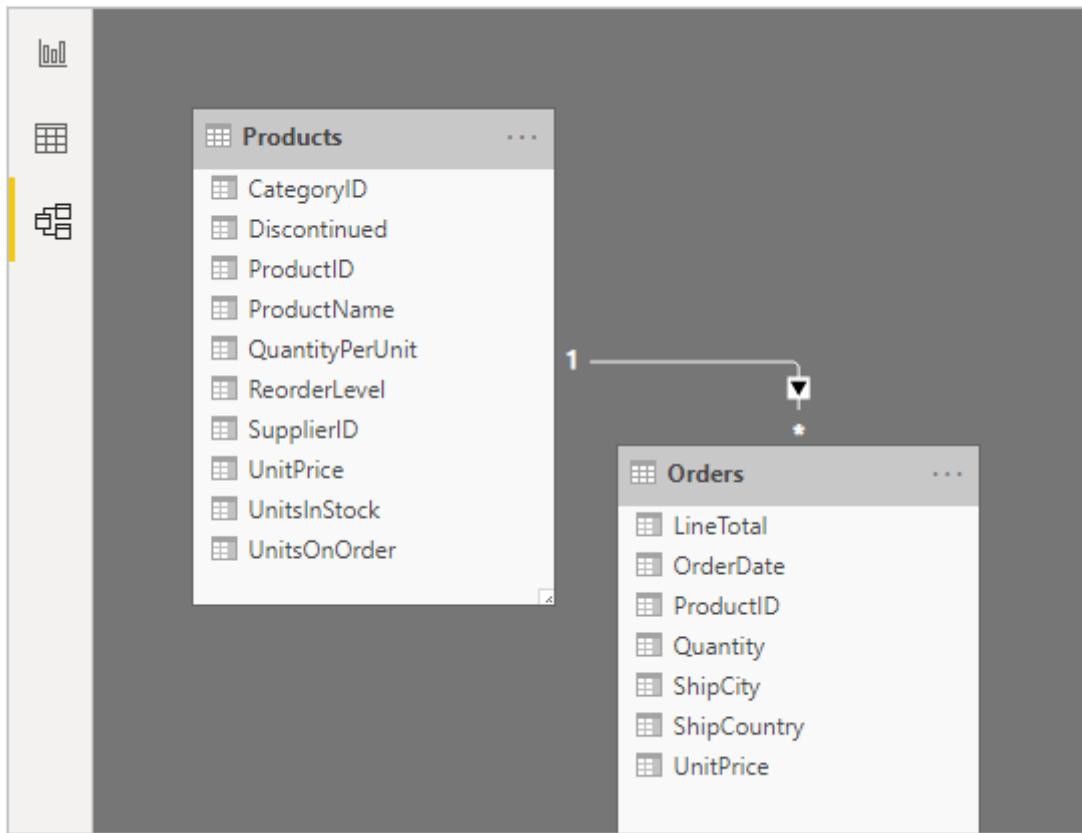
Assume referential integrity

OK

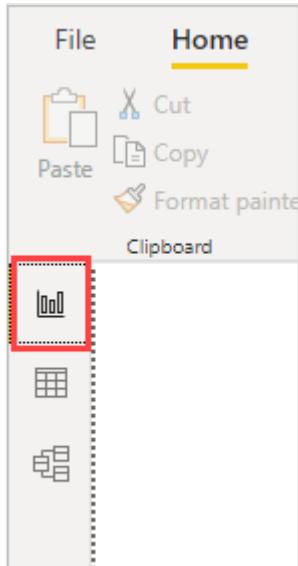
Cancel

3. Power BI Desktop has auto-detected the relationship correctly, so you can select **Cancel** and then **Close**.

In Power BI Desktop, on the left side, select **Model** to view and manage query relationships. Double-click the arrow on the line connecting the two queries to open the **Edit relationship** dialog and view or change the relationship.



To get back to **Report** view from **Model** view, select the **Report** icon.



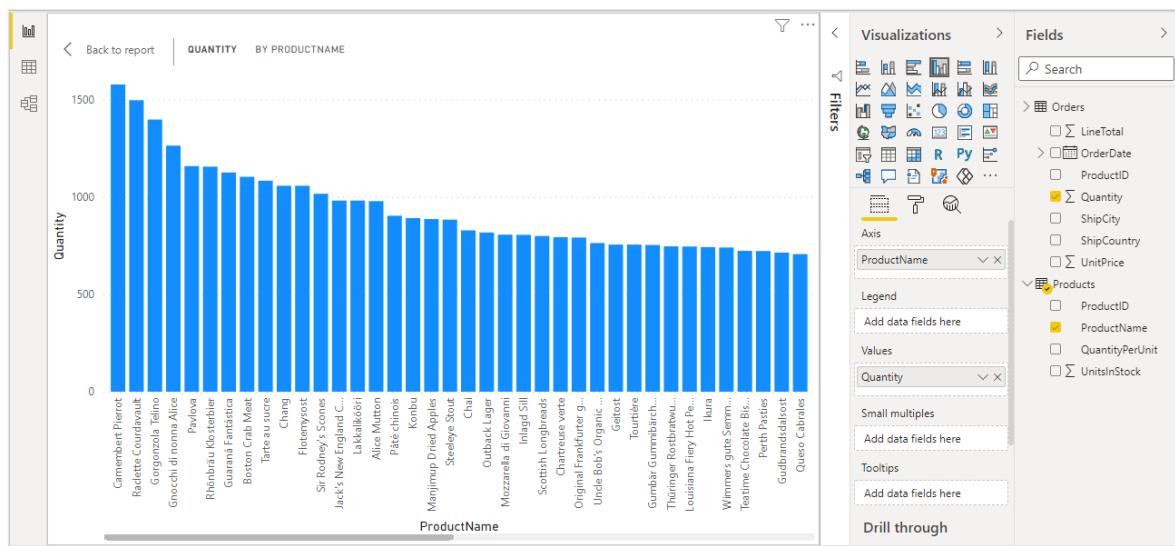
Create visualizations using your data

You can create different visualizations in Power BI Desktop Review View to gain data insights. Reports can have multiple pages, and each page can have multiple visuals. You and others can interact with your visualizations to help analyze and understand data. For more information, see [Interact with a report in Editing view in Power BI service](#).

You can use both of your data sets, and the relationship between them, to help visualize and analyze your sales data.

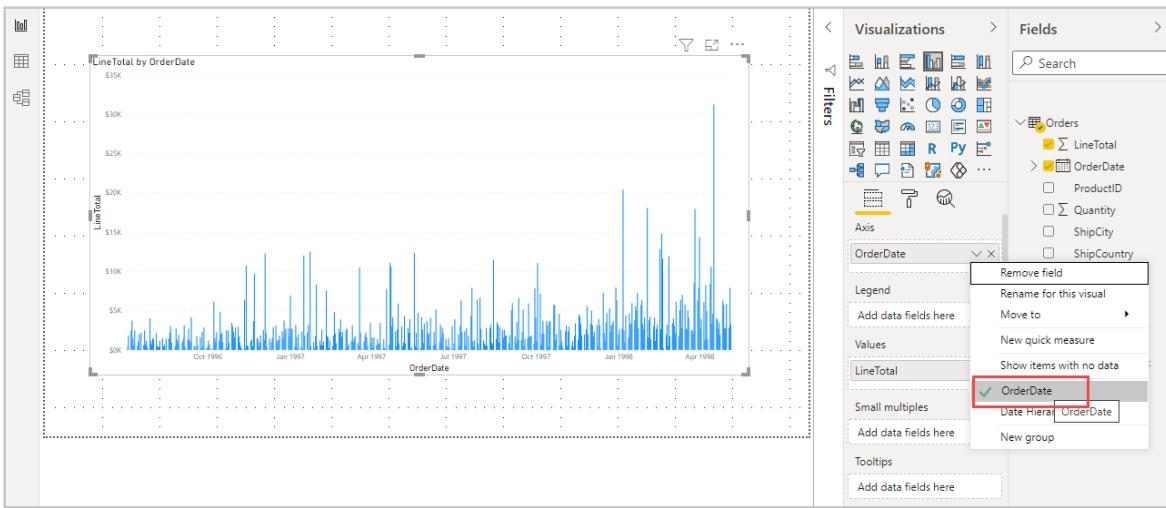
First, create a stacked column chart that uses fields from both queries to show the quantity of each product ordered.

1. Select the **Quantity** field from **Orders** in the **Fields** pane at the right, or drag it onto a blank space on the canvas. A stacked column chart is created showing the total quantity of all products ordered.
2. To show the quantity of each product ordered, select **ProductName** from **Products** in the **Fields** pane, or drag it onto the chart.
3. To sort the products by most to least ordered, select the **More options** ellipsis (...) at the visualization's upper right, and then select **Sort By > Quantity**.
4. Use the handles at the corners of the chart to enlarge it so more product names are visible.



Next, create a chart showing order dollar amounts (**LineTotal**) over time (**OrderDate**).

1. With nothing selected on the canvas, select **LineTotal** from **Orders** in the **Fields** pane, or drag it to a blank space on the canvas. The stacked column chart shows the total dollar amount of all orders.
2. Select the stacked chart, then select **OrderDate** from **Orders**, or drag it onto the chart. The chart now shows line totals for each order date.
3. Drag the corners to resize the visualization and see more data.

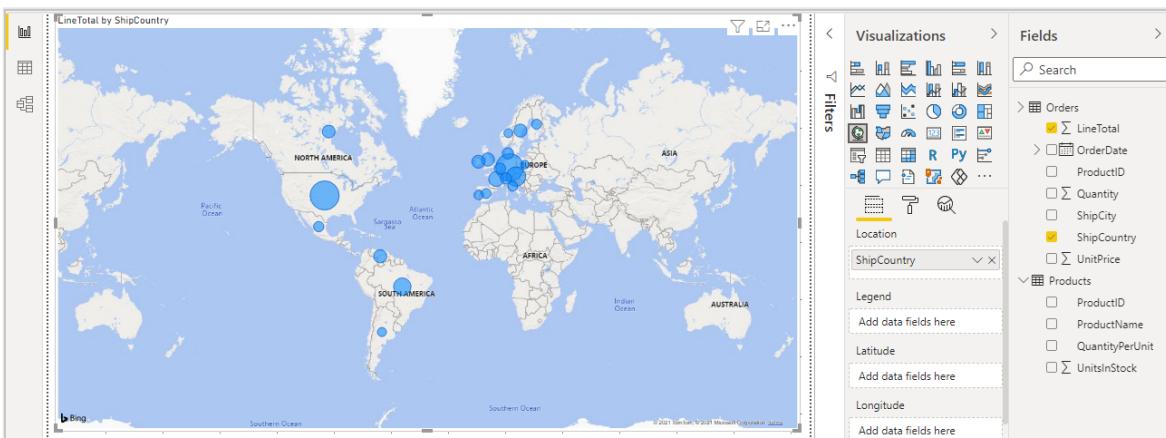


Tip

If you only see **Years** on the chart and only three data points, select the arrow next to **OrderDate** in the **Axis** field of the **Visualizations** pane, and select **OrderDate** instead of **Date Hierarchy**. Alternatively, you might need to select **Options and settings > Options** from the **File** menu, and under **Data Load**, clear the *Auto date/time for new files* option.

Finally, create a map visualization showing order amounts from each country or region.

1. With nothing selected on the canvas, select **ShipCountry** from **Orders** in the **Fields** pane, or drag it to a blank space on the canvas. Power BI Desktop detects that the data is country or region names. It then automatically creates a map visualization, with a data point for each country or region with orders.
2. To make the data point sizes reflect each country's/region's order amounts, drag the **LineTotal** field onto the map. You can also drag it to **Add data fields here** under **Size** in the **Visualizations** pane. The sizes of the circles on the map now reflect the dollar amounts of the orders from each country or region.

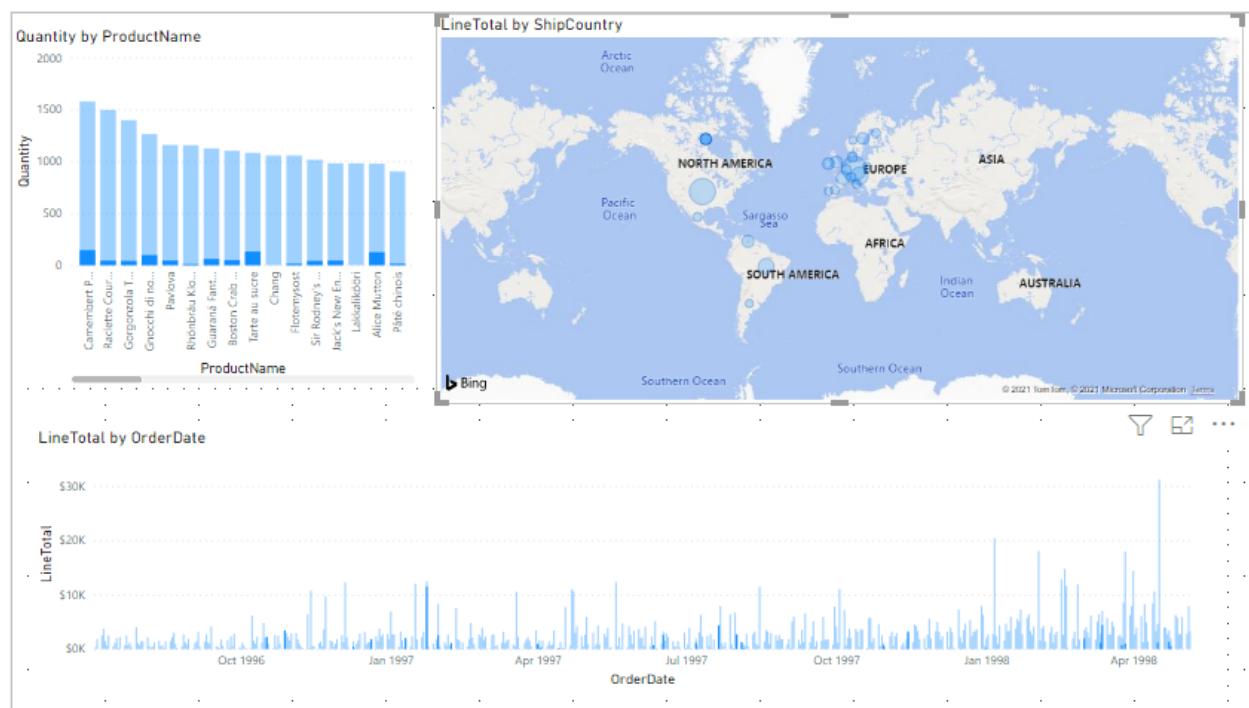


Interact with your report visuals to analyze further

In Power BI Desktop, you can interact with visuals that cross-highlight and filter each other to uncover further trends. For more information, see [Filters and highlighting in Power BI reports](#).

Because of the relationship between your queries, interactions with one visualization affect all the other visualizations on the page.

On the map visualization, select the circle centered in **Canada**. The other two visualizations filter to highlight the Canadian line totals and order quantities.



Select a **Quantity by ProductName** chart product to see the map and the date chart filter to reflect that product's data. Select a **LineTotal by OrderDate** chart date to see the map and the product chart filter to show that date's data.

Tip

To clear a selection, select it again, or select one of the other visualizations.

Complete the sales analysis report

Your completed report combines data from the *Products.xlsx* Excel file and the Northwind OData feed in visuals that help you analyze different countries' or regions'

order information, time frames, and products. When your report is ready, you can [upload it to the Power BI service](#) to share it with other Power BI users.

Next steps

- [Microsoft Learn training for Power BI](#)
- [Watch Power BI videos](#)
- [Visit the Power BI Forum ↗](#)
- [Read the Power BI Blog ↗](#)

Implement row-level security in an on-premises Analysis Services tabular model

Article • 09/23/2022

Using a sample dataset to work through the steps below, this tutorial shows you how to implement **row-level security** in an on-premises *Analysis Services Tabular Model* and use it in a Power BI report.

- Create a new security table in the [AdventureworksDW2012 database](#)
- Build the tabular model with necessary fact and dimension tables
- Define user roles and permissions
- Deploy the model to an *Analysis Services tabular* instance
- Build a Power BI Desktop report that displays data tailored to the user accessing the report
- Deploy the report to *Power BI service*
- Create a new dashboard based on the report
- Share the dashboard with your coworkers

This tutorial requires the [AdventureworksDW2012 database](#).

Task 1: Create the user security table and define data relationship

You can find many articles describing how to define row-level dynamic security with the *SQL Server Analysis Services (SSAS) tabular* model. For our sample, we use [Implement Dynamic Security by Using Row Filters](#).

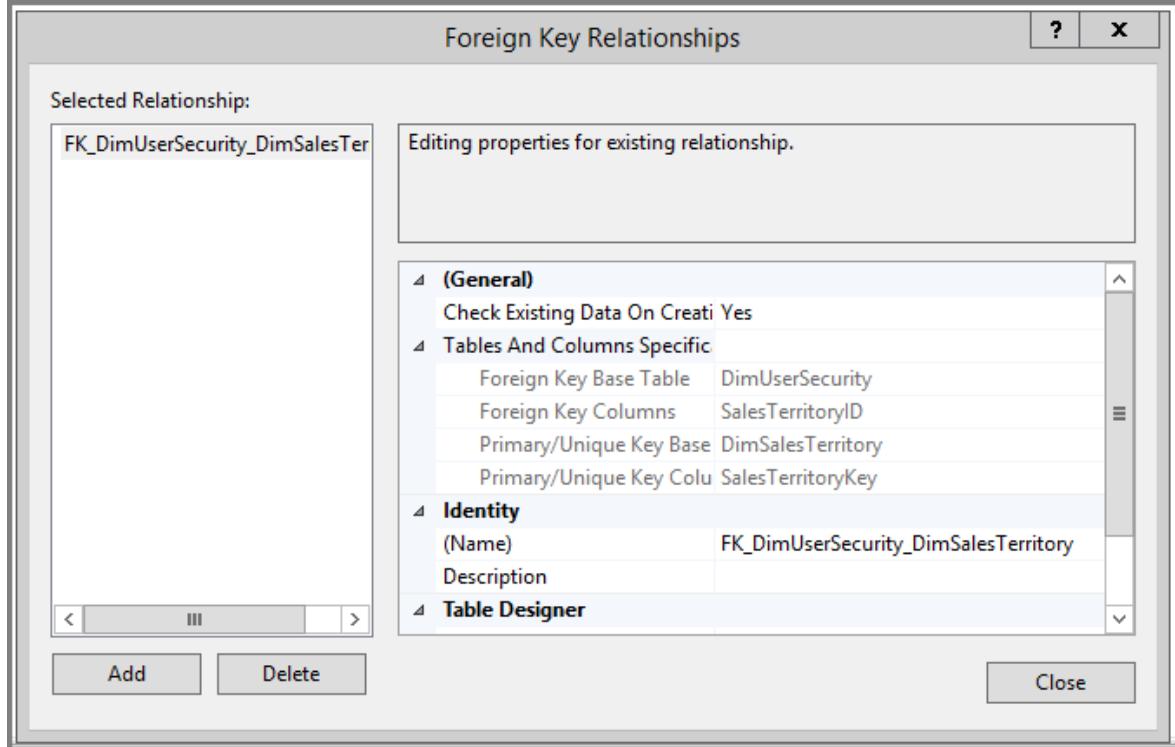
The steps here require using the AdventureworksDW2012 relational database.

1. In AdventureworksDW2012, create the `DimUserSecurity` table as shown below. You can use [SQL Server Management Studio \(SSMS\)](#) to create the table.

Column Name	Data Type	Allow Nulls
EmployeeID	int	<input type="checkbox"/>
SalesTerritoryID	int	<input type="checkbox"/>
FirstName	varchar(50)	<input type="checkbox"/>
LastName	varchar(50)	<input type="checkbox"/>
UserName	nvarchar(50)	<input type="checkbox"/>

- Once you create and save the table, you need to establish the relationship between the `DimUserSecurity` table's `SalesTerritoryID` column and the `DimSalesTerritory` table's `SalesTerritoryKey` column, as shown below.

In SSMS, right-click `DimUserSecurity`, and select **Design**. Then select **Table Designer > Relationships....** When done, save the table.



- Add users to the table. Right-click `DimUserSecurity` and select **Edit Top 200 Rows**. Once you've added users, the `DimUserSecurity` table should appear similar to the following example:

EmployeeID	SalesTerritor...	FirstName	LastName	UserName
1	1	Grace	Owens	contoso\grace
2	2	Lynn	Vanderstar	contoso\lynn
3	9	Rita	Santos	contoso\rita
4	12	Margarida	Gouveia	contoso\marga...

You'll see these users in upcoming tasks.

- Next, do an *inner join* with the `DimSalesTerritory` table, which shows the user associated region details. The SQL code here does the inner join, and the image shows how the table then appears.

SQL

```
select b.SalesTerritoryCountry, b.SalesTerritoryRegion, a.EmployeeID,
a.FirstName, a.LastName, a.UserName from [dbo].[DimUserSecurity] as a
```

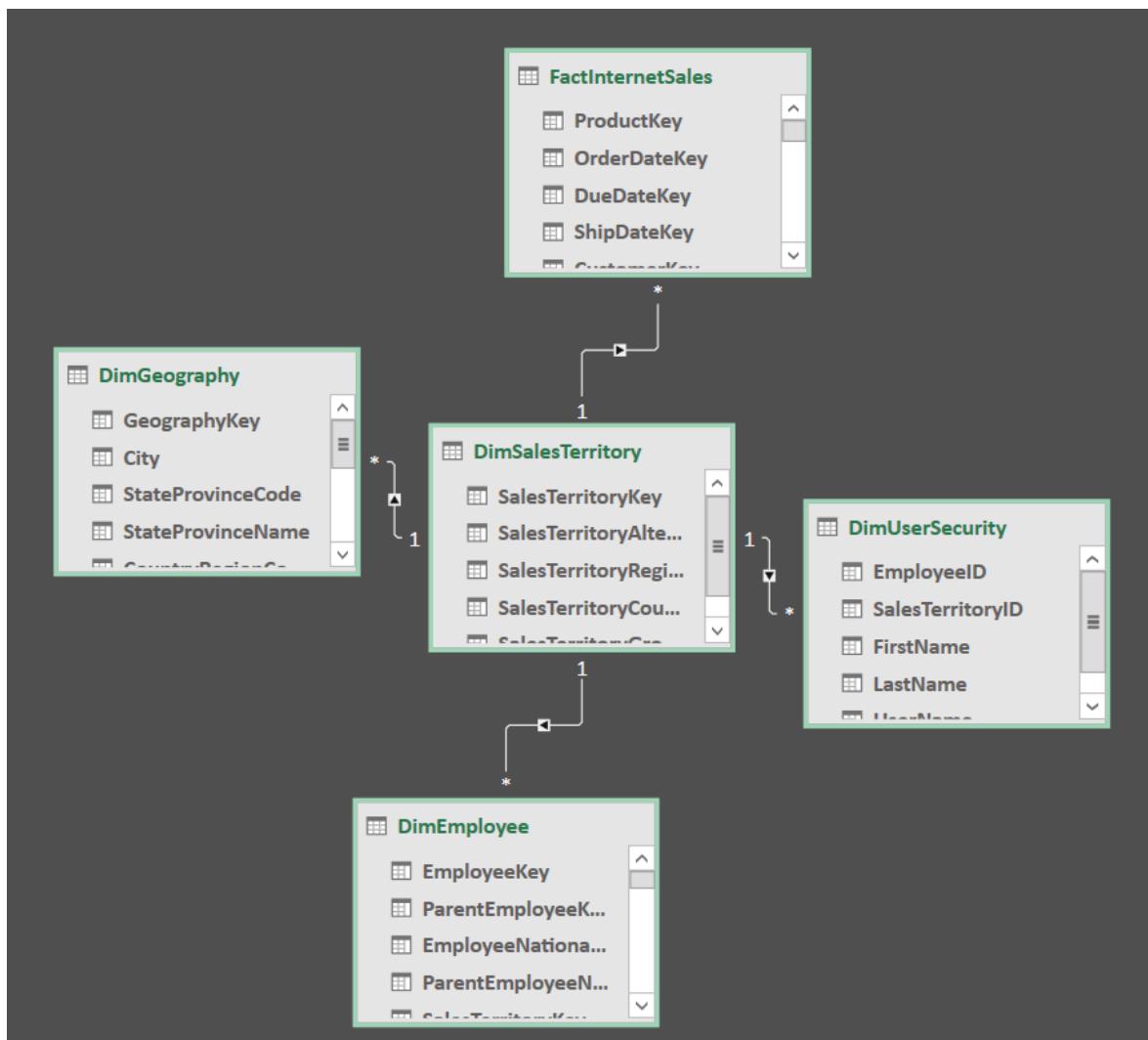
```
join [dbo].[DimSalesTerritory] as b on a.[SalesTerritoryID] = b.[SalesTerritoryKey]
```

The joined table shows who is responsible for each sales region, thanks to the relationship created in Step 2. For example, you can see that *Rita Santos* is responsible for *Australia*.

Task 2: Create the tabular model with facts and dimension tables

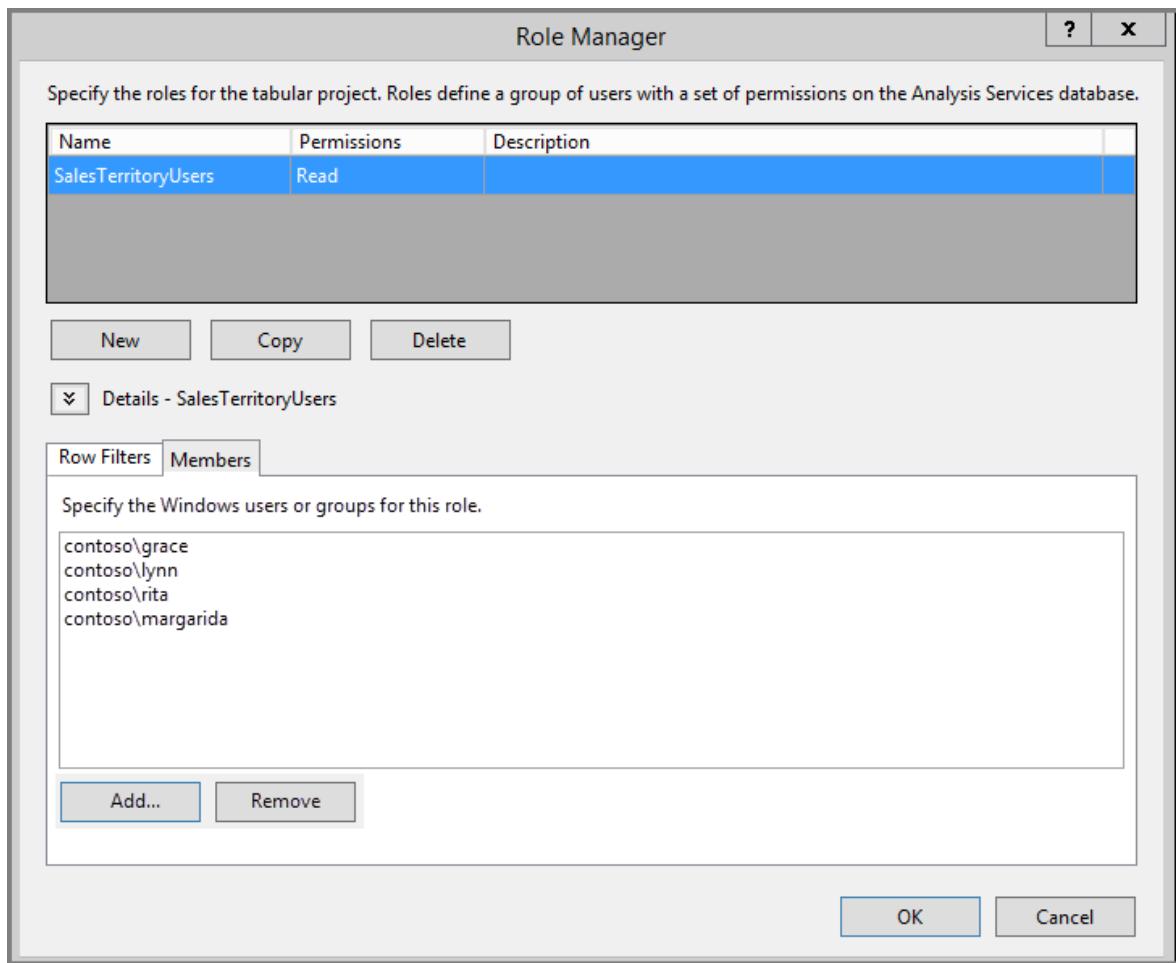
Once your relational data warehouse is in place, you need to define the tabular model. You can create the model using [SQL Server Data Tools](#) (SSDT). For more information, see [Create a New Tabular Model Project](#).

1. Import all the necessary tables into the model as shown below.

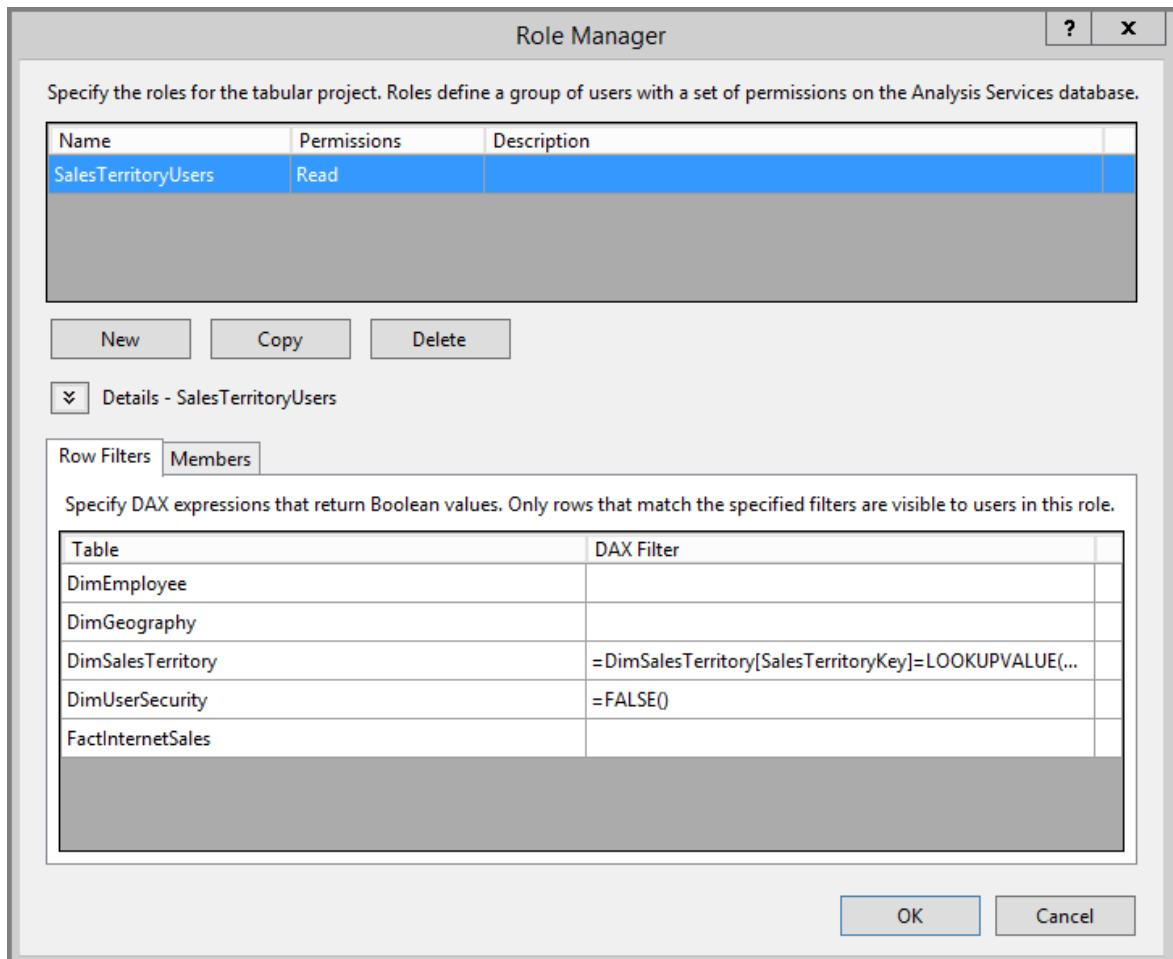


2. Once you've imported the necessary tables, you need to define a role called *SalesTerritoryUsers* with Read permission. Select the **Model** menu in SQL Server Data Tools, and then select **Roles**. In **Role Manager**, select **New**.

3. Under Members in the Role Manager, add the users that you defined in the `DimUserSecurity` table in Task 1.



4. Next, add the proper functions for both `DimSalesTerritory` and `DimUserSecurity` tables, as shown below under Row Filters tab.



5. The `LOOKUPVALUE` function returns values for a column in which the Windows user name matches the one the `USERNAME` function returns. You can then restrict queries to where the `LOOKUPVALUE` returned values match ones in the same or related table. In the **DAX Filter** column, type the following formula:

```
DAX
=DimSalesTerritory[SalesTerritoryKey]=LOOKUPVALUE(DimUserSecurity[Sales
TerritoryID], DimUserSecurity[UserName], USERNAME(),
DimUserSecurity[SalesTerritoryID],
DimSalesTerritory[SalesTerritoryKey])
```

In this formula, the `LOOKUPVALUE` function returns all values for the `DimUserSecurity[SalesTerritoryID]` column, where the `DimUserSecurity[UserName]` is the same as the current logged on Windows user name, and `DimUserSecurity[SalesTerritoryID]` is the same as the `DimSalesTerritory[SalesTerritoryKey]`.

ⓘ Important

When using row-level security, the DAX function **USERELATIONSHIP** is not supported.

The set of Sales `SalesTerritoryKey`'s `LOOKUPVALUE` returns is then used to restrict the rows shown in the `DimSalesTerritory`. Only rows where the `SalesTerritoryKey` value is in the IDs that the `LOOKUPVALUE` function returns are displayed.

6. For the `DimUserSecurity` table, in the **DAX Filter** column, add the following formula:

DAX

```
=FALSE()
```

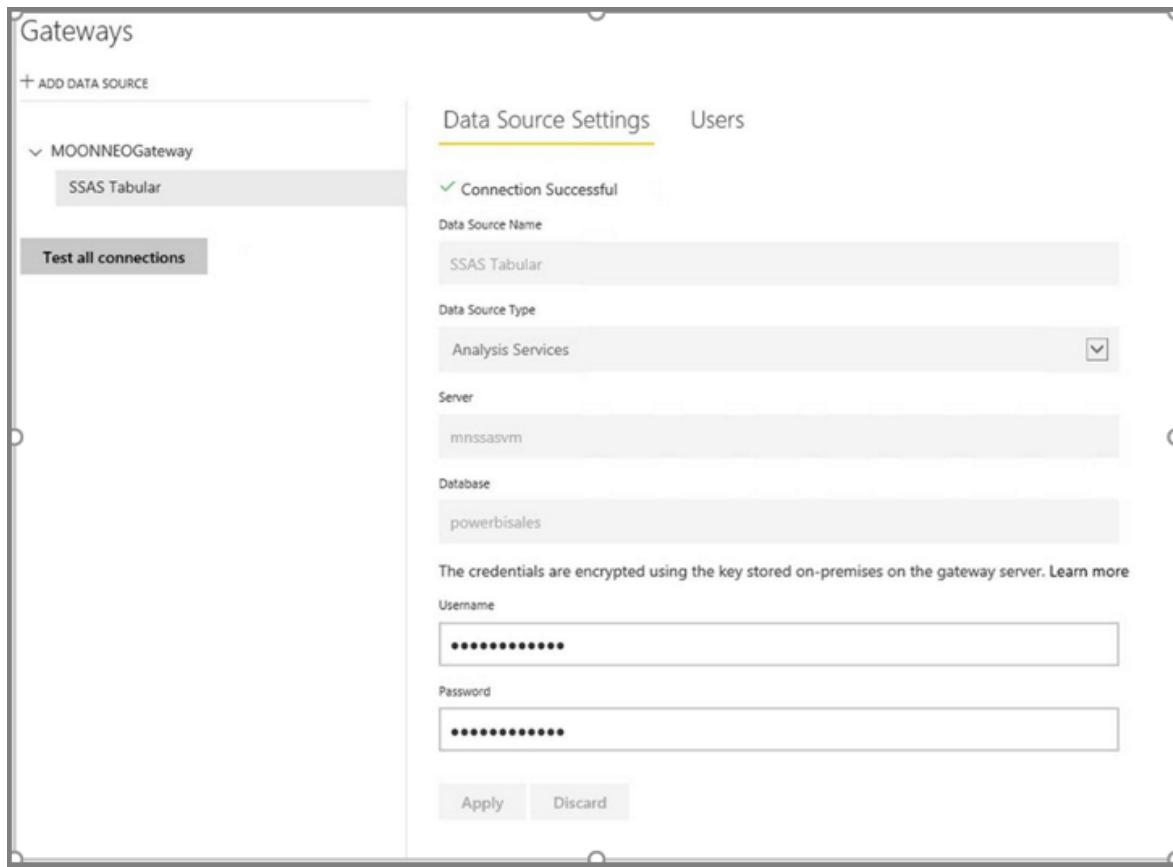
This formula specifies that all columns resolve to `false`; meaning `DimUserSecurity` table columns can't be queried.

Now you need to process and deploy the model. For more information, see [Deploy](#).

Task 3: Add Data Sources within your On-premises data gateway

Once your tabular model is deployed and ready for consumption, you need to add a data source connection to your on-premises Analysis Services tabular server.

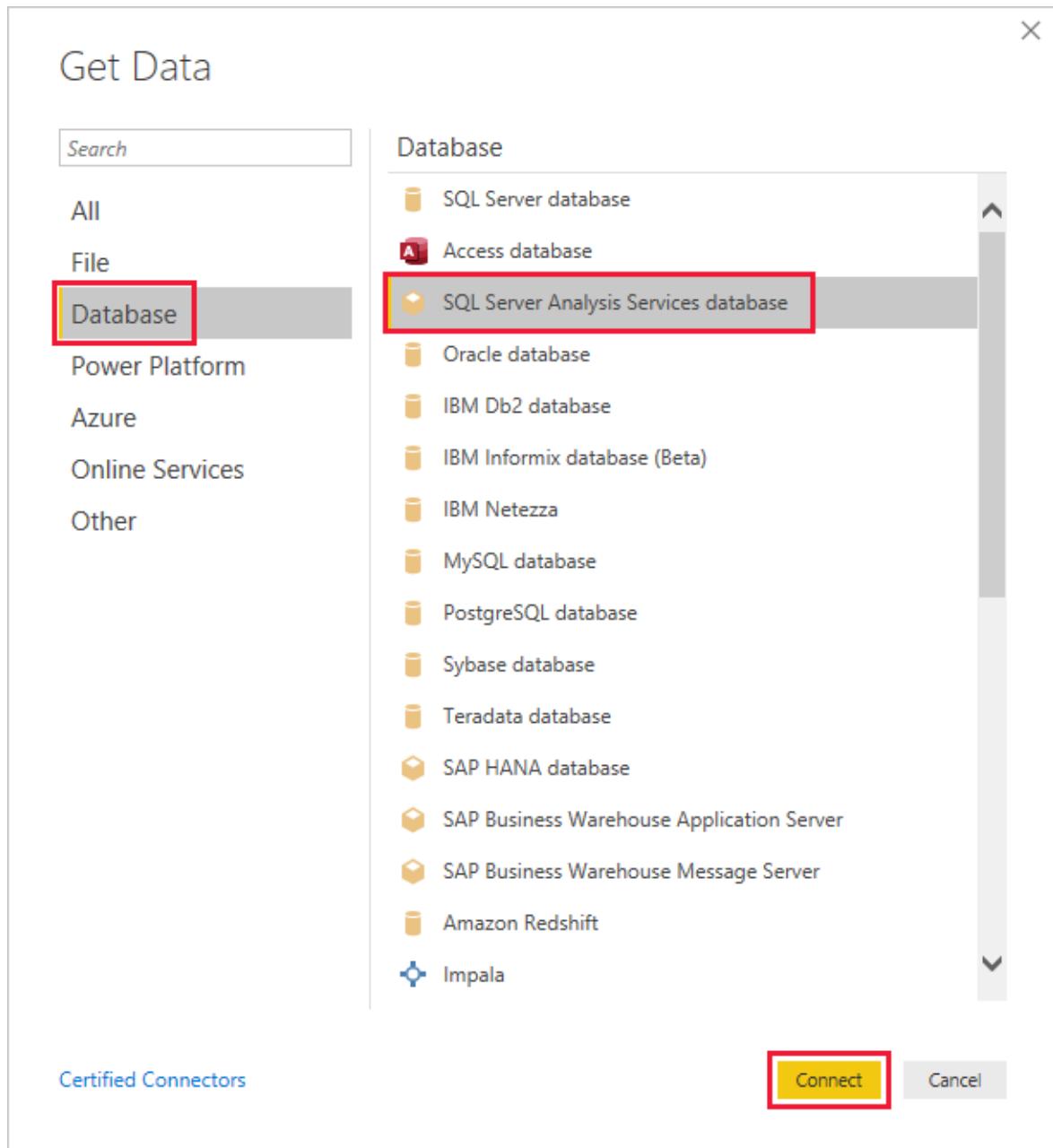
1. To allow the Power BI service access to your on-premises analysis service, you need an [on-premises data gateway](#) installed and configured in your environment.
2. Once the gateway is correctly configured, you need to create a data source connection for your *Analysis Services* tabular instance. For more information, see [Manage your data source - Analysis Services](#).



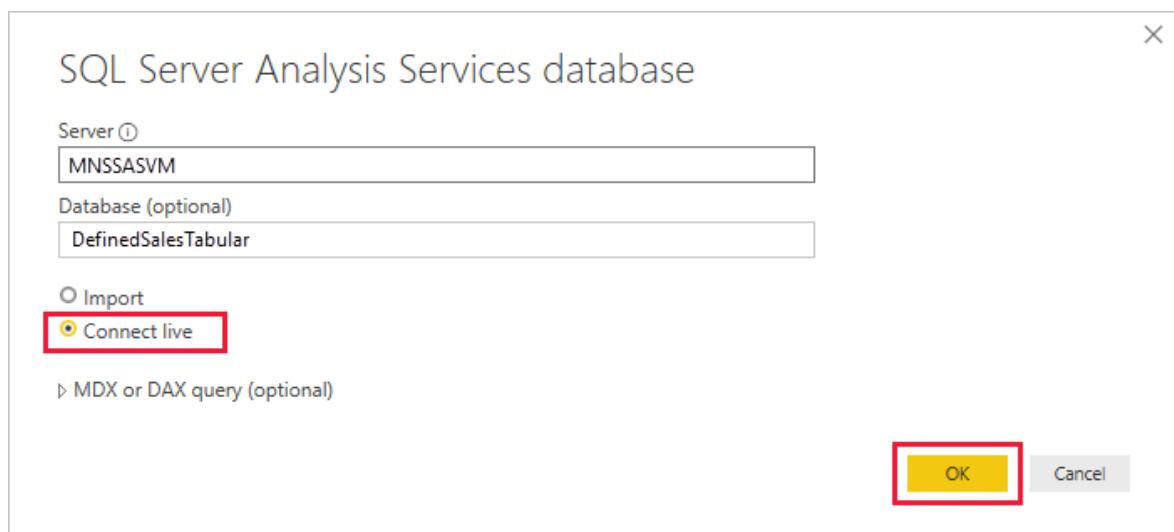
With this procedure complete, the gateway is configured and ready to interact with your on-premises Analysis Services data source.

Task 4: Create report based on analysis services tabular model using Power BI desktop

1. Start Power BI Desktop and select **Get data > Database**.
2. From the data sources list, select the **SQL Server Analysis Services Database** and select **Connect**.



3. Fill in your Analysis Services tabular instance details and select **Connect live**. Then select **OK**.



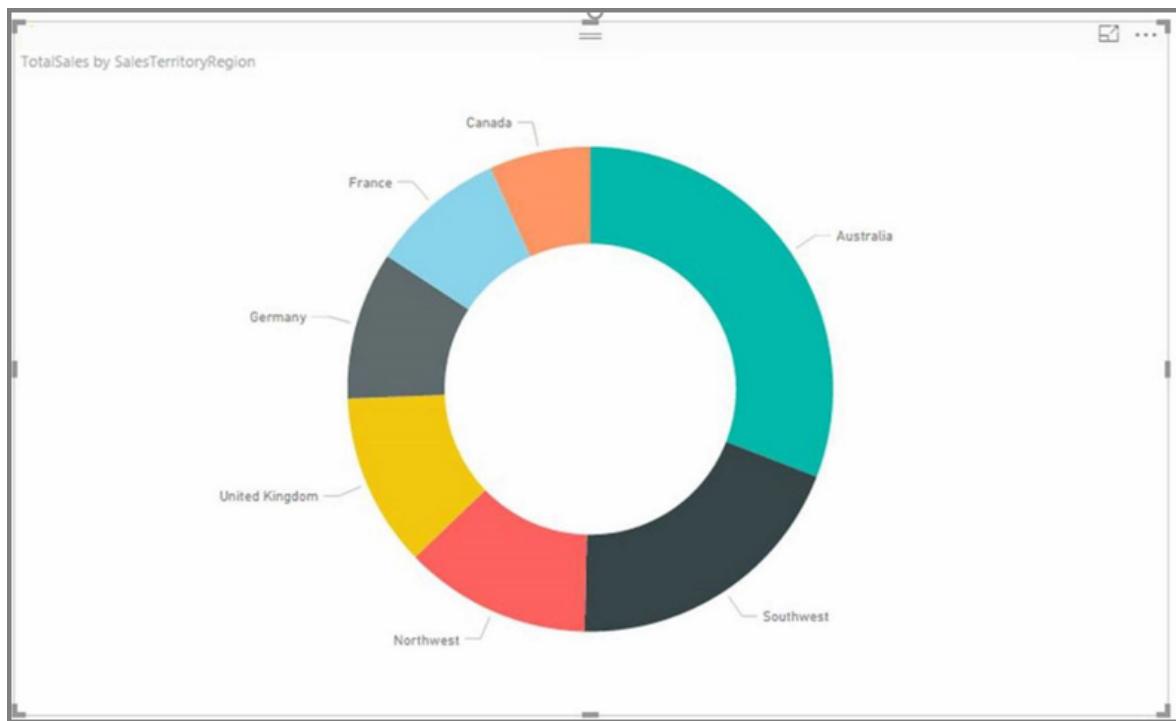
With Power BI, dynamic security works only with a live connection.

4. You can see that the deployed model is in the Analysis Services instance. Select the respective model and then select **OK**.

Power BI Desktop now displays all the available fields, to the right of the canvas in the **Fields** pane.

5. In the **Fields** pane, select the **SalesAmount** measure from the **FactInternetSales** table and the **SalesTerritoryRegion** dimension from the **SalesTerritory** table.

6. To keep this report simple, we won't add any more columns right now. To have a more meaningful data representation, change the visualization to **Donut chart**.



7. Once your report is ready, you can directly publish it to the Power BI portal. From the **Home** ribbon in Power BI Desktop, select **Publish**.

Task 5: Create and share a dashboard

You've created the report and published it to the Power BI service. Now you can use the example created in previous steps to demonstrate the model security scenario.

In the role as *Sales Manager*, the user Grace can see data from all the different sales regions. Grace creates this report and publishes it to the Power BI service. This report was created in the previous tasks.

Once Grace publishes the report, the next step is to create a dashboard in the Power BI service called *TabularDynamicSec* based on that report. In the following image, notice that Grace can see the data corresponding to all the sales region.



Now Grace shares the dashboard with a colleague, Rita, who is responsible for the Australia region sales.

Share report

TOTALSALES

Share Access

Only users with Power BI Pro will have access to this report. Recipients will have the same access as you unless row-level security on the dataset further restricts them. [Learn more](#)

Grant access to

rita@contoso.com X Enter email addresses

Hi Rita,

Check out the sales from your region.

Thanks,

Sales Manager

- Allow recipients to share your report
- Allow users to build new content using the underlying datasets
- Send email notification to recipients

Report link ⓘ

<https://contoso.com/reports/53430272-d2a3-40d6-bec1-a8184bdb01bb>

Share

Cancel

When Rita logs in to the Power BI service and views the shared dashboard that Grace created, only sales from the Australia region are visible.

Congratulations! The Power BI service shows the dynamic row-level security defined in the on-premises Analysis Services tabular model. Power BI uses the `EffectiveUserName` property to send the current Power BI user credential to the on-premises data source to run the queries.

Task 6: Understand what happens behind the scenes

This task assumes you're familiar with [SQL Server Profiler](#), since you need to capture a SQL Server profiler trace on your on-premises SSAS tabular instance.

The session gets initialized as soon as the user, Rita, accesses the dashboard in the Power BI service. You can see that the **salesterritoryusers** role takes an immediate effect with the effective user name as

```
<EffectiveUserName>rita@contoso.com</EffectiveUserName>
```

```
<PropertyList><Catalog>DefinedSalesTabular</Catalog>
<Timeout>600</Timeout><Content>SchemaData</Content><Format>Tabular</Format>
<AxisFormat>TupleFormat</AxisFormat><BeginRange>-1</BeginRange>
<EndRange>-1</EndRange><ShowHiddenCubes>false</ShowHiddenCubes>
<VisualMode>0</VisualMode><DbpropMsmdFlattened2>true</DbpropMsmdFlattened2>
<SspropInitAppName>PowerBI</SspropInitAppName>
<SecuredCellValue>0</SecuredCellValue><ImpactAnalysis>false</ImpactAnalysis>
<SQLQueryMode>Calculated</SQLQueryMode>
<ClientProcessID>6408</ClientProcessID><Cube>Model</Cube>
<ReturnCellProperties>true</ReturnCellProperties>
<CommitTimeout>0</CommitTimeout><ForceCommitTimeout>0</ForceCommitTimeout>
<ExecutionMode>Execute</ExecutionMode><RealTimeOlap>false</RealTimeOlap>
<MdxMissingMemberMode>Default</MdxMissingMemberMode>
<DisablePrefetchFacts>false</DisablePrefetchFacts>
<UpdateIsolationLevel>2</UpdateIsolationLevel>
<DbpropMsmdOptimizeResponse>0</DbpropMsmdOptimizeResponse>
<ResponseEncoding>Default</ResponseEncoding>
<DirectQueryMode>Default</DirectQueryMode><DbpropMsmdActivityID>4ea2a372-
dd2f-4edd-a8ca-1b909b4165b5</DbpropMsmdActivityID>
<DbpropMsmdRequestID>2313cf77-b881-015d-e6da-
eda9846d42db</DbpropMsmdRequestID><LocaleIdentifier>1033</LocaleIdentifier>
<EffectiveUserName>rita@contoso.com</EffectiveUserName></PropertyList>
```

Based on the effective user name request, Analysis Services converts the request to the actual `contoso\rita` credential after querying the local Active Directory. Once Analysis Services gets the credential, Analysis Services returns the data the user has permission to view and access.

If more activity occurs with the dashboard, with SQL Profiler you would see a specific query coming back to the Analysis Services tabular model as a DAX query. For example, if Rita goes from the dashboard to the underlying report, the following query occurs.

The screenshot shows the Power BI Data Model Editor with the 'Discover End' tab selected. The 'Session Initialize' section contains the following DAX code:

```

Discover End          26 - DISCO... <RestrictionList xmlns="urn:schemas-microsoft-com:xml-analysis">
Session Initialize    salesterritoryusers
Query Begin          3 - DAXQuery EVALUATE ROW(
DAX Query Plan       1 - DAX Ve... AddColumns: RelLogOp_DependentCols() 0=0 RequiredColS(0)('[SumEmployeeKey]) Sum_Vertipad: ScalogOp_DependentCols() Integer
Vertipaq SE Query Begin 0 - Vertip... SET DC_KIND="AUTO"; SELECT SUM([Employee_539e28fd-f21d-44ba-838b-ab38af0d9e5d].[EmployeeKey]) FROM [Employee_539e28fd-f21d-44ba-838b-ab38af0d9e5d]
Vertipaq SE Query Cache M... 0 - Vertip... SET DC_KIND="AUTO"; SELECT SUM([Employee_539e28fd-f21d-44ba-838b-ab38af0d9e5d].[EmployeeKey]) FROM [Employee_539e28fd-f21d-44ba-838b-ab38af0d9e5d]
Vertipaq SE Query End 0 - Vertip... SET DC_KIND="AUTO"; SELECT SUM([Employee_539e28fd-f21d-44ba-838b-ab38af0d9e5d].[EmployeeKey]) FROM [Employee_539e28fd-f21d-44ba-838b-ab38af0d9e5d]
DAX Query Plan       2 - DAX Ve... AddColumns: IterPhyOp_IterColS(0)('[SumEmployeeKey]) SingletonTable: IterPhyOp_IterColS(0)('[SumEmployeeKey]) Spool: Lookup
<

```

The 'PropertyList' section includes the following XML:

```

<PropertyList xmlns="urn:schemas-microsoft-com:xml-analysis">```
  <Catalog>DefinedSalesTabular</Catalog>
  <Cube>Model</Cube>
  <SspropInitAppName>PowerBI</SspropInitAppName>
  <EffectiveUserName>rita@contoso.com</EffectiveUserName>
  <LocaleIdentifier>1033</LocaleIdentifier>
  <ClientProcessID>6408</ClientProcessID>
  <Format>Tabular</Format>
  <Content>SchemaData</Content>
  <Timeout>600</Timeout>
  <DbpropMsmdRequestID>8510d758-f07b-a025-8fb3-
a0540189ff79</DbpropMsmdRequestID>
  <DbPropMsmdActivityID>f2dbe8a3-ef51-4d70-a879-
5f02a502b2c3</DbPropMsmdActivityID>
  <ReturnCellProperties>true</ReturnCellProperties>
  <DbpropMsmdFlattened2>true</DbpropMsmdFlattened2>
  <DbpropMsmdActivityID>f2dbe8a3-ef51-4d70-a879-
5f02a502b2c3</DbpropMsmdActivityID>
</PropertyList>

```

You can also see below the DAX query that is getting executed to populate report data.

DAX

```

EVALUATE
ROW(
  "SumEmployeeKey", CALCULATE(SUM(Employee[EmployeeKey]))
)

<PropertyList xmlns="urn:schemas-microsoft-com:xml-analysis">```
  <Catalog>DefinedSalesTabular</Catalog>
  <Cube>Model</Cube>
  <SspropInitAppName>PowerBI</SspropInitAppName>
  <EffectiveUserName>rita@contoso.com</EffectiveUserName>
  <LocaleIdentifier>1033</LocaleIdentifier>
  <ClientProcessID>6408</ClientProcessID>
  <Format>Tabular</Format>
  <Content>SchemaData</Content>
  <Timeout>600</Timeout>
  <DbpropMsmdRequestID>8510d758-f07b-a025-8fb3-
a0540189ff79</DbpropMsmdRequestID>
  <DbPropMsmdActivityID>f2dbe8a3-ef51-4d70-a879-
5f02a502b2c3</DbPropMsmdActivityID>
  <ReturnCellProperties>true</ReturnCellProperties>
  <DbpropMsmdFlattened2>true</DbpropMsmdFlattened2>
  <DbpropMsmdActivityID>f2dbe8a3-ef51-4d70-a879-
5f02a502b2c3</DbpropMsmdActivityID>
</PropertyList>

```

Considerations

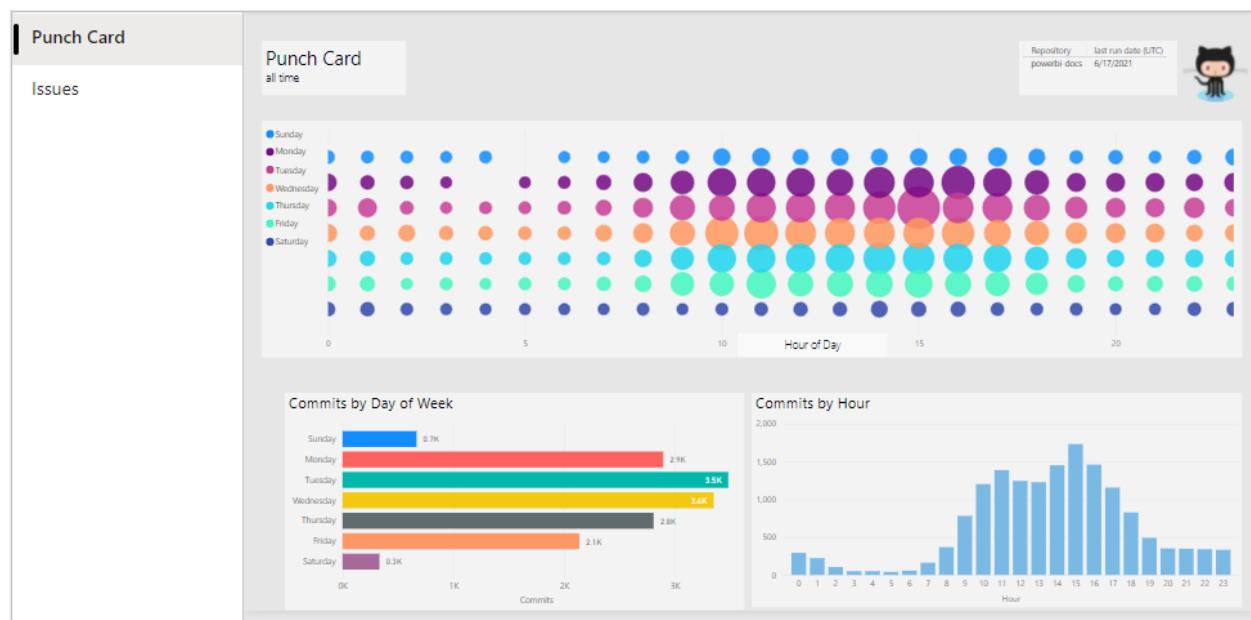
- On-premises row-level security with Power BI is only available with live connection.
- Any changes in the data after processing the model would be immediately available for the users accessing the report with live connection from the Power BI service.

Tutorial: Connect to a GitHub repo with Power BI

Article • 02/08/2023

In this tutorial, you connect to real data: the Power BI content public repository (also known as a *repo*) in the GitHub service. Power BI automatically creates a dashboard and report with the data. You see answers to questions like: How many people contribute to the Power BI public repo? Who contributes the most? Which day of the week has the most contributions? And other questions.

You can connect to your own private or public GitHub repos too. To use a Power BI *template app* to connect to your repos, see [Connect to GitHub with Power BI](#).



In this tutorial, you complete the following steps:

- ✓ Sign up for a GitHub account, if you don't have one yet.
- ✓ Sign in to your Power BI account, or sign up, if you don't have one yet.
- ✓ Open the Power BI service.
- ✓ Find the GitHub app.
- ✓ Enter the information for the Power BI public GitHub repo.
- ✓ View the dashboard and report with GitHub data.
- ✓ Clean up resources by deleting the app.

If you're not signed up for Power BI, [sign up for a free trial](#) before you begin.

Prerequisites

To complete this tutorial, you need a GitHub account, if you don't already have one.

- Sign up for a [GitHub account](#).

How to connect

1. Sign in to the Power BI service (app.powerbi.com).

2. In the nav pane, select Apps, then Get apps.

The screenshot shows the Microsoft Power BI Apps interface. On the left, there's a navigation pane with icons for Home, Create, Browse, Data hub, Metrics, and Apps (which is highlighted with a red box). The main area is titled 'Apps' and contains three cards: 'Github', 'Power BI Release Plan', and 'Zendesk'. A green 'Get apps' button is located in the top right corner of the main area, also highlighted with a red box.

3. Enter GitHub in the search box. Select the app, and then choose Get it now.

The screenshot shows the Microsoft AppSource page for the 'GitHub' app. The top navigation bar includes 'Microsoft', 'AppSource', 'Apps', a search bar, and user profile icons. The main content area shows the 'GitHub' app card, which includes a logo, the name 'GitHub', a 'save for later' button, the publisher 'Microsoft', a rating of '3.4 (20)', and tabs for 'Overview' and 'Reviews'. Below the card, there's a description of the app's functionality and a large blue 'GET IT NOW' button, which is highlighted with a red box. To the right of the card, there's a preview of a Power BI dashboard with various charts and metrics related to GitHub data.

4. Select Install.

Install this Power BI app?

X



Github Repository Usage

by Microsoft

[View in AppSource](#)

Apps may contain security or privacy risks. Only install apps from trusted authors and sources.

[Learn more](#)

By installing this app I agree to the publisher's [privacy policy](#) and [terms of service](#)

[Install](#)

Cancel

5. When you see the notification, **Your new app is ready!**, select **Go to app**.

6. On the app landing page, select **Connect your data**.

The screenshot shows the Power BI app interface for the 'Github' repository. The top navigation bar includes 'Power BI' and the date '1/20/2023, 9:2...'. On the right, there are links for 'Dashboard', 'File', 'Chat in Teams', 'Comment', 'Email subscription', and more. Below the navigation is a yellow banner with the text '⚠ You're viewing this app with sample data' and a green button labeled 'Connect your data'. The main content area displays a chart titled 'Pull Requests by Month' with two categories: 'Open pull requests' (blue) and 'Closed pull requests' (black).

7. In the connect dialog, enter the repository name and repository owner of the repo.

The URL for this repo is <https://github.com/MicrosoftDocs/powerbi-docs>. Enter

MicrosoftDocs as the repository **Owner**, and **powerbi-docs** as the **Repo** name.

Select **Next**.

X



Connect to Github

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

Parameters

Make sure all required (*) parameters are filled in before connecting to your data.

Owner *

Please enter repository owner

Repo *

Please enter the repository name

Advanced ▾

[Go to the app documentation ↗](#)

Next

Cancel

8. Make sure that **Authentication Method** is set to `OAuth2`, and then select **Sign in and connect**.



Connect to Github

X

You are connecting to

ExtensionDataSourceKind

github

ExtensionDataSourcePath

github



Authentication method

OAuth2

Privacy level setting for this data source [Learn more](#)

Private



[Go to the app documentation ↗](#)

Back

Sign in and connect

Cancel

9. If prompted, follow the GitHub authentication instructions and give Power BI permission to access your data.

- After Power BI can connect with GitHub, the data in your Power BI dataset is refreshed once a day.
- After Power BI imports the data, you see the contents in your new GitHub workspace.

10. Select **Workspaces** in the nav pane to see the dashboard, reports, and datasets. You can select **More options (...)** to view settings.

The screenshot shows the Microsoft Power BI workspace list interface. On the left, a sidebar menu includes Home, Create, Browse, Data hub, Metrics, Apps, Deployment pipelines, Learn, and Workspaces. The 'Workspaces' icon is highlighted with a red box. Below the sidebar is a search bar. The main area displays a list of workspaces:

- Dynamics Performance Test Res...
- Events
- FY22 Adv Spec Tracker
- Github** (selected workspace, highlighted with a red box)
- MSX Insights PRD
- my-test-workspace
- Next Best Workload
- Power BI Premium Capacity Met...
- Power BI Release Plan
- Smartsheet
- SMC Analytical Framework
- Solutions [DataverseADL]
- TSA Dashboards
- Zendesk

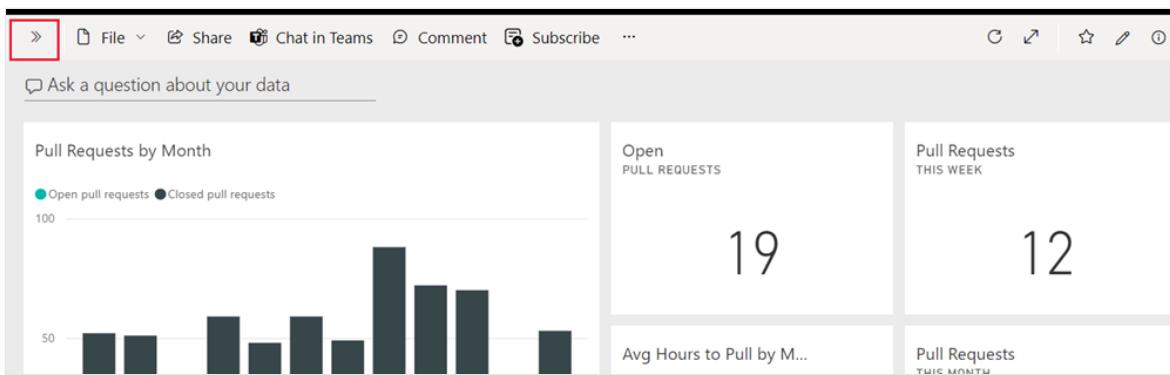
A context menu is open over the 'Github' workspace, showing options: 'Type', 'Workspace settings', and 'Workspace access'. At the bottom of the workspace list is a green button labeled '+ New workspace'.

11. In workspace **Settings**, you can rename or delete the workspace.

The screenshot shows the GitHub workspace settings interface. At the top, there is a search bar and a user profile icon. Below the header, the title 'Settings' is displayed with a gear icon. The main section is titled 'My GitHub workspace'. There are three navigation tabs: 'About' (underlined), 'Premium', and 'Azure connections'. Under 'About', there is a 'Workspace image' section with a GitHub logo, an 'Upload' button with an upward arrow, and a 'Delete' button with a trash can icon. A red box highlights the 'Workspace name *' input field, which contains the text 'My GitHub workspace'. Below this, there is a 'Available' status indicator and a 'Description' section with a placeholder 'Describe this workspace'. At the bottom, there is a link 'Learn more about workspace settings' and an 'Advanced' dropdown menu. The footer contains three buttons: 'Delete workspace', 'Save' (which is highlighted with a red box), and 'Cancel'.

12. Select your GitHub dashboard. You can minimize or expand the nav pane, so you have more room to see your data.

The GitHub dashboard contains live data, so the values you see may be different.



Ask a question

1. Select the **Ask a question about your data** text box. Power BI opens the Q&A window and offers some sample questions.
2. Enter **how many users**, Power BI offers a list of questions.

The Q&A window displays the following suggestions for the query **how many users**:

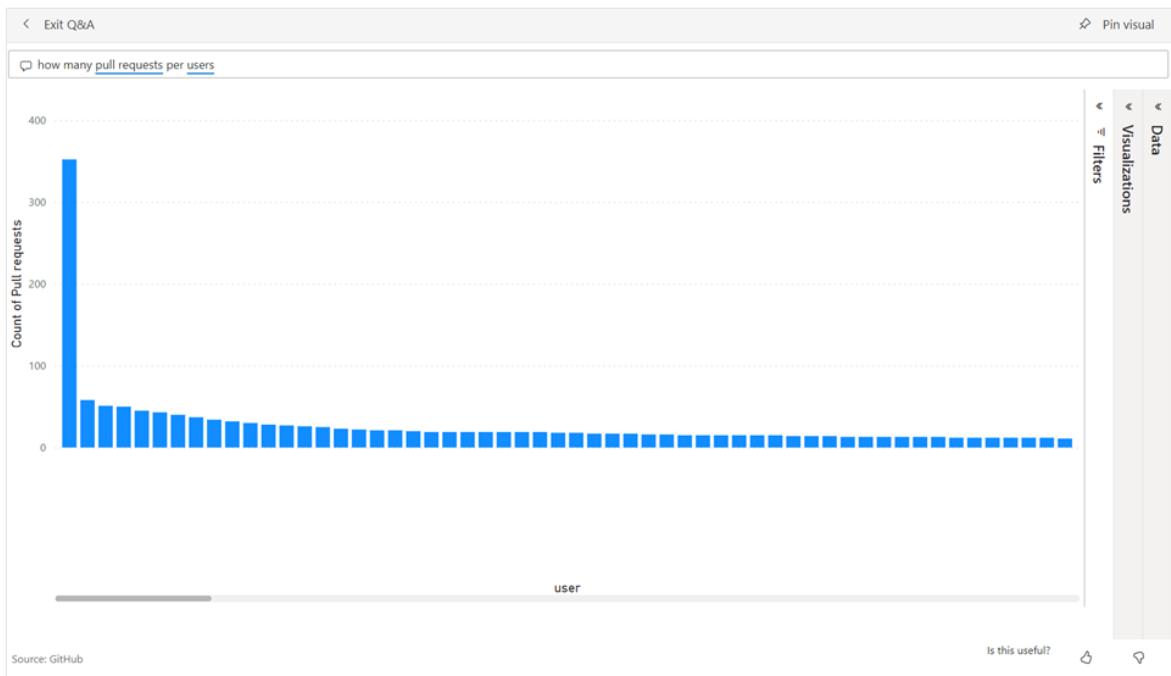
- how many users **were calculated**
- how many users **are named woodbridge**
- how many users **by date of first commit**
- how many users **sorted by rank by commits this year**

Show more

2689
Count of Users

3. You can edit your question, for example, in between **how many** and **users**, type **pull requests per**.

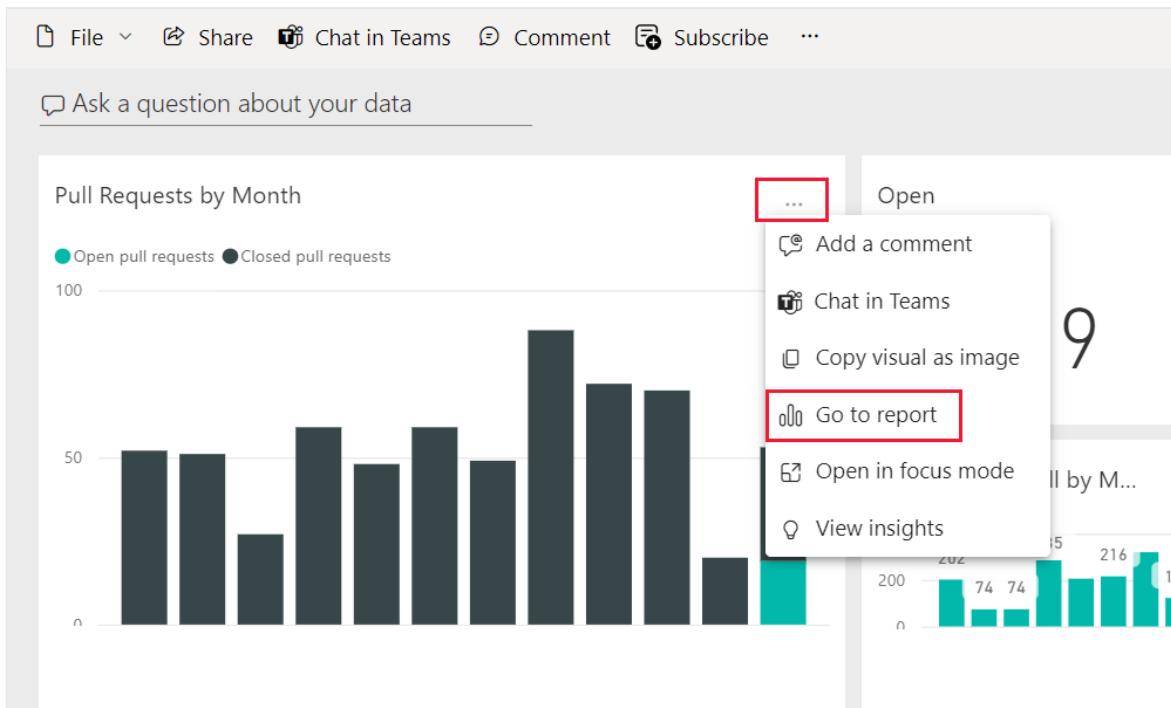
Power BI creates a bar chart visual that shows the number of pull requests per person.



4. Select the pin icon to pin the visual to your dashboard, then **Exit Q&A**.

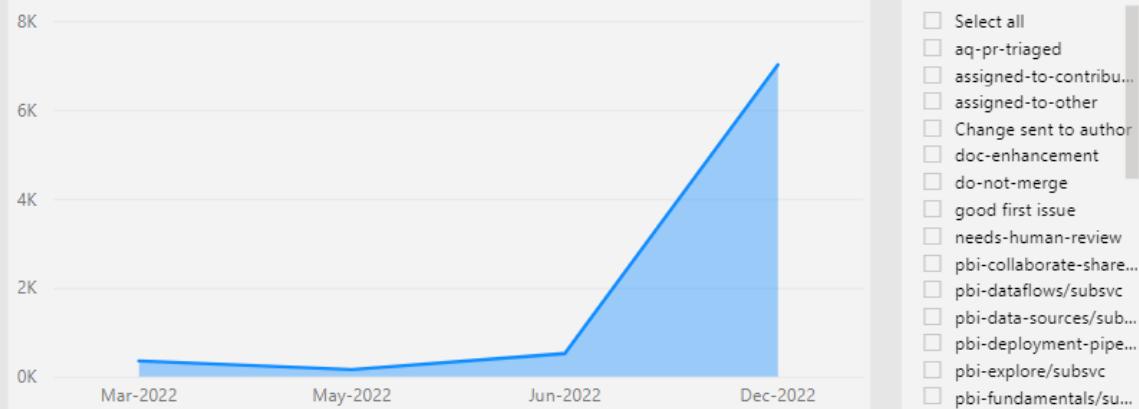
View the GitHub report

1. On the GitHub dashboard, select **More options (...)** on the column chart **Pull Requests by Month**. Choose **Go to report**.



2. Select a user name in the **Total pull requests by user** chart. A new tile appears with results for one user.

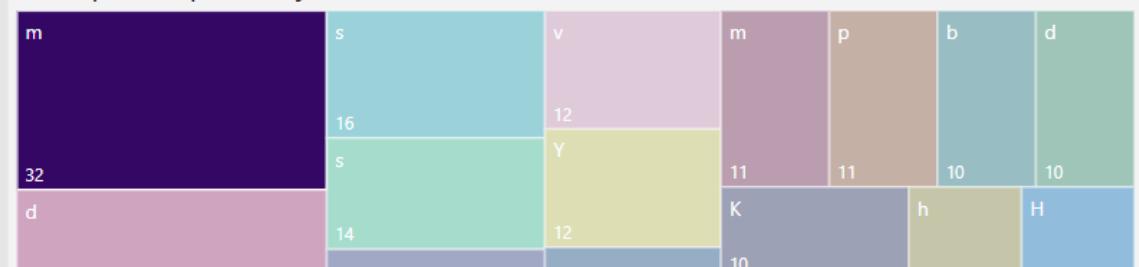
Total Hours to Pull by Month



Label

Select all
aq-pr-triaged
assigned-to-contribu...
assigned-to-other
Change sent to author
doc-enhancement
do-not-merge
good first issue
needs-human-review
pbi-collaborate-share...
pbi-dataflows/subsvc
pbi-data-sources/sub...
pbi-deployment-pipe...
pbi-explore/subsvc
pbi-fundamentals/su...

Total pull requests by user



3. Select the **Punch Card** tab to view the next page of the report. Now you can see volumes of work by hour of the day and day of the week.



Clean up resources

Now that you've finished the tutorial, you can delete the GitHub app.

1. In the nav pane, select Apps.

2. On the app tile, select **More options (...)** and then choose **Delete**.

The screenshot shows the Power BI 'Apps' page. On the left, a vertical sidebar lists various options: Home, Create, Browse, Data hub, Metrics, Apps (which is highlighted with a red box), Deployment pipelines, Learn, and Workspaces. The main area is titled 'Apps' and contains a brief description: 'Apps are collections of dashboards and reports in one easy-to-find place.' Below this, there are several app tiles. The first tile is for 'Github', featuring a black cat icon and the text 'Explore and monitor key repository in Po... Version 19'. To the right of the Github tile is another tile for 'Power BI Release Plan' with a bar chart icon and the text 'Because this report lists fe... Version 8'. At the bottom of the list is a tile for 'Smartsheet' with the text 'Get insights into your work with the Smart... Version 8'. To the right of the Smartsheet tile is a context menu with three options: 'Edit', 'Delete' (which is highlighted with a red box), and '...'. The entire screenshot is framed by a thin black border.

Next steps

In this tutorial, you've connected to a GitHub public repo and gotten data, which Power BI has formatted in a dashboard and report. You've answered some questions about the data by exploring the dashboard and report. Now you can learn more about connecting to other services, such as Salesforce, Microsoft Dynamics, and Google Analytics.

[Connect to GitHub with a Power BI template app](#)

Tutorial: Use Cognitive Services in Power BI

Article • 06/01/2023

Power BI provides access to a set of functions from Azure Cognitive Services to enrich your data in the self-service data prep for dataflows. The services that are supported today are [sentiment analysis](#), [key phrase extraction](#), [language detection](#), and [image tagging](#). The transformations run on the Power BI service and don't require an Azure Cognitive Services subscription. This feature requires Power BI Premium.

Cognitive Services transforms are supported in the [self-service data prep for dataflows](#). Use the step-by-step examples for text analytics and image tagging in this article to get started.

In this tutorial, you learn how to:

- ✓ Import data into a dataflow
- ✓ Score sentiment and extract key phrases of a text column in a dataflow
- ✓ Connect to the results from Power BI Desktop

Prerequisites

To complete this tutorial, you need the following prerequisites:

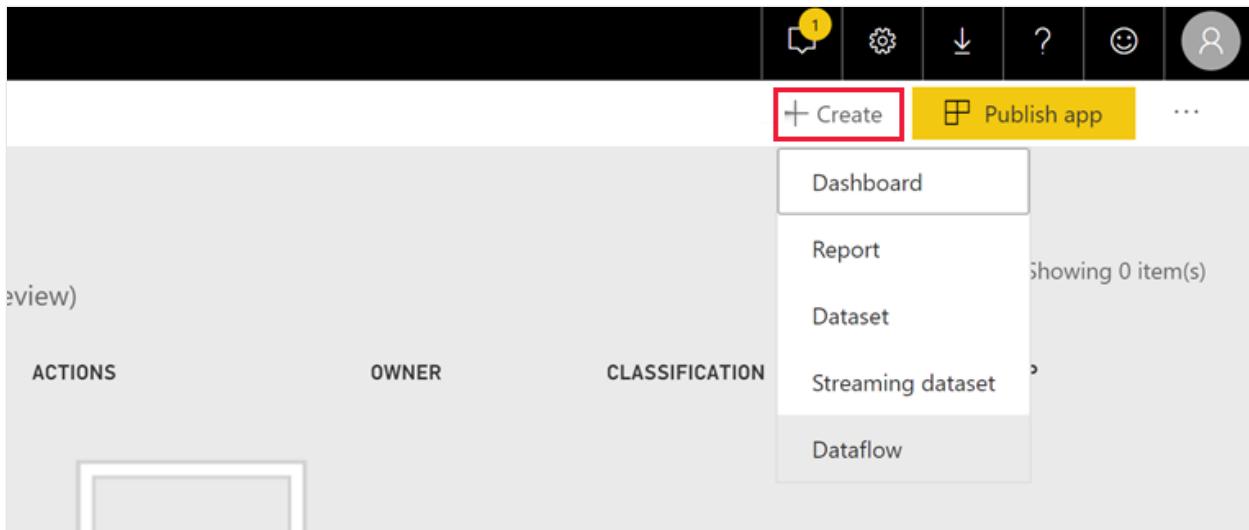
- A Power BI account. If you're not signed up for Power BI, [sign up for a free trial](#) before you begin.
- Access to a Power BI Premium capacity with the AI workload enabled. This workload is turned off by default during preview. If you are in on a Premium capacity and AI Insights aren't showing up, contact your Premium capacity administrator to enable the AI workload in the admin portal.

Text analytics

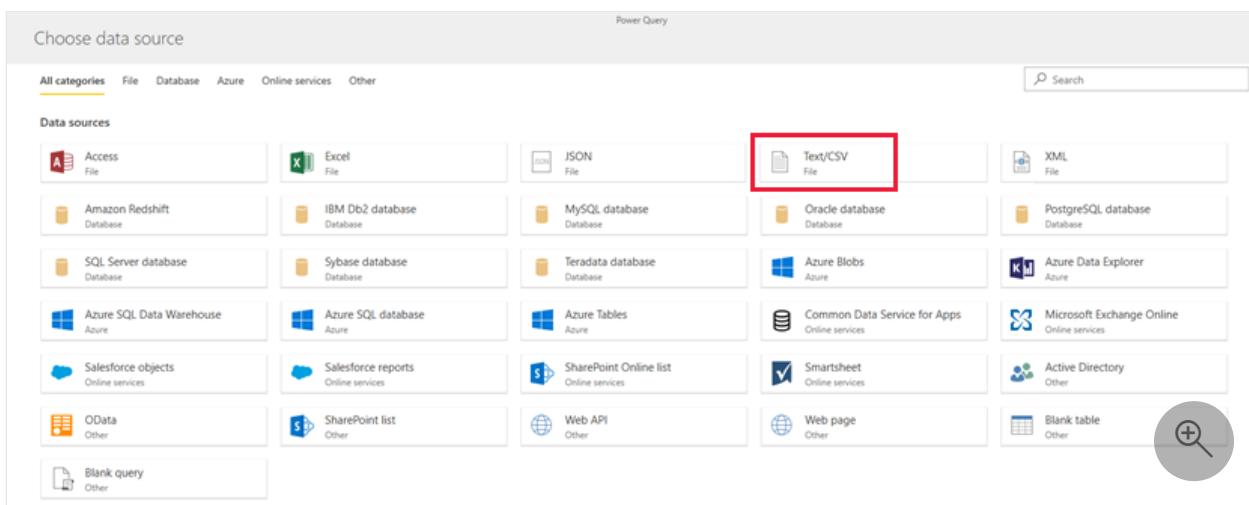
Follow the steps in this section to complete the text analytics portion of the tutorial.

Step 1: Apply sentiment scoring in the Power BI service

To get started, navigate to a Power BI workspace with Premium capacity and create a new dataflow using the **Create** button in the upper right of the screen.



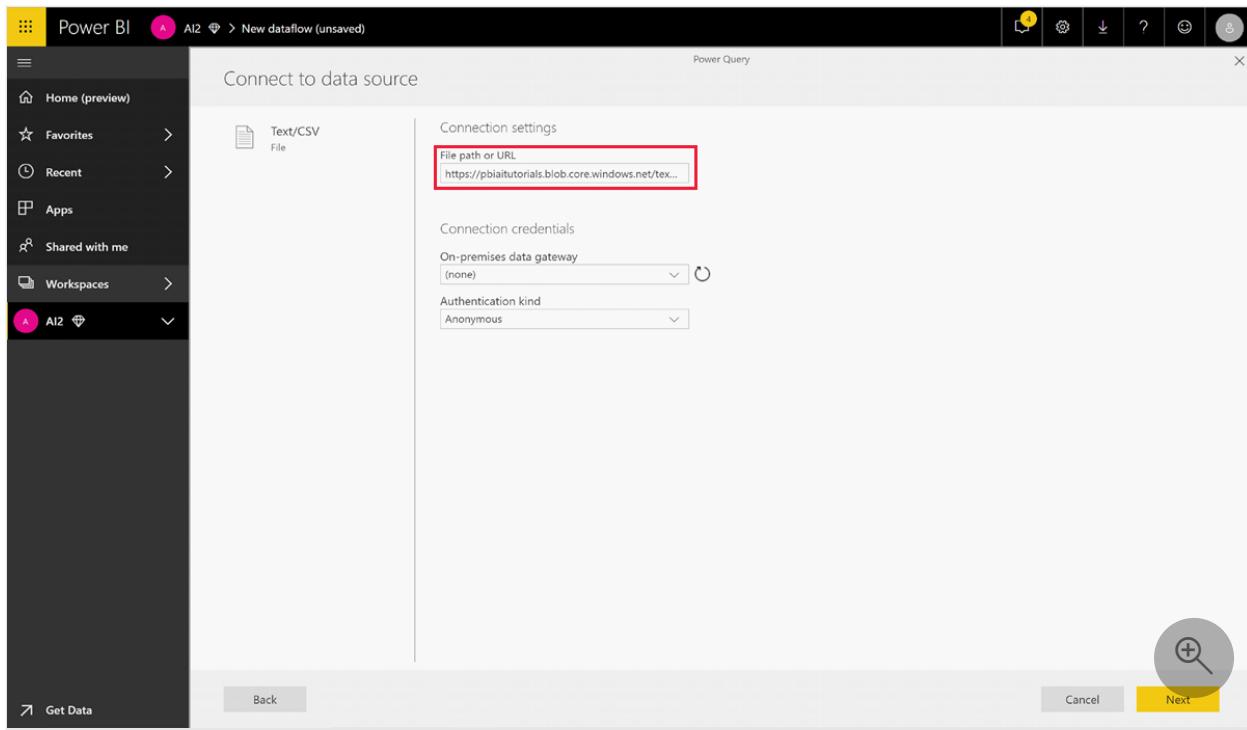
The dataflow dialog shows you the options for creating a new dataflow, select **Add new entities**. Next, choose **Text/CSV** from the menu of data sources.



Paste this URL into the URL field:

<https://pbaitutorials.blob.core.windows.net/textanalytics/FabrikamComments.csv>

and select **Next**.

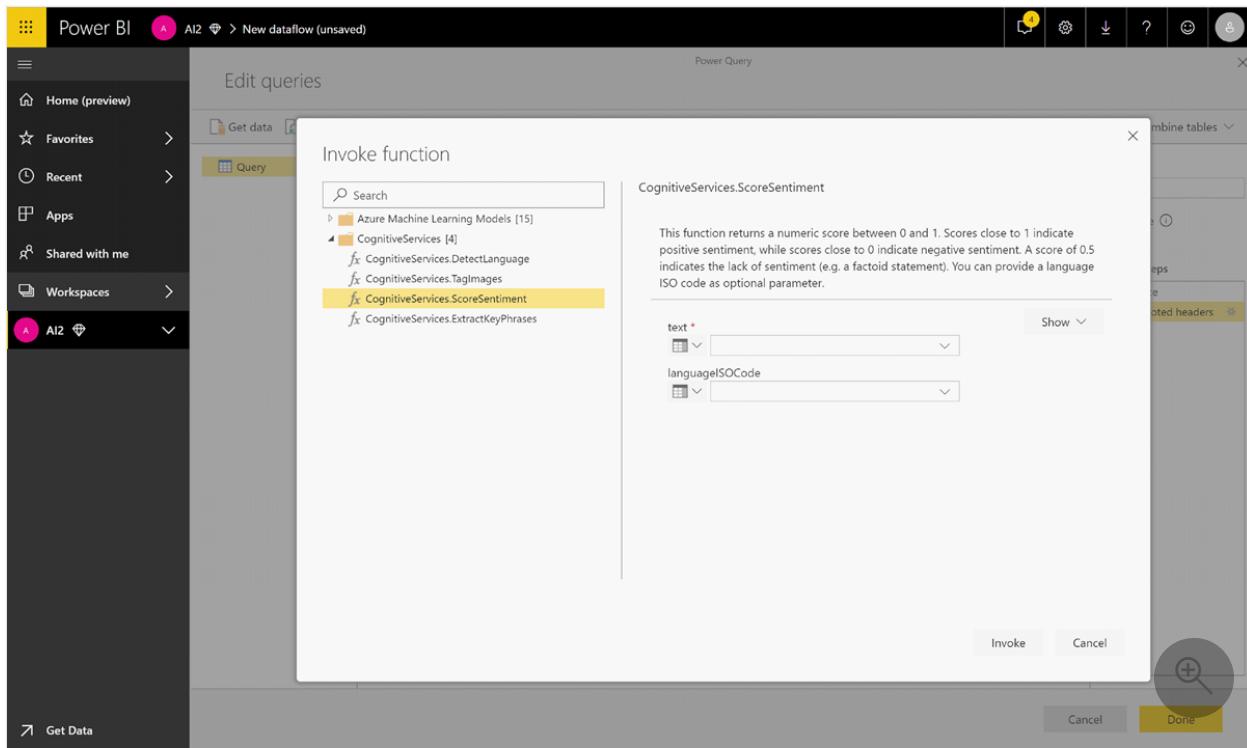


The data is now ready to use for text analytics. You can use Sentiment Scoring and Key Phrase Extraction on the customer comments column.

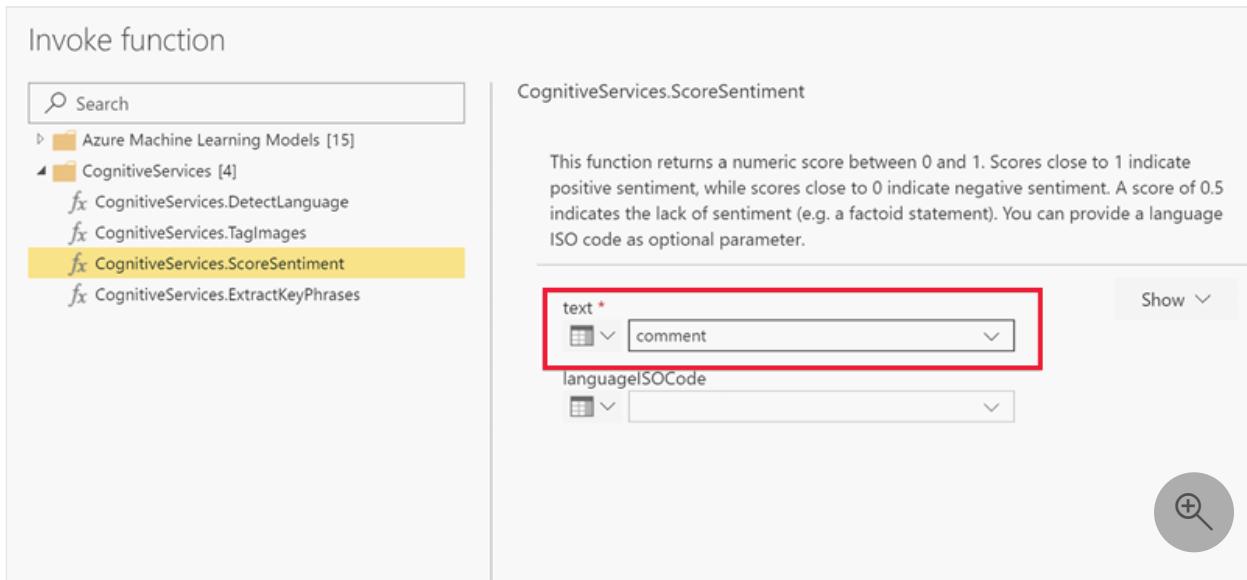
In Power Query Editor, select **AI Insights**

A ^B id	A ^B inreplyto	A ^B name	A ^B user_id	A ^B emailaddress
1	1102	Donald Erie	derie	don@example.com
2	1103	Jake Bering	jakeb	jake.bering@example.com
3	1104	Ann Huron	annh	ahuron@example.com
4	1105	David Columbia	davidc	davidc78@example.com
5	1106	Maria Michigan	mariam	maria@example.com
6	1107	Juan Brazos	juann	juan1@example.com
7	1108	Jake Bering	jakeb	jake.bering@example.com
8	1109	Susan Colorado	susanc	sueco@example.com
9	1110	Ron Puget	ronp	ronnyp@example.com
10	1111	Elena Pecos	elenab	pecos@example.com
11	1112	Darius Willamette	dariusw	go-dawgs@example.com
12	1113	Fabrikam Support	fsupport	support@fabrikam.com
13	1114	Roy Ontario	royo	roy_it@example.com
14	1115	Jake Bering	jakeb	jake.bering@example.com
15	1116	Susan Colorado	susanc	sueco@example.com
16	1117	Fabrikam Sales	fsales	sales@fabrikam.com
17	1118	Wen Niagara	wenn	wen@example.com
18	1119	Susan Colorado	susanc	sueco@example.com
19	1120	Jake Bering	jakeb	jake.bering@example.com
20	1121	Riya Champlain	riyac	riya@example.com

Expand the **Cognitive Services** folder and select the function you would like to use. This example scores the sentiment of the comment column, but you can follow the same steps to try out Language Detection and Key Phrase Extraction.



After you select a function, the required and optional fields appear. To score the sentiment of the example reviews, select the reviews column as text input. Culture information is an optional input and requires an ISO format. For example, enter *en* if you want the text to be treated as English. When the field is left blank, Power BI first detects the language of the input value before it scores the sentiment.



Now select **Invoke** to run the function. The function adds a new column with the sentiment score for each row to the table. You can go back to **AI insights** to extract key phrases of the review text in the same way.

Once you finish the transformations, change the query name to *Customer comments* and select **Done**.

The screenshot shows the 'Edit queries' interface in Power BI Dataflows. A table is displayed with the following columns: A1c subject, A1c comment, A1c spamscore, A1c votes, and CognitiveServices.ScoreSentiment. The 'Name' field in the applied steps pane is highlighted with a red box and set to 'Customer comments'. The 'Applied steps' pane also shows 'Promoted headers' and 'Invoked CognitiveSer...'. The bottom right corner features a 'Done' button.

Next, Save the dataflow and name it *Fabrikam*. Select the Refresh Now button that pops up after you save the dataflow.

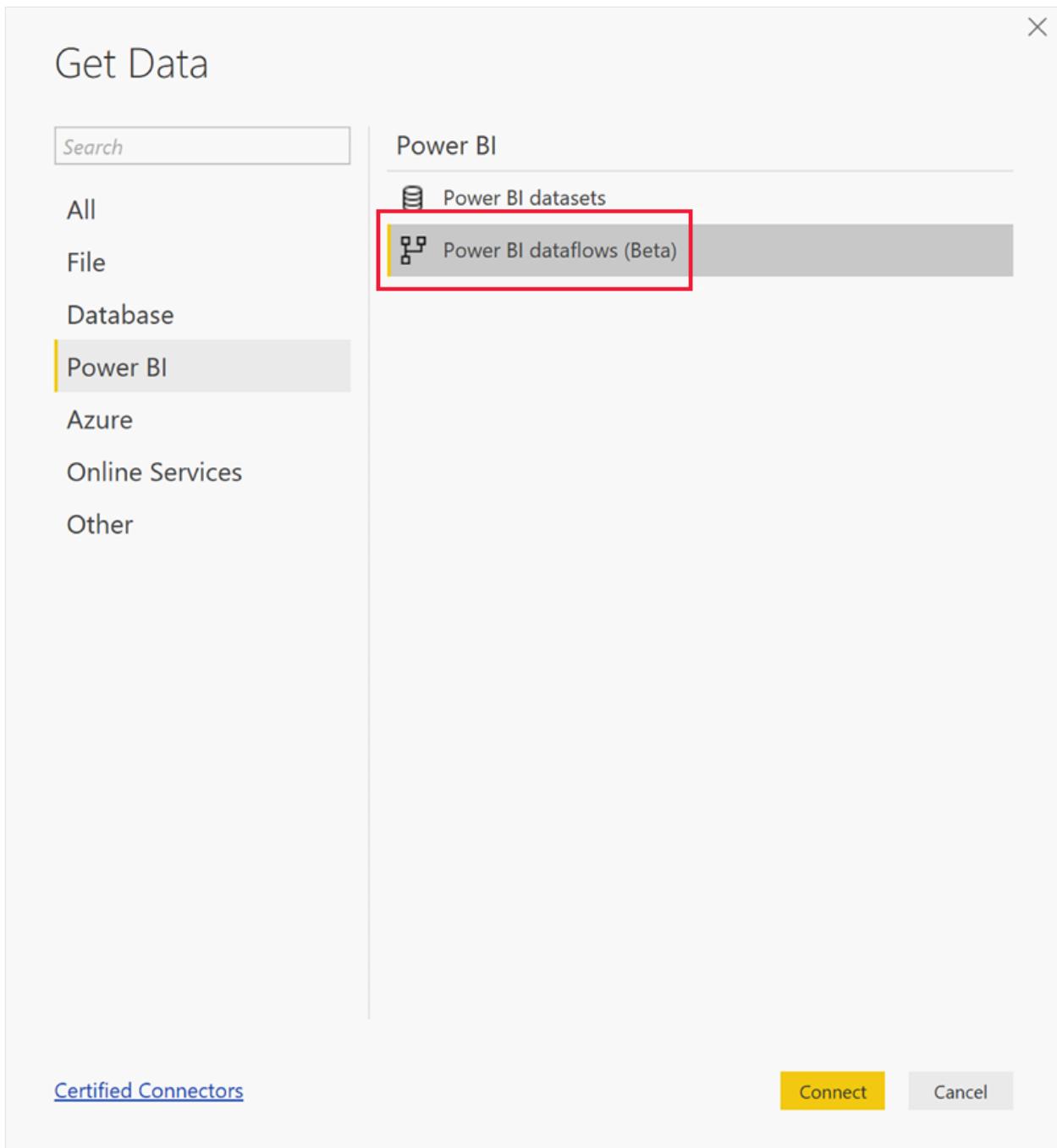
The screenshot shows the 'Save' dialog box in Power BI Dataflows. The 'Save' button is highlighted with a red box. The dialog includes sections for 'ENTITY TYPE' (set to 'Custom') and 'ACTIONS' (with icons for Edit entities, Add entities, Save, and Close).

After you save and refresh the dataflow, you can use it in a Power BI report.

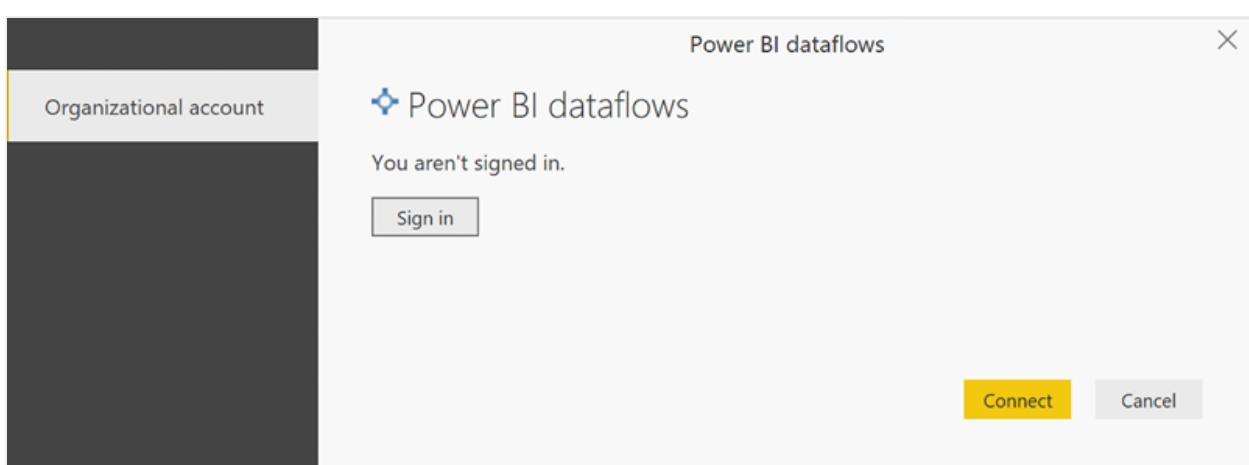
Step 2: Connect from Power BI Desktop

Open Power BI Desktop. In the Home ribbon, select Get data.

Select Power BI and then choose Power BI dataflows. Select Connect.



Sign in with your organization account.



Select the dataflow you created. Navigate to the **Customer comments** table and choose Load.

The screenshot shows the Power BI Navigator interface. On the left, there's a tree view of datasets under 'Fabrikam'. The 'Customer comments' dataset is selected and highlighted with a red box. On the right, there's a table titled 'Customer comments' with columns: spamscore, votes, and CognitiveServices.ScoreSentiment. The table contains several rows of customer comments and their associated scores. At the bottom right of the Navigator window, there are buttons for 'Load' (highlighted with a red box), 'Edit', and 'Cancel'.

Now that the data is loaded, you can start building a report.

Image tagging

In the Power BI service, navigate to a workspace with Premium capacity. Create a new dataflow using the **Create** button in the upper right of the screen.

The screenshot shows the Power BI service workspace. At the top, there's a navigation bar with icons for notifications, settings, download, help, and user profile. Below the navigation bar, there's a 'Create' button (highlighted with a red box) and a 'Publish app' button. A dropdown menu is open from the 'Create' button, showing options: Dashboard, Report, Dataset, Streaming dataset, and Dataflow (which is highlighted with a red box). Below the dropdown, it says 'Showing 0 item(s)'. At the bottom, there are sections for 'ACTIONS', 'OWNER', and 'CLASSIFICATION'.

Select **Add new entities**.

Add entities to start creating your dataflow

Define new entities

Choose a data source to define the entities for your dataflow. You can map your data to [standard Common Data Model entities](#), or define custom entities instead.

[Learn more](#)

Add new entities

Once you're asked to choose a data source, select **Blank query**.

Choose data source

Power Query

All categories File Database Azure Online services Other Search

Data sources

Access File	Excel File	JSON	Text/CSV File	XML File
Amazon Redshift Database	IBM Db2 database Database	MySQL database Database	Oracle database Database	PostgreSQL database Database
SQL Server database Database	Sybase database Database	Teradata database Database	Azure Blobs Azure	Azure Data Explorer Azure
Azure SQL Data Warehouse Azure	Azure SQL database Azure	Azure Tables Azure	Common Data Service for Apps Online services	Microsoft Exchange Online Online services
Salesforce objects Online services	Salesforce reports Online services	SharePoint Online list Online services	Smartsheet Online services	Active Directory Other
OData Other	Blank query Other	Web API Other	Web page Other	Blank table Other

Copy this query in the query editor and select **Next**. You can replace the URL paths with other images or add more rows. The `Web.Contents` function imports the image URL as binary. If you have a data source with images stored as binary, you can also use that directly.

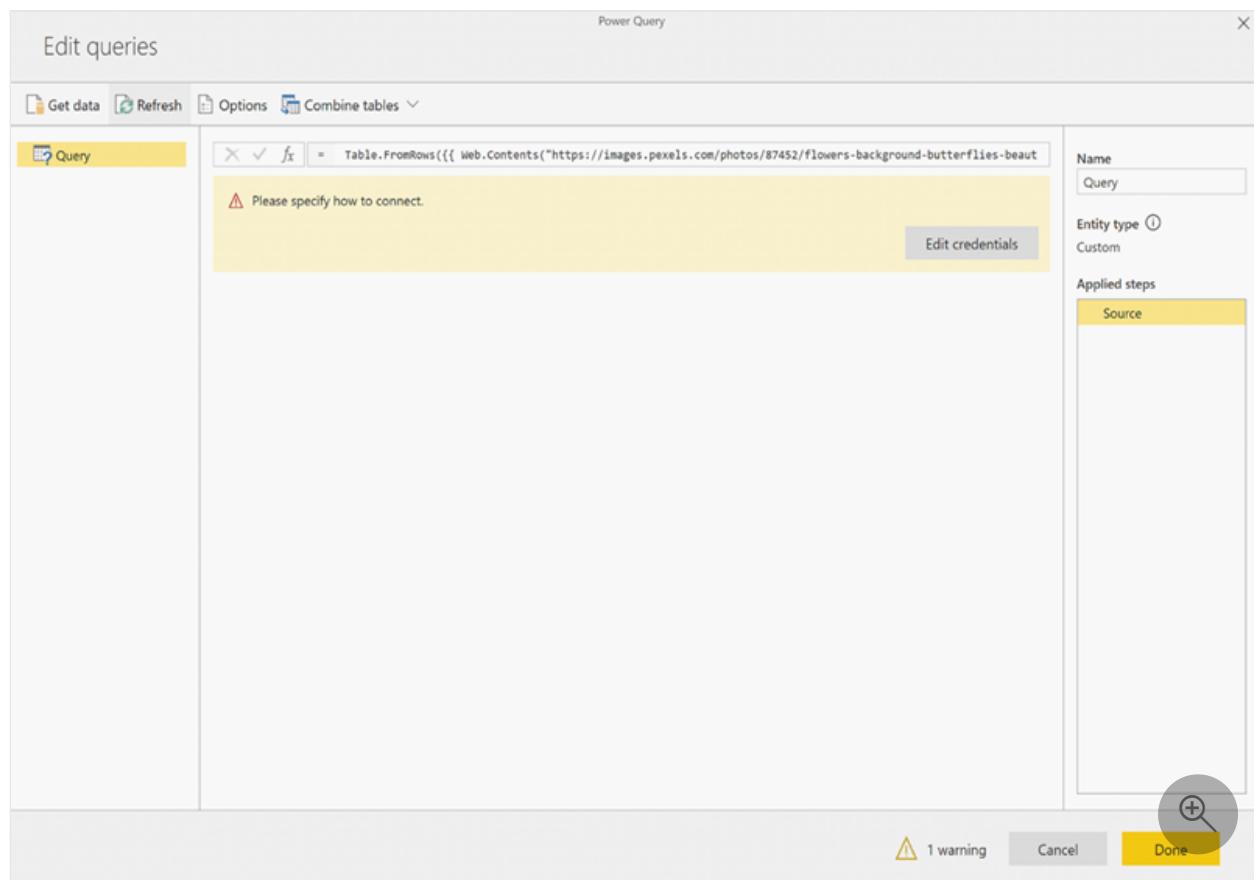
Python

```
let
    Source = Table.FromRows({
        { Web.Contents("https://images.pexels.com/photos/87452/flowers-background-butterflies-beautiful-87452.jpeg") },
        {
            Web.Contents("https://upload.wikimedia.org/wikipedia/commons/5/53/Colosseum_in_Rome%2C_Italy_-_April_2007.jpg") }
    }, { "Image" })
```

in
Source

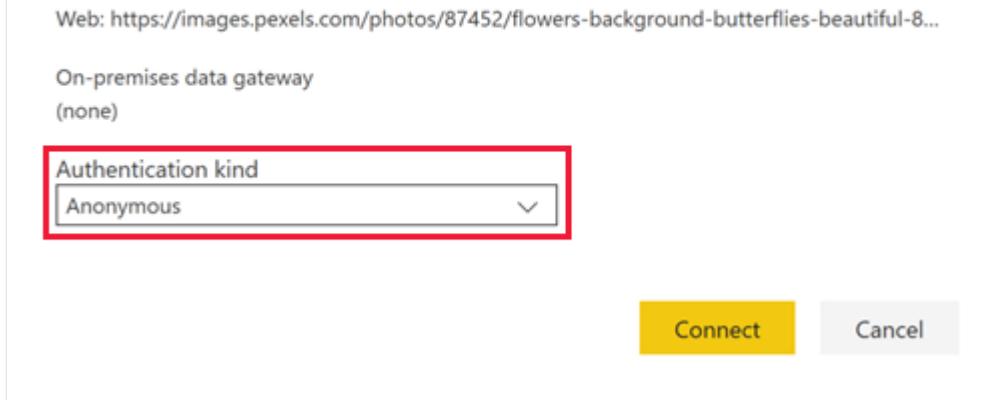


When prompted for credentials, select *anonymous*.



You see the following dialog.

Enter credentials

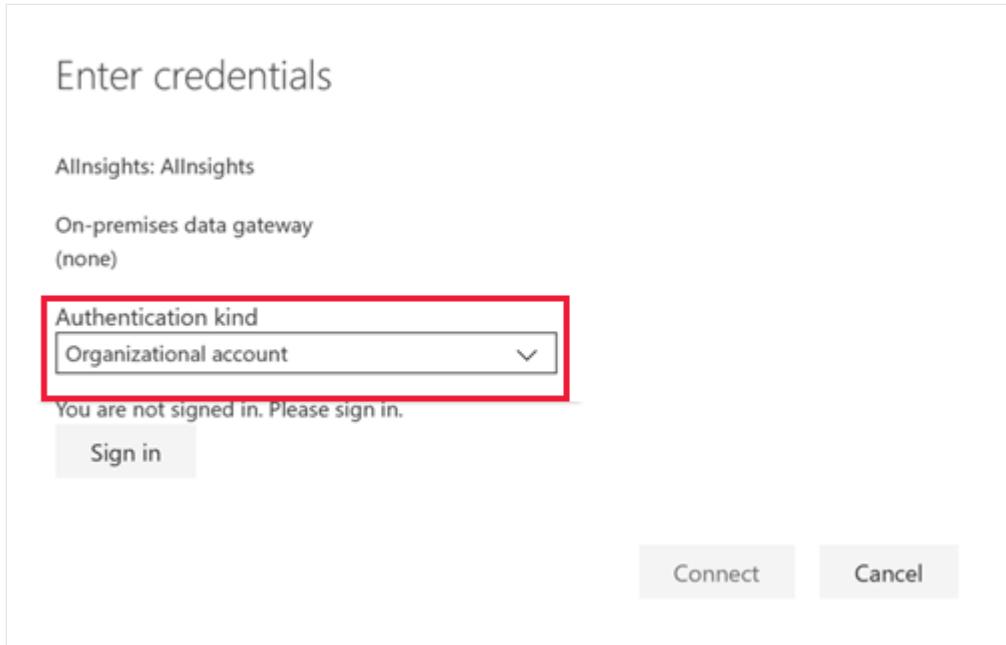


Power BI prompts you for credentials for each web page.

Select **AI Insights** in the query editor.

The screenshot shows the Power Query Editor window titled 'Edit queries'. The ribbon at the top includes 'Get data', 'Refresh', 'Options', 'Manage columns', 'Transform table', 'Reduce rows', 'Add column', 'AI insights' (which is highlighted with a red box), 'Map to standard', and 'Combine tables'. The main area shows a table with two rows, both labeled '[Binary]'. On the right side, there's a panel for 'Name' (set to 'Query'), 'Entity type' (set to 'Custom'), and 'Applied steps' (with 'Source' selected). At the bottom, there are status indicators ('1 warning'), a 'Cancel' button, and a 'Done' button with a magnifying glass icon.

Next, sign in with your **organizational account**.



Select the Tag Images function, enter *[Binary]* in the column field, and enter *en* in the culture info field.

ⓘ Note

You currently cannot pick a column using a dropdown. This issue will be resolved as soon as possible during the private preview.

Invoke function

Search

CognitiveServices.TagImages

image
[Binary]

cultureInfo
en

Show ▾

Invoke Cancel

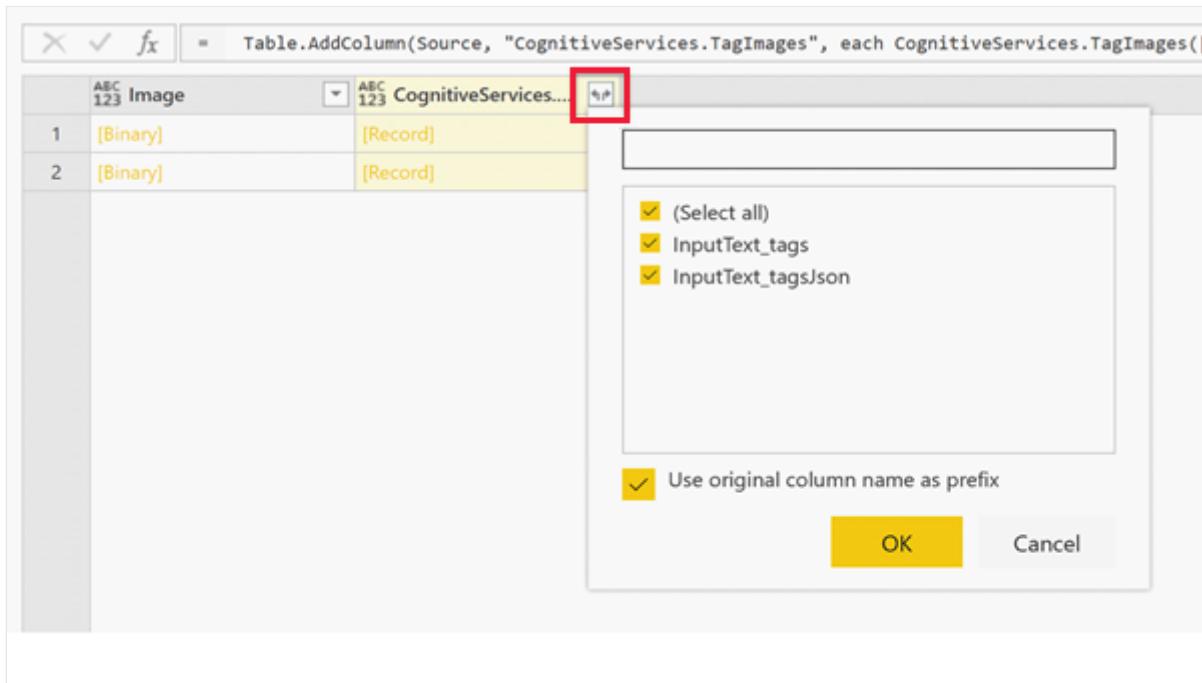
In the function editor, remove the quotation marks around the column name.

⚠ Note

Removing the quotation marks is a temporary workaround. This issue will be resolved as soon as possible during preview.



The function returns a record with both the tags in comma-separated format and as a *json* record. Select the expand-button to add one or both as columns to the table.



Select **Done** and save the dataflow. Once you've refreshed the dataflow one, you can connect to it from Power BI Desktop using the Dataflows connectors.

Clean up resources

When you're done using this tutorial, delete the query by right-clicking the query name in the Power Query Editor and selecting **Delete**.

Limitations

There are some known issues with using Gateway with Cognitive Services. If you need to use a gateway, we recommend creating a dataflow that imports the necessary data by using a gateway first. Then create another dataflow that references the first dataflow to apply these functions.

If your AI work with dataflows fails, you may need to enable Fast Combine when using AI with dataflows. Once you have imported your table and *before* you begin to add AI features, select **Options** from the Home ribbon, and in the window that appears select the checkbox beside *Allow combining data from multiple sources* to enable the feature, then select **OK** to save your selection. Then you can add AI features to your dataflow.

Next steps

In this tutorial, you applied sentiment scoring and image tagging functions on a Power BI dataflow. To learn more about Cognitive Services in Power BI, see the following articles.

- [Azure Cognitive Services](#)
- Get started [with self-service data prep on dataflows](#)
- Learn more about [Power BI Premium](#)

You might also be interested in the following articles.

- [Tutorial: Consume Azure Machine Learning models in Power BI](#)
- [AI with dataflows](#)

Tutorial: Build a machine learning model in Power BI

Article • 05/31/2023

In this tutorial, you use *automated machine learning* to create and apply a binary prediction model in Power BI. You create a Power BI dataflow, and use the entities you define in the dataflow to train and validate a machine learning model directly in Power BI. You then use that model to score new data and generate predictions.

First, you create a binary prediction machine learning model to predict the purchase intent of online shoppers, based on a set of their online session attributes. You use a benchmark machine learning dataset for this exercise. Once you train a model, Power BI automatically generates a validation report that explains the model results. You can then review the validation report and apply the model to your data for scoring.

This tutorial consists of the following steps:

- ✓ Create a dataflow with the input data.
- ✓ Create and train a machine learning model.
- ✓ Review the model validation report.
- ✓ Apply the model to a dataflow entity.
- ✓ Use the scored output from the model in a Power BI report.

Create a dataflow with the input data

Create a dataflow with input data by following these steps.

Get data

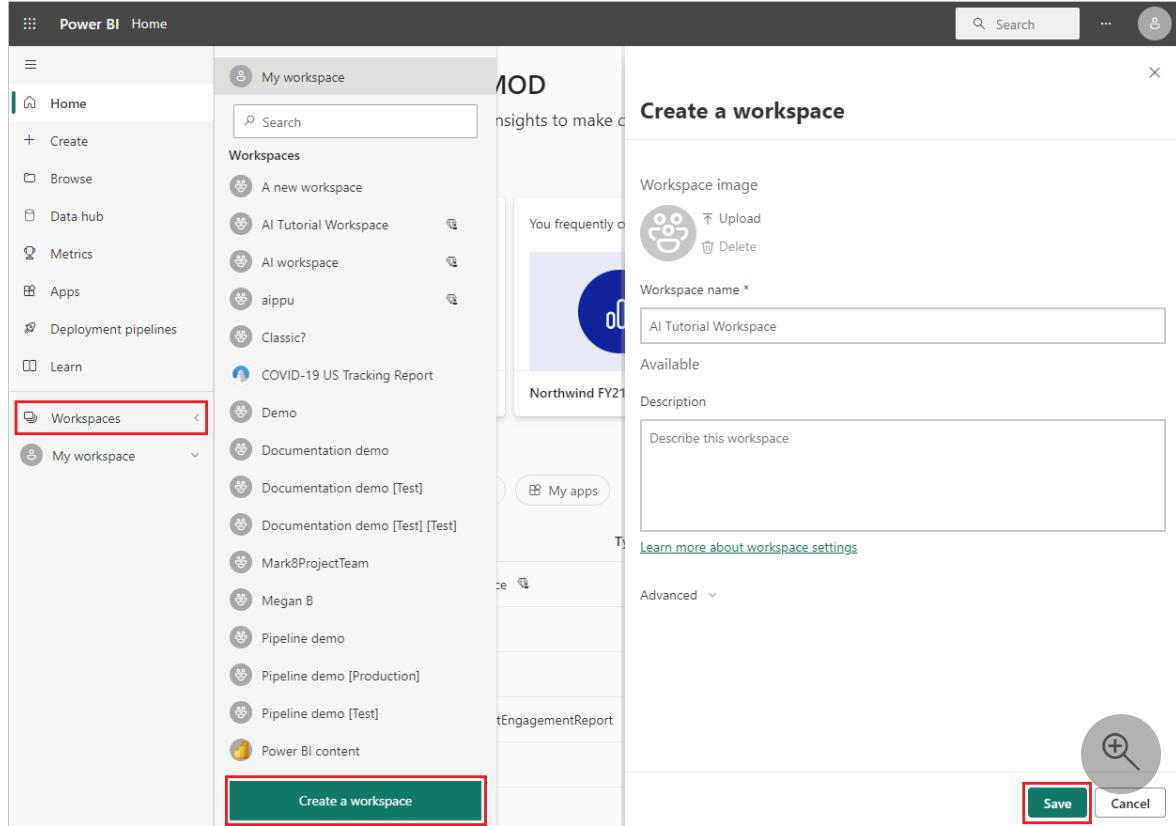
The first step in creating a dataflow is to have your data sources ready. In this case, you use a machine learning dataset from a set of online sessions, some of which culminated in a purchase. The dataset contains a set of attributes about these sessions, which you use to train your model.

You can download the dataset from the UC Irvine website or by downloading the [online_shoppers_intention.csv](#). Later in this tutorial, you connect to the dataset by specifying its URL.

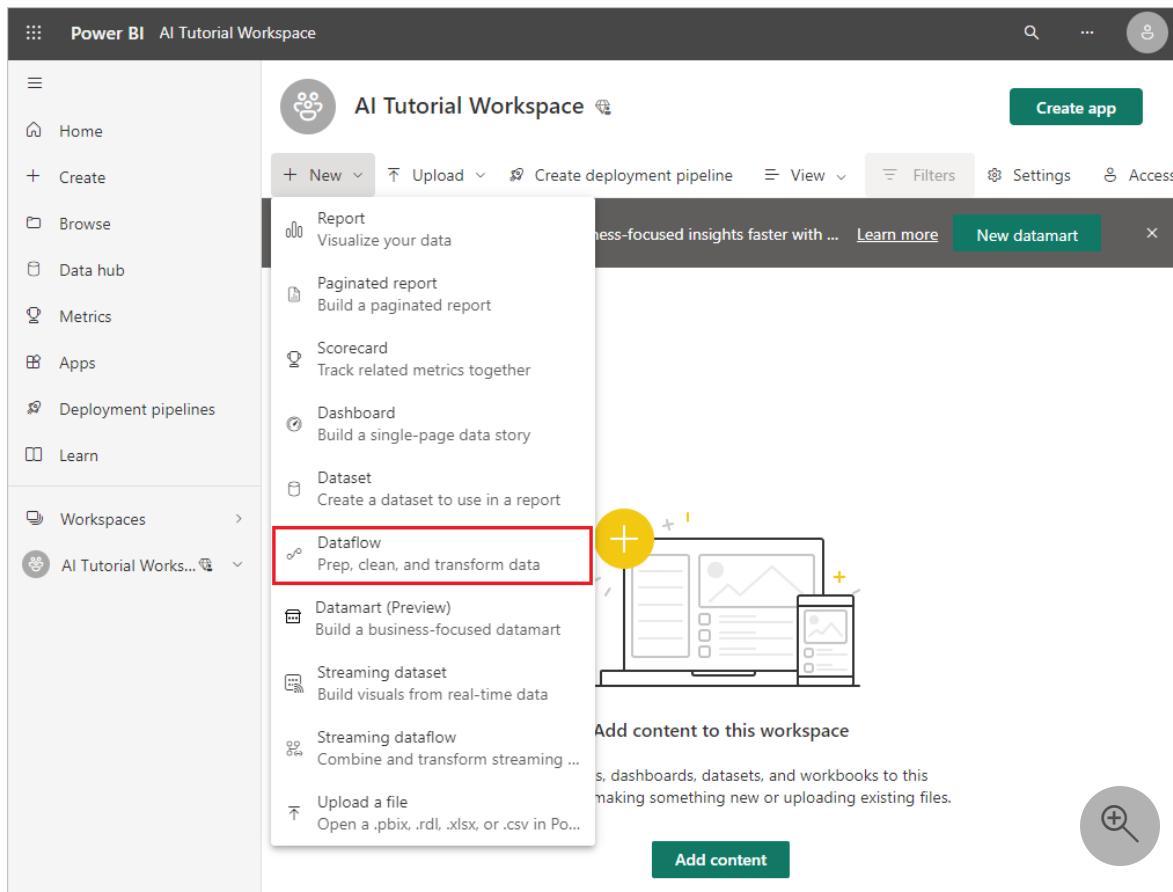
Create the tables

To create the entities in your dataflow, sign into the Power BI service and navigate to a workspace.

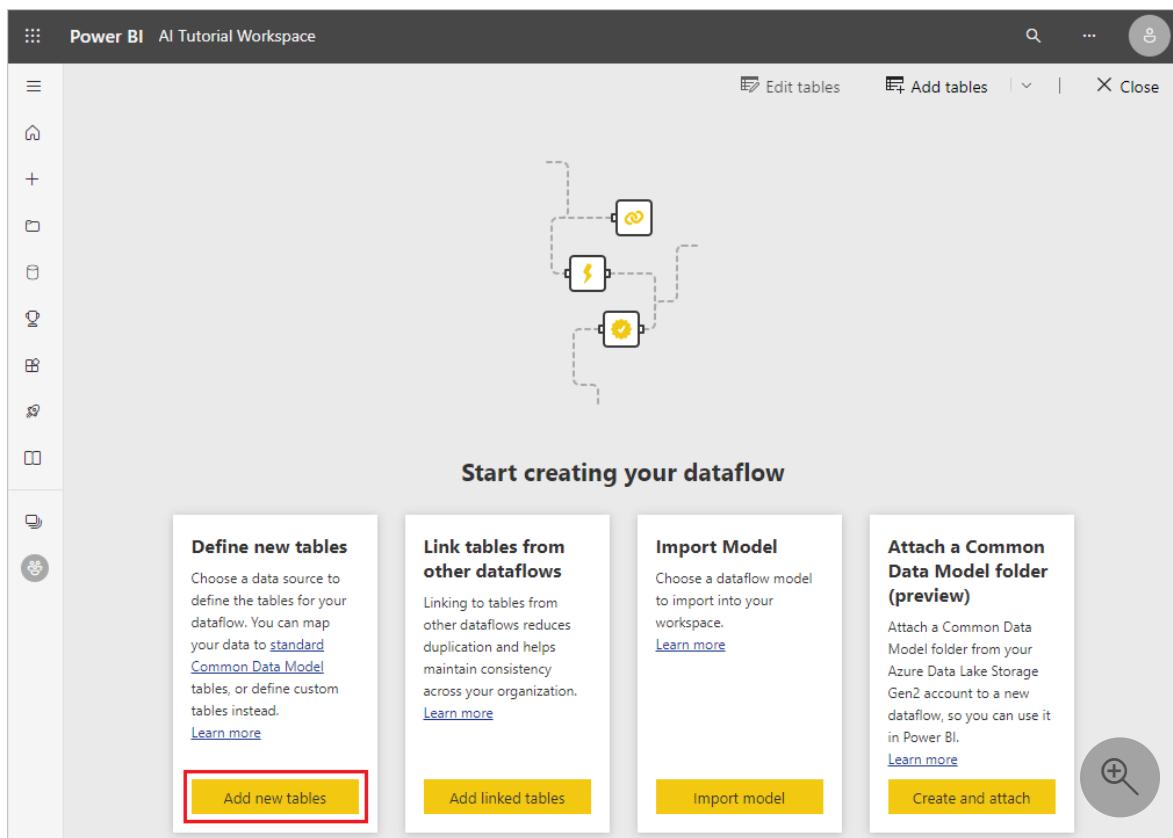
1. If you don't have a workspace, create one by selecting **Workspaces** in the Power BI left navigation pane and selecting **Create a workspace**. In the **Create a workspace** panel, enter a workspace name and select **Save**.



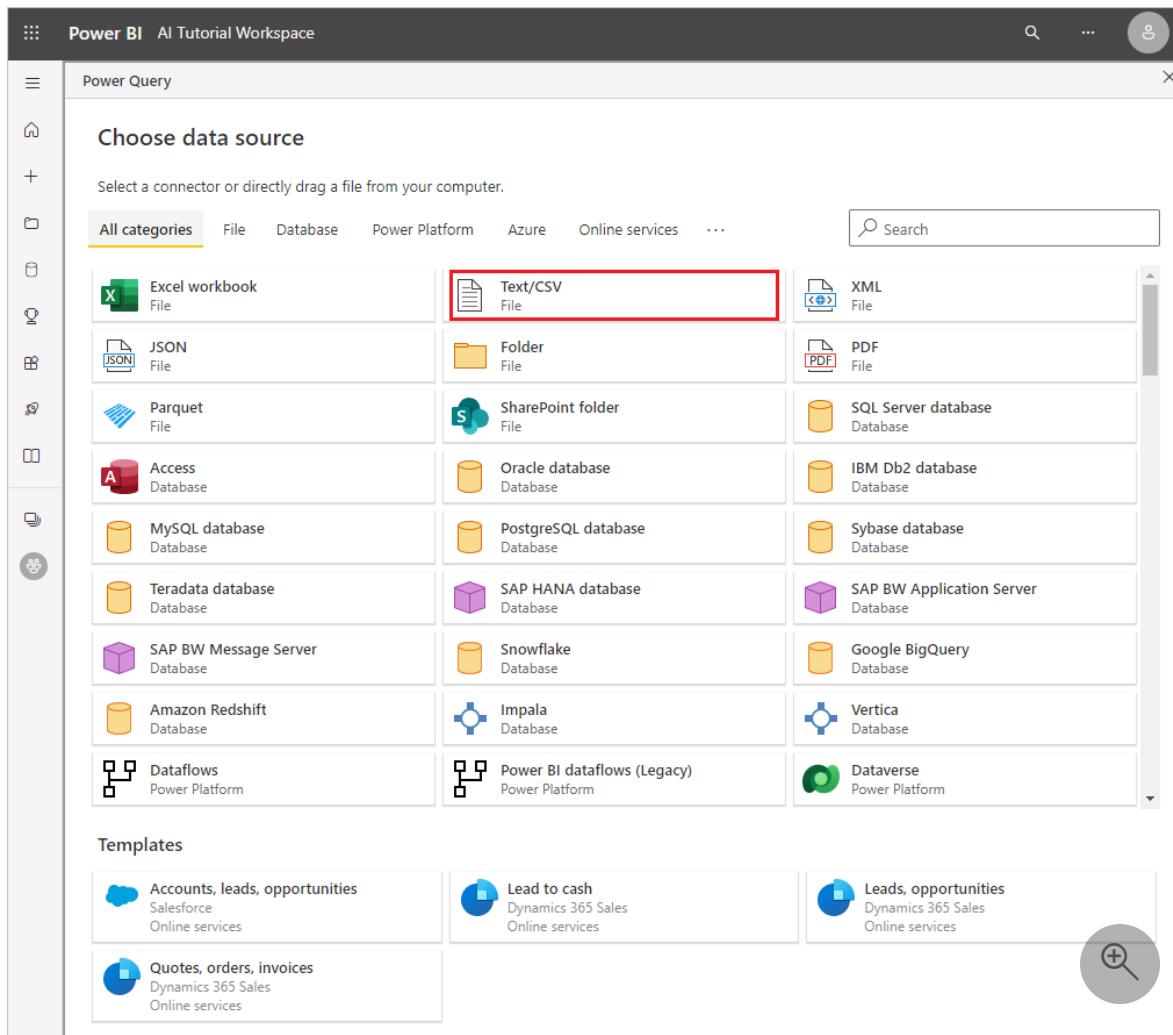
2. Select **New** at the top of the new workspace, and then select **Dataflow**.



3. Select Add new tables to launch a Power Query editor in the browser.

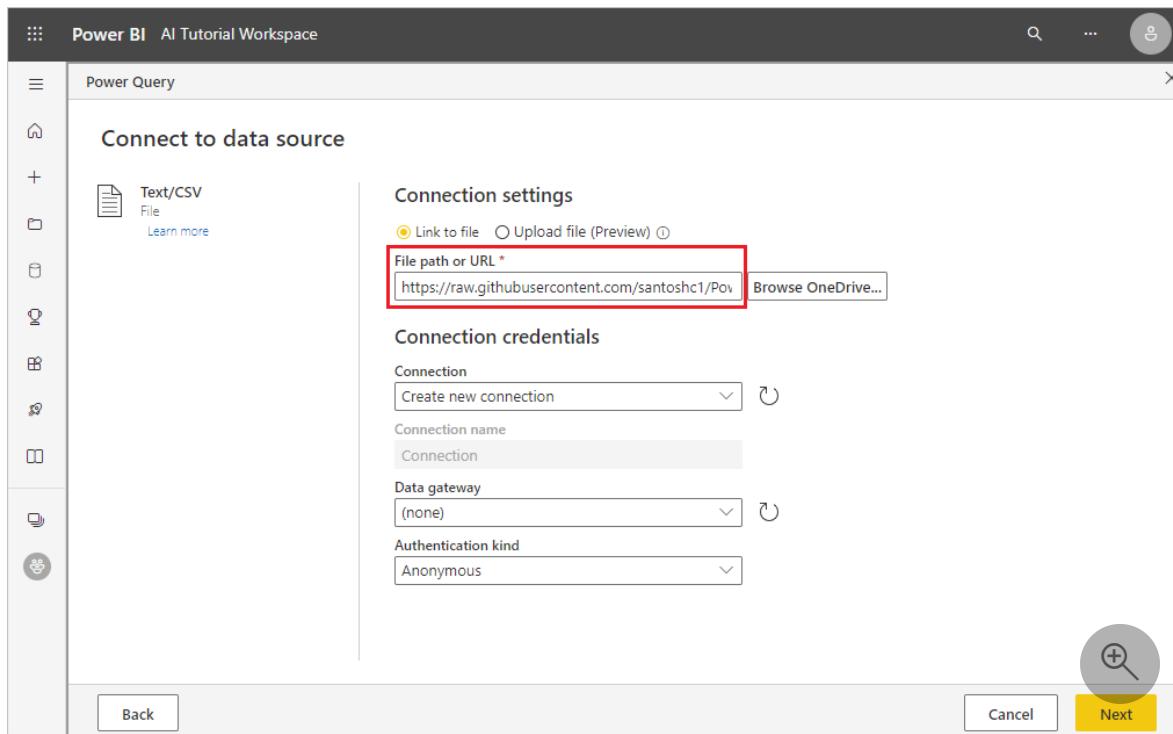


4. On the Choose data source screen, select Text/CSV as the data source.



5. On the **Connect to a data source** page, paste the following link to the *online_shoppers_intention.csv* file into the **File path or URL** box, and then select **Next**.

```
https://raw.githubusercontent.com/santoshc1/PowerBI-AI-samples/master/Tutorial_AutomatedML/online_shoppers_intention.csv
```



6. The Power Query Editor shows a preview of the data from the CSV file. To make changes in the data before loading it, select **Transform data**.

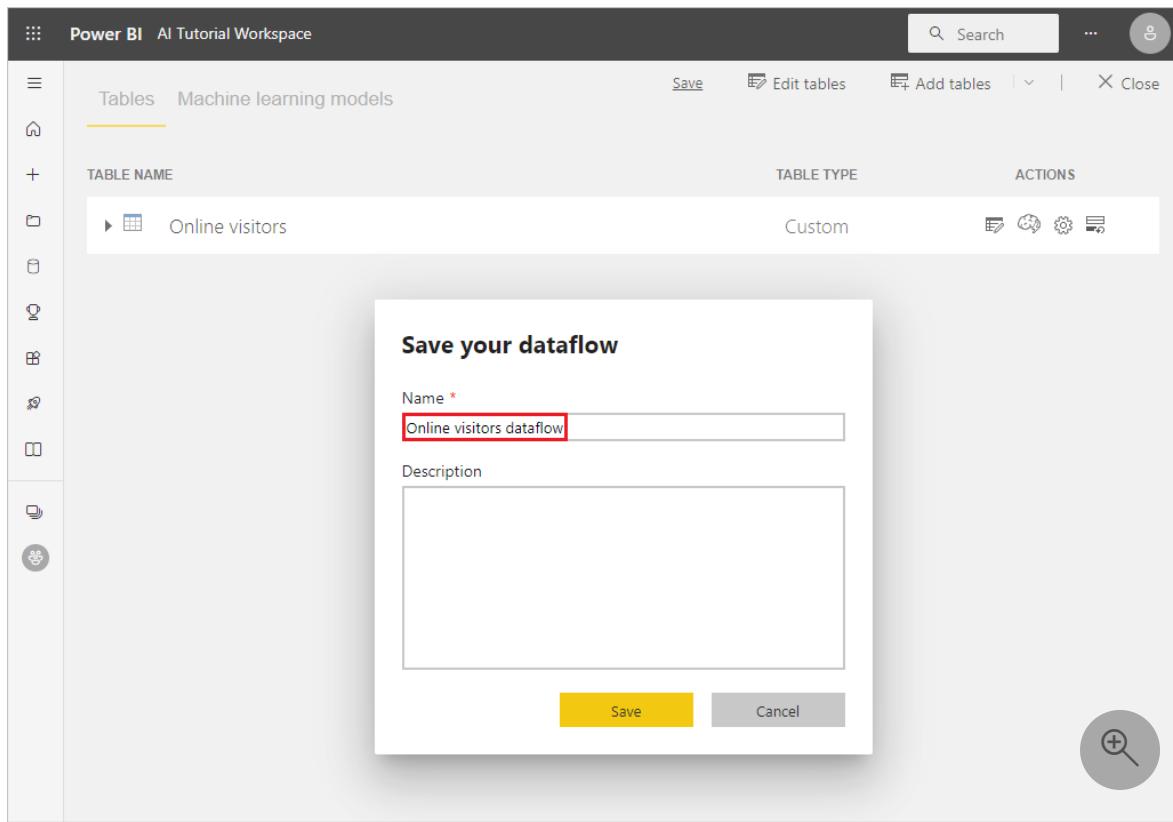
1	2	3	Administrative	1, 2 Administrative_Duration	1, 2 Informational	1, 2 Informational_Duration	1, 2 ProductRelated	1, 2 ProductRelated_Duration	1, 2 BounceRates	1, 2 ExitRates	1, 2 P
1			0	0	2	211.25	144	4627.489571	0.001360544	0.020664031	
7			150.3571429		1	9	221	11431.00124	0.011148992	0.021904109	
3				16	3	86	15	2773.5	0	0.03	
0			0	0	0	0	7	705.8333333	0.028571429	0.085714286	
0			0	0	0	0	7	212.5	0	0.033333333	
0			0	0	0	0	44	615	0.013636364	0.036363636	
3			100.5		0	0	95	1453.640882	0.002105263	0.012761333	
0			0	0	0	0	6	279	0	0.033333333	
0			0	0	0	0	21	1128.5833333	0	0.013043478	
0			0	0	0	0	8	143.5833333	0.014285714	0.05	
7			191.2	4	654.3333333	31	1204.600433	0.000854701	0.013342491		
2			47	0	0	0	39	1743.366667	0	0.016410256	
0			0	0	0	0	10	236	0	0.026666667	
4			724		0	0	2	159	0	0.008	
0			0	0	0	0	2	24	0	0.1	
8			237.5	4	142	30	1569.625	0	0.008108108		
3			66.333333333	0	0	15	424.8333333	0	0.0125		
3			227	0	0	2	62	0	0.025		
1			23	0	0	0	15	656.25	0.028571429	0.028571429	
0			0	0	0	0	29	617.5	0.006896552	0.01954023	
0			0	0	2	39	15	245	0	0.0125	
0			0	0	0	0	3	20	0	0.066666667	
0			0	0	0	0	38	1372	0.005405405	0.018918919	
2			41	0	0	0	7	139	0.022222222	0.055555556	
4			74	2	629	149	5042.458059	0.003870968	0.0072851		

7. Power Query automatically infers the data types of the columns. You can change the data types by selecting the attribute type icon at the tops of the column headers. Change the type of the **Revenue** column to **True/False**.

You can rename the query to a friendlier name by changing the value in the Name box in the right pane. Change the query name to *Online visitors*.

The screenshot shows the Power BI Power Query Editor interface. On the left, there's a sidebar with icons for Home, Transform, Add column, View, and Help. The main area is titled 'Power Query' and shows a 'Queries [1]' list with one item: 'Online visitors'. To the right of the list is a table preview showing 25 rows of data with various columns like 'Region', 'TrafficType', 'VisitorType', 'Weekend', and 'Revenue'. Below the preview, it says 'Completed (0.72 s) Columns: 18 Rows: 99+'. On the far right, there's a 'Query settings' pane with sections for 'Properties' (where 'Name' is set to 'Online visitors') and 'Applied steps' (which includes 'Source', 'Promoted h...', and 'Changed co...'). At the bottom right of the editor are 'Cancel' and 'Save & close' buttons, with 'Save & close' being highlighted with a red box.

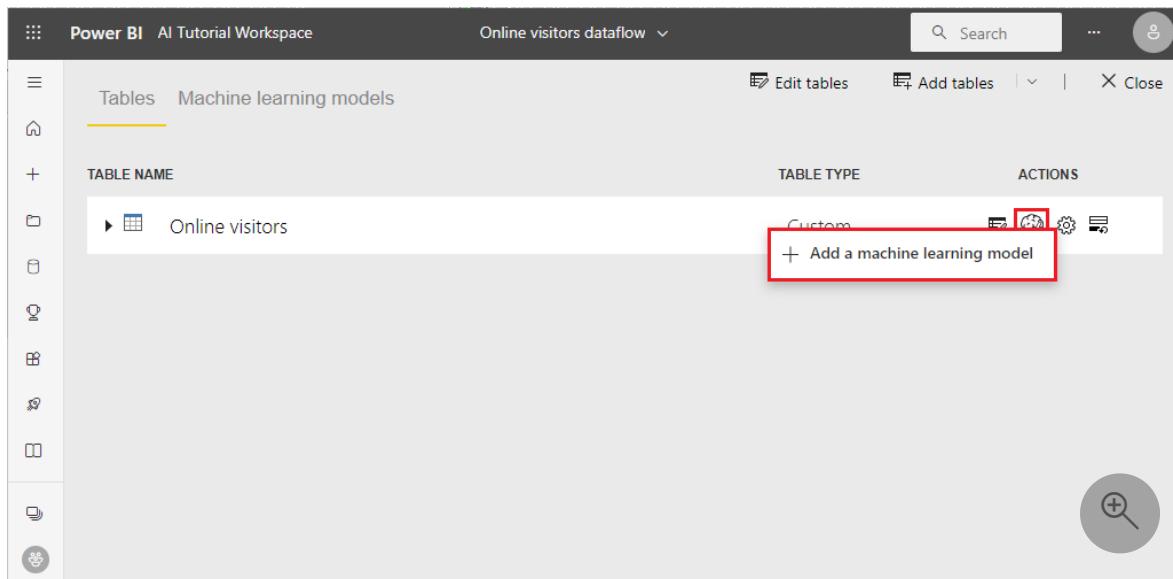
8. Select **Save & close**, and in the dialog box, provide a name for the dataflow and then select **Save**.



Create and train a machine learning model

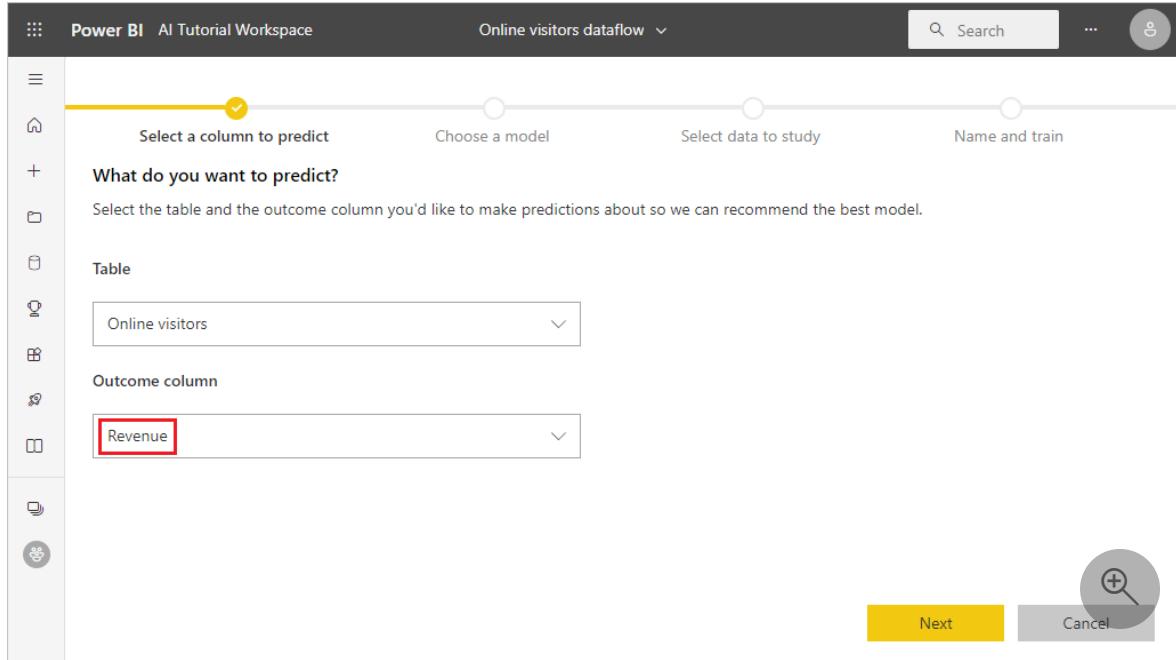
To add a machine learning model:

1. Select the **Apply ML model** icon in the **Actions** list for the table that contains your training data and label information, and then select **Add a machine learning model**.



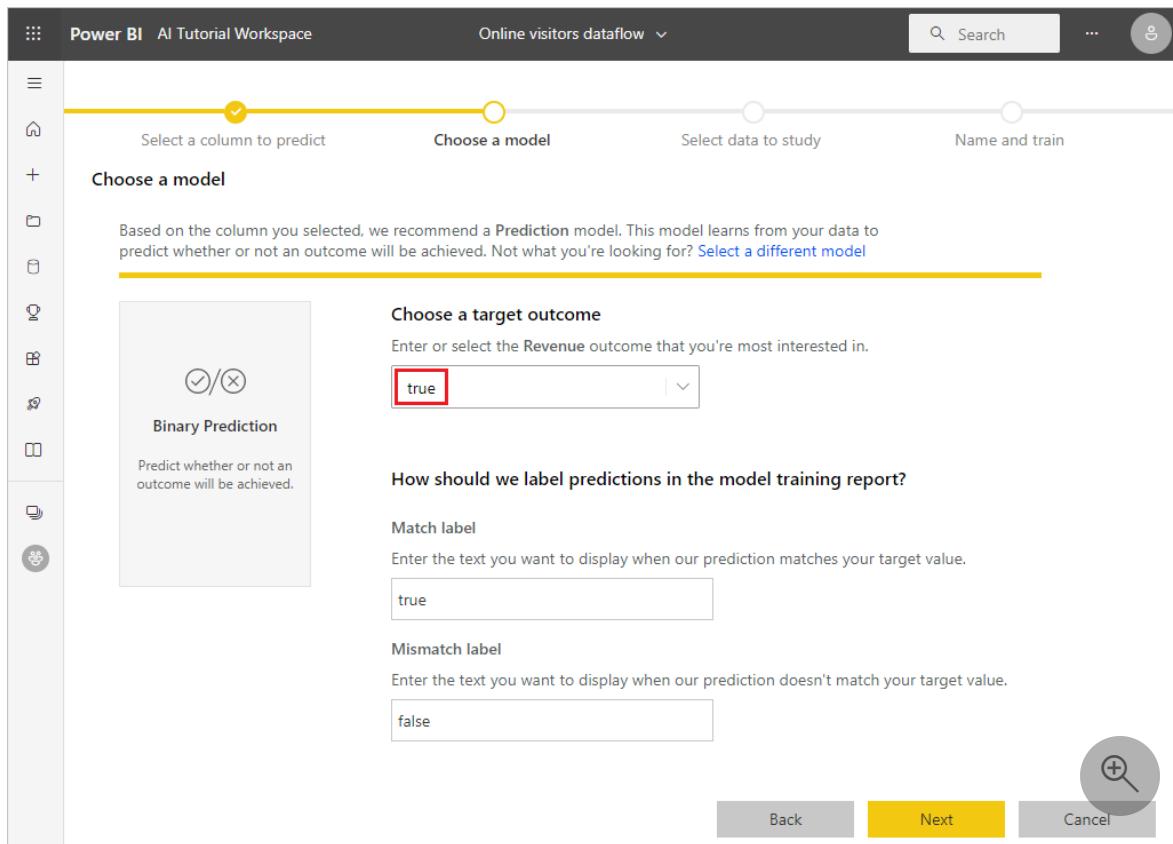
2. The first step to create your machine learning model is to identify the historical data, including the outcome field that you want to predict. The model is created by learning from this data. In this case, you want to predict whether or not visitors are

going to make a purchase. The outcome you want to predict is in the **Revenue** field. Select **Revenue** as the **Outcome column** value, and then select **Next**.

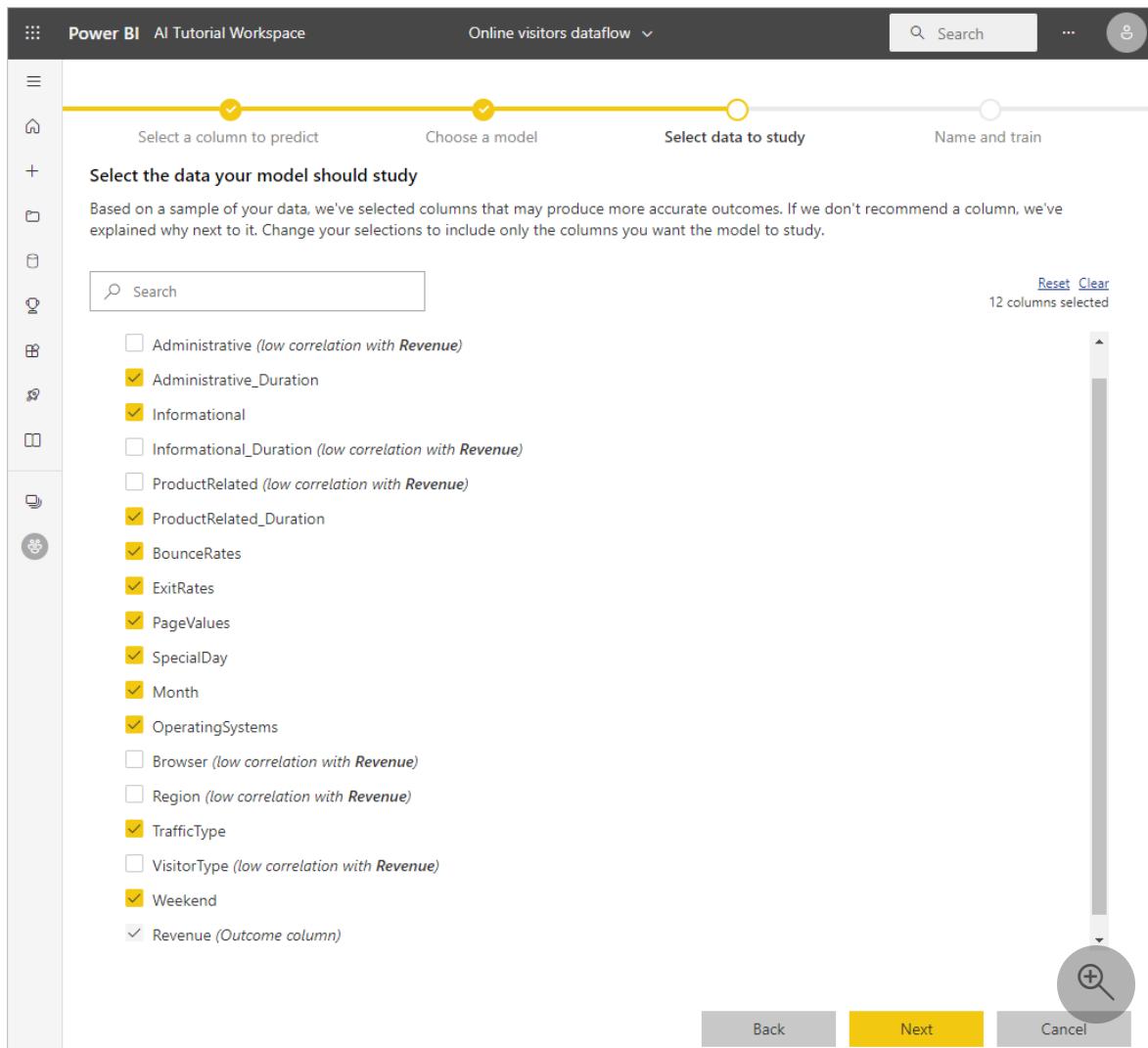


3. Next, you select the type of machine learning model to create. Power BI analyzes the values in the outcome field that you identified, and suggests the types of machine learning models that it can create to predict that field.

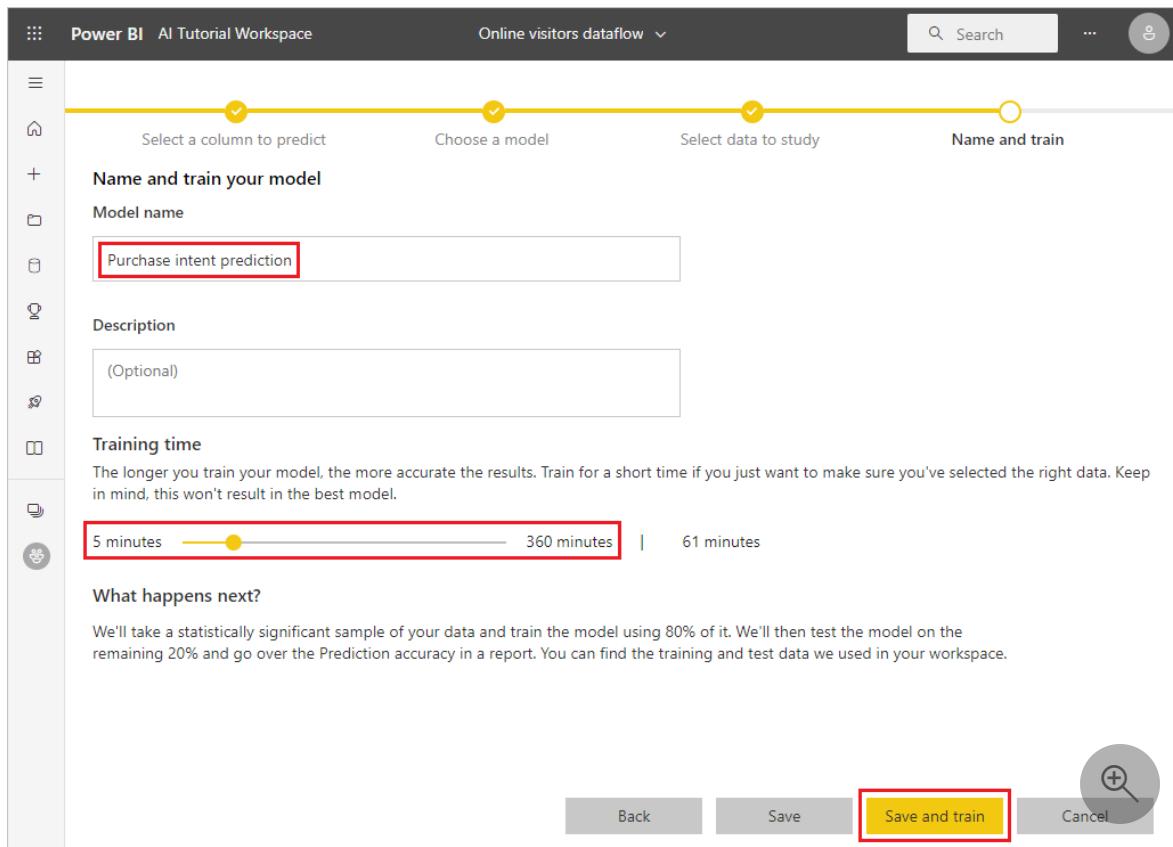
In this case, since you want to predict a binary outcome of whether or not a visitor is going to make a purchase, Power BI recommends **Binary Prediction**. Because you're interested in predicting visitors who are going to make a purchase, select **true** under **Choose a target outcome**. You can also provide different labels to use for the outcomes in the automatically generated report that summarizes the model validation results. Then select **Next**.



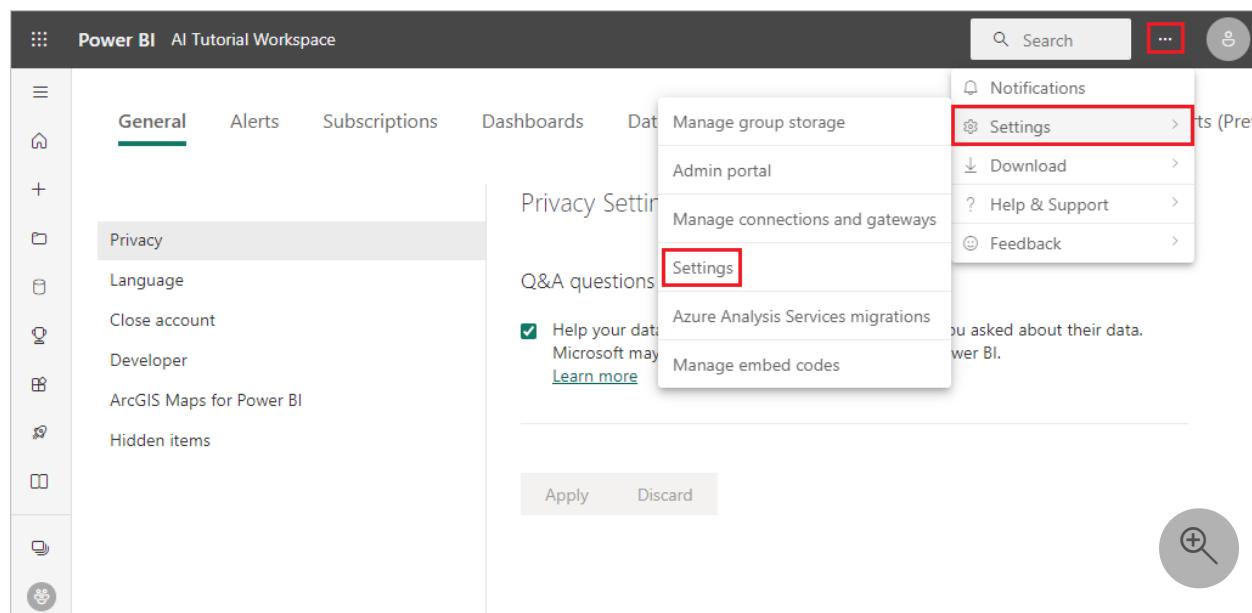
4. Power BI does a preliminary scan of a sample of your data and suggests inputs that might produce more accurate predictions. If Power BI doesn't recommend a column, it explains why not next to the column. You can change the selections to include only the fields you want the model to study by selecting or deselecting the checkboxes next to column names. Select **Next** to accept the inputs.



5. In the final step, name the model *Purchase intent prediction*, and choose the amount of time to spend in training. You can reduce the training time to see quick results or increase the time to get the best model. Then select **Save and train** to start training the model.



If you get an error similar to **Credentials not found for data source**, you need to update your credentials so Power BI can score the data. To update your credentials, select **More options ...** in the header bar and then select **Settings > Settings**.



Select your dataflow under **Dataflows**, expand **Data source credentials**, and then select **Edit credentials**.

Settings for Online visitors dataflow

This dataflow has been last modified by [admin@contoso.com](#)

Refresh in progress... [Refresh history](#)

Gateway Connection

Dataflow on-premises gateways are currently editable through the Power Query Online experience. [Learn how to edit](#)

Data source credentials

AIFunctions	Edit credentials	Show in lineage view
AllInsightsInProc	Edit credentials	Show in lineage view
PowerBI	Edit credentials	Show in lineage view
Web	Edit credentials	Show in lineage view

▷ Scheduled refresh

▷ Enhanced compute engine settings

▷ Endorsement

Track training status

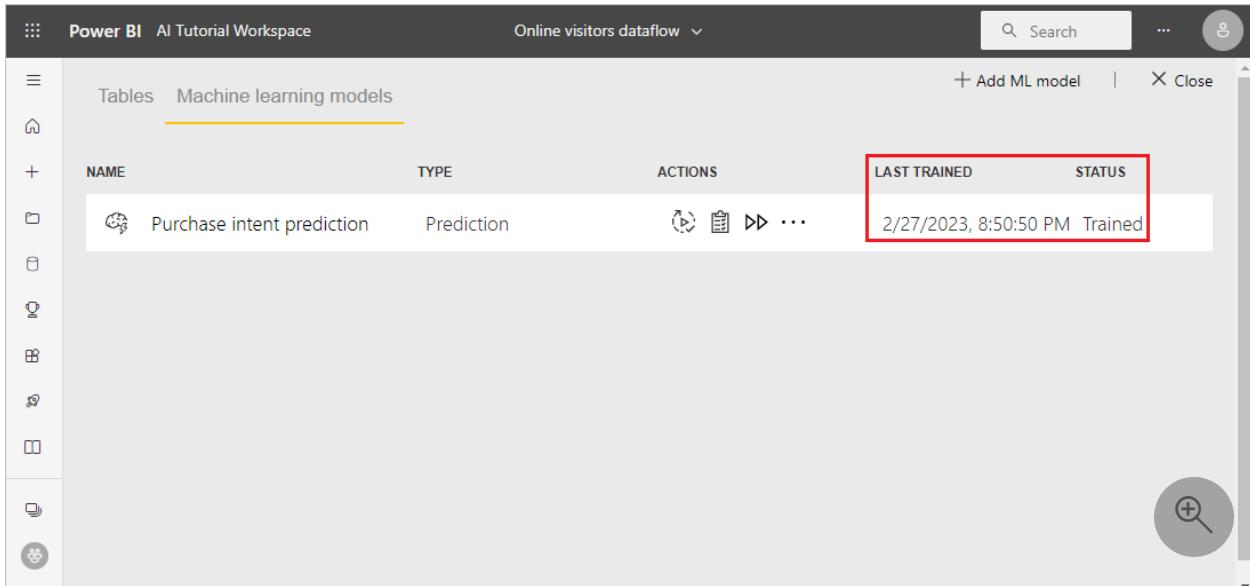
The training process begins by sampling and normalizing your historical data and splitting your dataset into two new entities: **Purchase Intent Prediction Training Data** and **Purchase Intent Prediction Testing Data**.

Depending on the size of the dataset, the training process can take anywhere from a few minutes up to the training time you selected. You can confirm that the model is being trained and validated through the status of the dataflow. The status appears as a data refresh in progress in the **Datasets + dataflows** tab of the workspace.

Name	Type	Owner	Refreshed	Next refresh
Online visitors dataflow	Dataflow	MOD Administrator	2/27/23, 8:40:03 PM	N/A

You can see the model in the **Machine learning models** tab of the dataflow. **Status** indicates whether the model has been queued for training, is under training, or is

trained. Once the model training is completed, the dataflow displays an updated **Last trained** time and a status of **Trained**.



NAME	TYPE	ACTIONS	LAST TRAINED	STATUS
Purchase intent prediction	Prediction	View training report More	2/27/2023, 8:50:50 PM	Trained

Review the model validation report

To review the model validation report, in the **Machine learning models** tab, select the **View training report** icon under **Actions**. This report describes how your machine learning model is likely to perform.

In the **Model Performance** page of the report, select **See top predictors** to view the top predictors for your model. You can select one of the predictors to see how the outcome distribution is associated with that predictor.

Power BI AI Tutorial Workspace Online visitors dataflow Search ...

Purchase intent prediction model training report This report summarizes the model performance and training details and enables you find an optimal threshold for defining your business outcome.

PERFORMANCE

How the model was evaluated

The model predicted Revenue probabilities for a test set of 2466 records and compared the predicted outcomes (based on the selected threshold) to the historical outcomes.

Model performance

The Area under the curve (AUC) observed on the test set is :

94%

Different features have varying influence on the predicted outcome. Click below for details.

See top predictors

Top predictors by influence

PageValues	Month is Nov
BounceRates	Month is May
Month is Mar	ProductRelated_Duration
Administrative_Duration	ExitRates
TrafficType is 1	TrafficType is 2

Percentage of true records for each feature value

Percentage of target outcome records

156.89 to 209.19
More than 209.19
52.3 to 104.6
104.6 to 156.89
52.3 and less

Model Performance Accuracy Report Training Details

You can use the **Probability Threshold** slicer on the **Model Performance** page to examine the influence of model **Precision** and **Recall** on the model.

Power BI AI Tutorial Workspace Online visitors dataflow Search ...

Purchase intent prediction model training report This report summarizes the model performance and training details and enables you find an optimal threshold for defining your business outcome.

MODEL PERFORMANCE

How the model was evaluated

The model predicted Revenue probabilities for a test set of 2466 records and compared the predicted outcomes (based on the selected threshold) to the historical outcomes.

Model performance

The Area under the curve (AUC) observed on the test set is :

94%

Different features have varying influence on the predicted outcome. Click below for details.

See top predictors

Predicted true Predicted false

Actual true	353.00	22.00
Actual false	585.00	1.51K

36% of records predicted as true are likely to actually be true

Precision

95% of records that are actually true are likely to be predicted as true

Recall

Probability Threshold

0.00 0.35 Increase Recall Increase Precision

Model Performance Accuracy Report Training Details

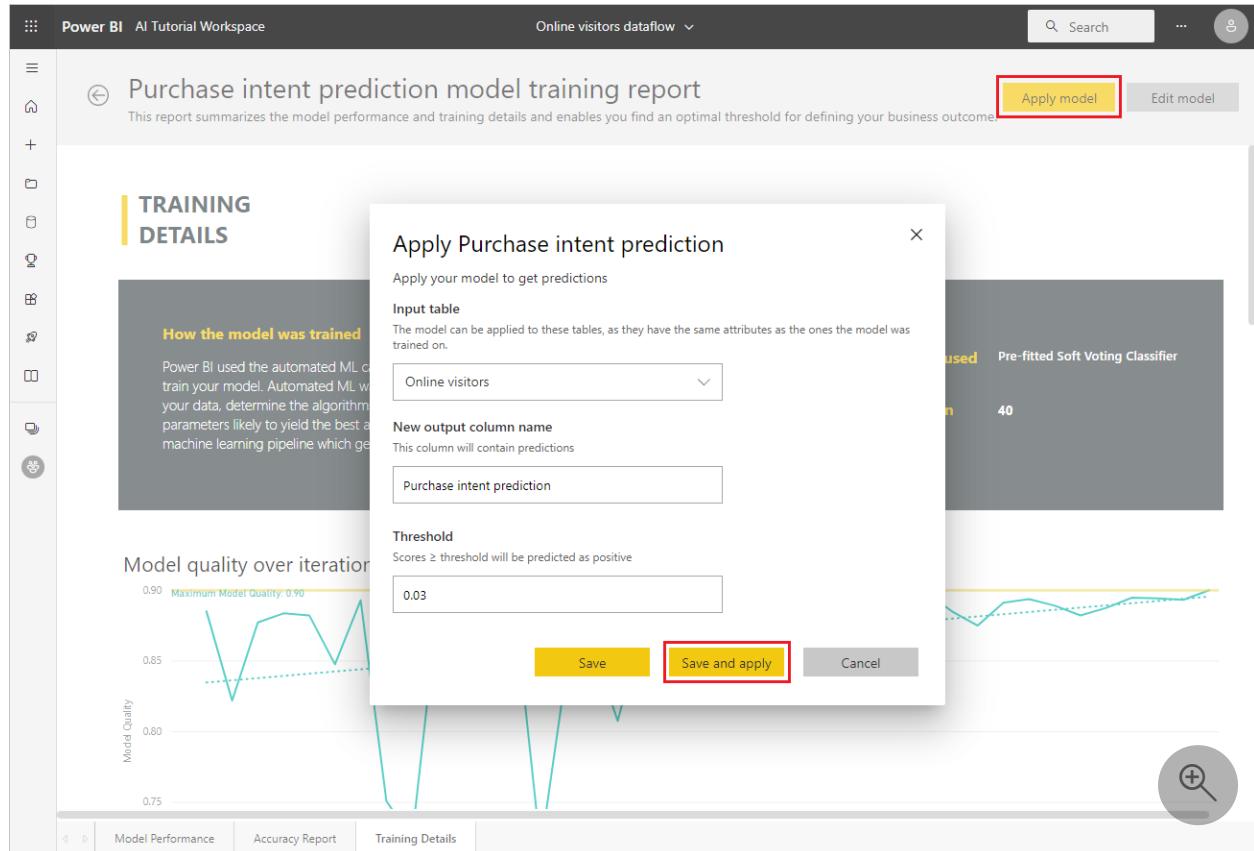
The other pages of the report describe the statistical performance metrics for the model.

The report also includes a **Training Details** page that describes the **Iterations run**, how features were extracted from the inputs, and the hyperparameters for the **Final model**.

used.

Apply the model to a dataflow entity

Select the **Apply model** button at the top of the report to invoke this model. In the **Apply** dialog, you can specify the target entity that has the source data to apply the model to. Then select **Save and apply**.



Applying the model creates two new tables, with the suffixes **enriched <model_name>** and **enriched <model_name> explanations**. In this case, applying the model to the **Online visitors** table creates:

- **Online visitors enriched Purchase intent prediction**, which includes the predicted output from the model.
- **Online visitors enriched Purchase intent prediction explanations**, which contains top record-specific influencers for the prediction.

Applying the binary prediction model adds four columns: **Outcome**, **PredictionScore**, **PredictionExplanation**, and **ExplanationIndex**, each with a **Purchase intent prediction** prefix.

Tables Machine learning models

Edit tables Add tables Close

TABLE NAME	TABLE TYPE	ACTIONS
ExitRates	Double	
PageValues	Double	
SpecialDay	Double	
Month	String	
OperatingSystems	Int64	
Browser	Int64	
Region	Int64	
TrafficType	Int64	
VisitorType	String	
Weekend	Boolean	
Revenue	Boolean	
Purchase intent prediction.Outcome	Boolean	
Purchase intent prediction.PredictionScore	Decimal	
Purchase intent prediction.PredictionExplanation	String	
Purchase intent prediction.ExplanationIndex	Int64	

Online visitors enriched Purchase Custom



Once the dataflow refresh completes, you can select the **Online visitors enriched Purchase intent prediction** table to view the results.

The screenshot shows the Power Query Editor interface. In the center, there is a table view with columns labeled 'je', '1,2 Purchase intent prediction.ExplanationIndex', 'Purchase intent prediction.Outcome', 'Purchase intent prediction.PredictionScore', and 'Purchase intent prediction.PredictionExplanation'. The table contains 99 rows of data. On the right side of the editor, there is a vertical pane titled 'Query settings' which includes sections for 'Properties' (Name: Online visitors enriched ...), 'Entity type' (Custom), and 'Applied steps'. A red box highlights the 'Applied steps' section, specifically the 'fx Enriched re...' step. At the bottom right of the editor, there are buttons for 'Step', 'Cancel', and 'Save & close'.

You can also invoke any automated machine learning model in the workspace directly from the Power Query Editor in your dataflow. To access the automated machine learning models, select **Edit** for the table that you want to enrich with insights from your automated machine learning model.

The screenshot shows the Power BI AI Tutorial Workspace. On the left, there is a navigation bar with icons for Home, Transform, Add column, View, Help, Get data, Enter data, Refresh, Choose columns, Sort, Filter, ABC, and 123. The main area is titled 'Power BI AI Tutorial Workspace' and shows the 'Machine learning models' page. It lists five tables under 'TABLE NAME': 'Online visitors', 'Purchase intent prediction Training Data', 'Purchase intent prediction Testing Data', 'Online visitors enriched Purchase intent prediction', and 'Online visitors enriched Purchase intent explanations'. Each table has a 'TABLE TYPE' of 'Custom' and a set of actions icons. A red box highlights the 'Edit' icon for the first table, 'Online visitors'. At the top right, there are buttons for 'Changes saved', 'Edit tables', 'Add tables', and 'Close'. A large circular icon with a magnifying glass is overlaid on the bottom right of the table list.

In the Power Query Editor, select **AI insights** in the ribbon.

The screenshot shows the Power Query interface with the 'AI insights' button highlighted in the top right. The main area displays a table with four rows and various columns. The 'Applied steps' pane on the right lists several steps, including 'Source', 'AddExplana...', 'fx Invoked Pur...', 'DataflowPr...', 'Workflow...', 'Dataflow...', 'EnrichedPre...', and 'fx Enriched re...'. The 'Properties' pane on the far right shows the query name as 'Online visitors enriched ...'.

On the **AI insights** screen, select the **Power BI Machine Learning Models** folder from the navigation pane. The list shows all the machine learning models you have access to as Power Query functions. The input parameters for the machine learning model automatically map as parameters of the corresponding Power Query function. The automatic parameter mapping happens only if the names and data types of the parameter are the same.

To invoke a machine learning model, you can select any of the selected model's columns as an input in the dropdown list. You can also specify a constant value to use as an input by toggling the column icon next to the input line.

The screenshot shows the 'AI insights' dialog box. In the center, there is a preview of a table with columns: '1.2 Administrative Duration', '1.2 ProductRelated Duration', '1.2 BounceRates', '1.2 ExitRates', '1.2 PageValues', '1.2 SpecialDay', 'A_B Month', and '1.2 OperatingSystems'. To the left, the 'Queries [8]' pane shows 'Purchase intent prediction [1]' and 'Online visitors [1]'. The 'Applied steps' pane on the right shows 'Source', 'Promoted h...', 'Changed co...', and 'Changed co...'. The 'Properties' pane on the far right shows the query name as 'Online visitors'.

Select **Apply** to view the preview of the machine learning model output as new columns in the table. You also see the model invocation under **Applied steps** for the query.

After you save your dataflow, the model automatically invokes when the dataflow refreshes, for any new or updated rows in the entity table.

Using the scored output from the model in a Power BI report

To use the scored output from your machine learning model, you can connect to your dataflow from Power BI Desktop by using the **Dataflows** connector. You can now use the **Online visitors enriched Purchase intent prediction** table to incorporate the predictions from your model in Power BI reports.

Limitations

There are some known issues with using gateways with automated machine learning. If you need to use a gateway, it's best to create a dataflow that imports the necessary data via the gateway first. Then create another dataflow that references the first dataflow to create or apply these models.

If your AI work with dataflows fails, you may need to enable Fast Combine when using AI with dataflows. Once you have imported your table and *before* you begin to add AI features, select **Options** from the Home ribbon, and in the window that appears select the checkbox beside *Allow combining data from multiple sources* to enable the feature, then select **OK** to save your selection. Then you can add AI features to your dataflow.

Next steps

In this tutorial, you created and applied a binary prediction model in Power BI by doing these steps:

- Created a dataflow with the input data.
- Created and trained a machine learning model.
- Reviewed the model validation report.

- Applied the model to a dataflow entity.
- Learned how to use the scored output from the model in a Power BI report.

For more information about Machine Learning automation in Power BI, see [Automated machine learning in Power BI](#).

Tutorial: Consume Azure Machine Learning models in Power BI

Article • 10/05/2022

This tutorial walks you through creating a Power BI report based on a machine learning model. By the end of this tutorial, you'll be able to:

- ✓ Score machine learning models (deployed using Azure Machine Learning) in Power BI.
- ✓ Connect to an Azure Machine Learning model in the Power Query Editor.
- ✓ Create a report with a visualization based on that model.
- ✓ Publish that report to the Power BI service.
- ✓ Set up scheduled refresh for the report.

ⓘ Note

Currently, Online endpoints (v2 endpoint) are not supported in Power BI yet. ACI/AKS web services (v1 web services) are only supported.

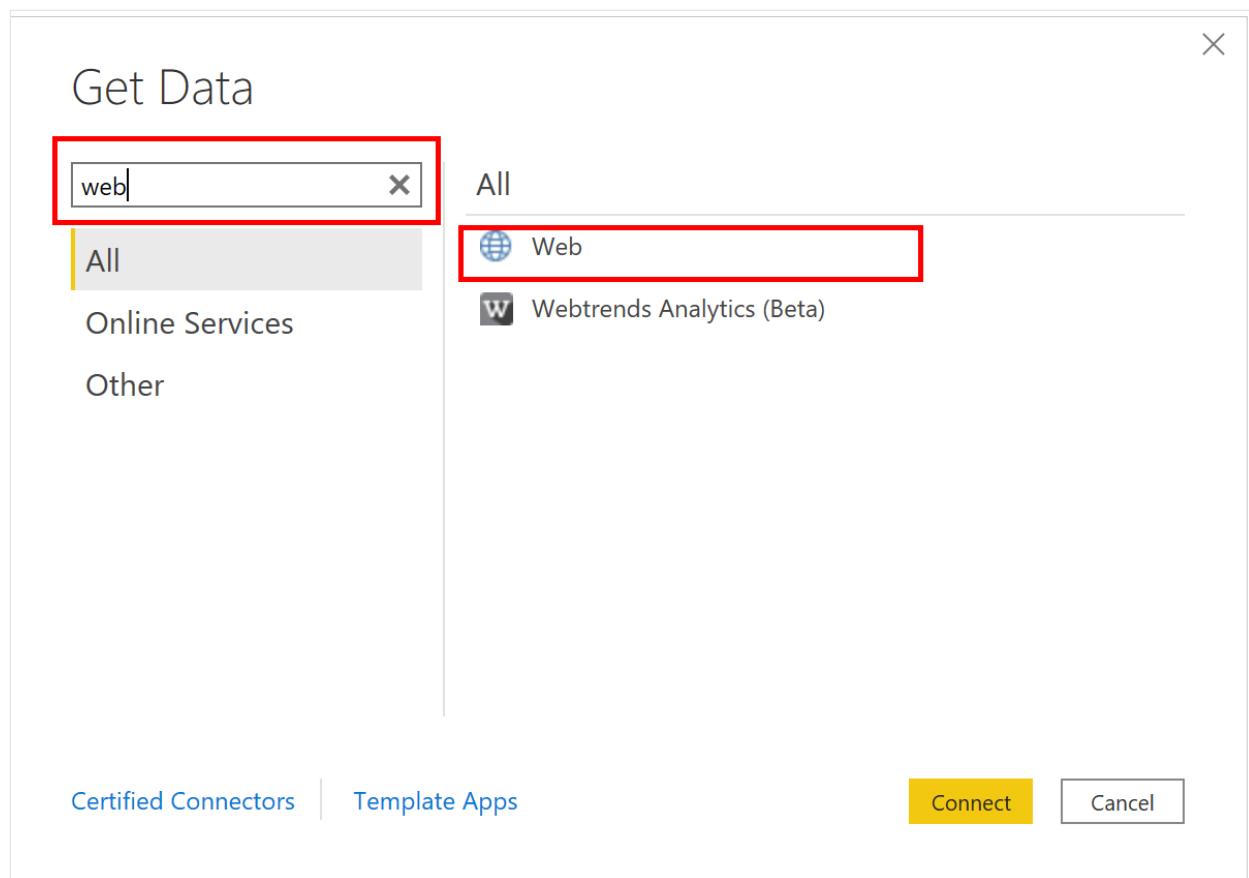
Prerequisites

Before starting this tutorial, you need to:

- Train and deploy a machine learning model in Azure Machine Learning. Use one of these three Azure Machine Learning tutorials:
 - [Option A: Code](#)
 - [Option B: Designer](#)
 - [Option C: Automated ML](#)
- Sign up for a [free Power BI trial ↗](#).
- [Install Power BI Desktop ↗](#) on a local computer.

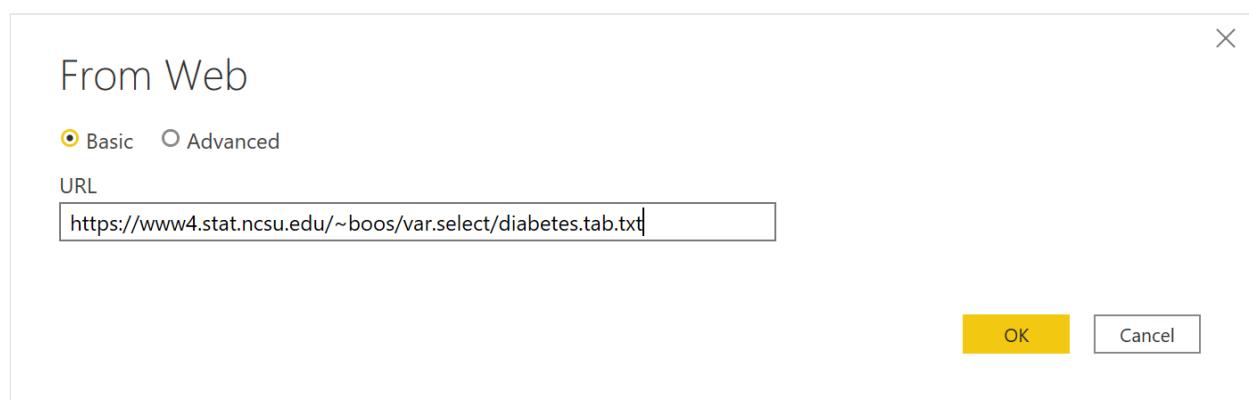
Create the data model

Open Power BI Desktop and select **Get Data**. In the **Get Data** dialog box, search for **web**. Select the **Web** source > **Connect**.



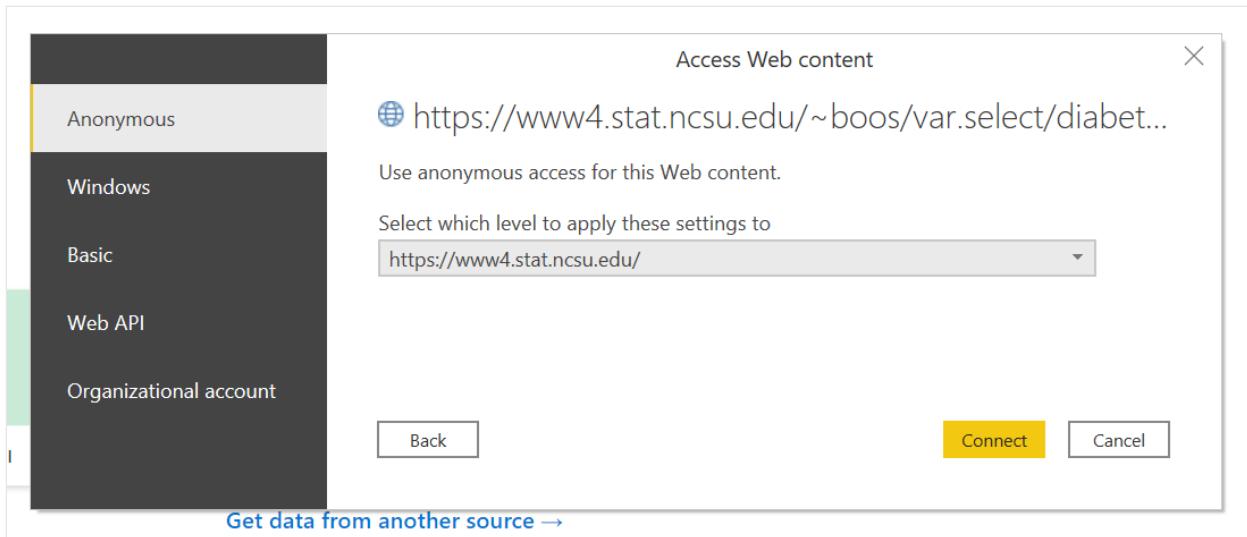
In the **From Web** dialog box, copy and paste the following URL in the box:

```
txt  
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.tab.txt
```



Select **OK**.

In Access Web content, select **Anonymous > Connect**.

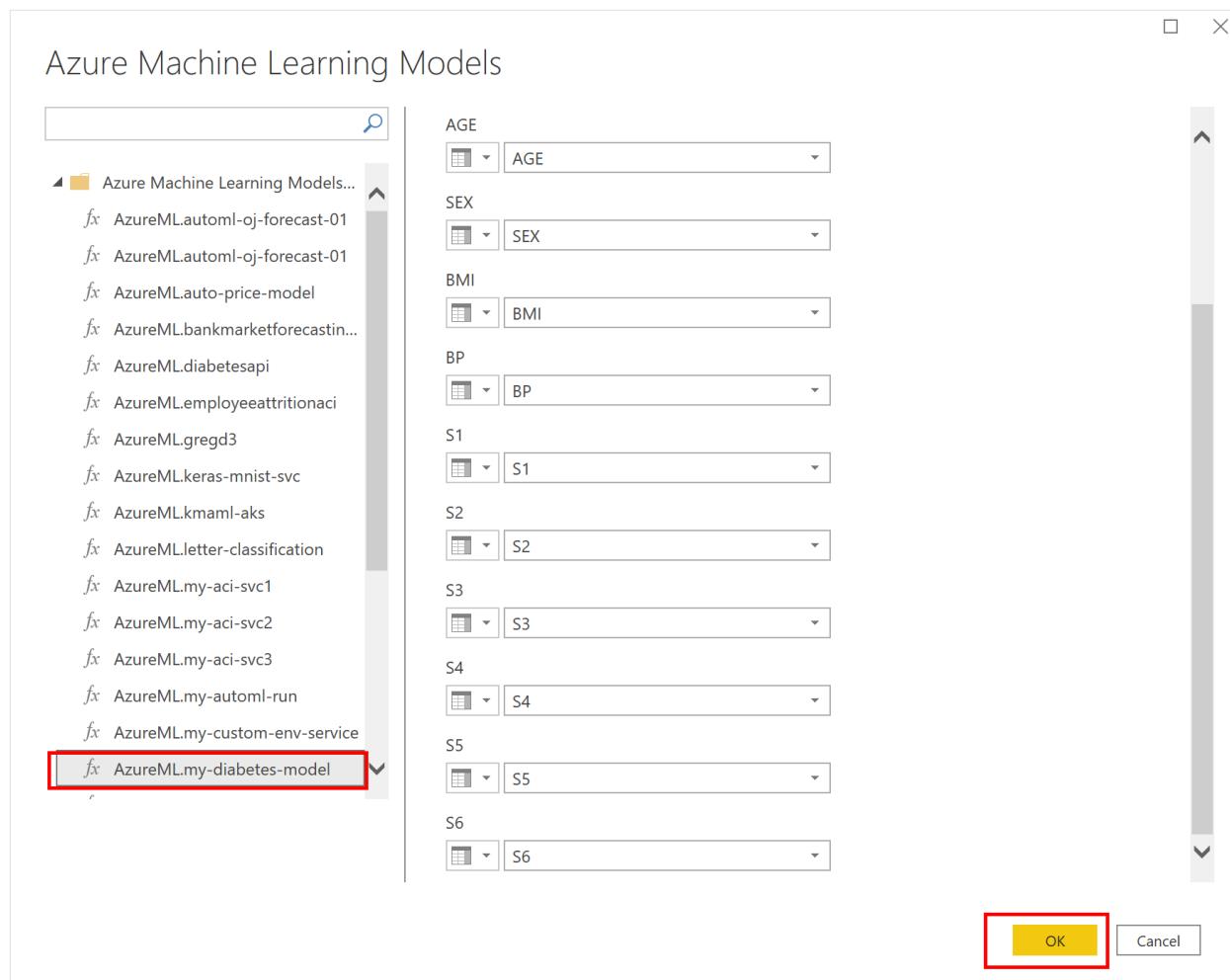


Select **Transform data** to open the **Power Query Editor** window.

In the Home ribbon of the Power Query Editor, select the **Azure Machine Learning** button.

After signing in to your Azure account using single sign-on, you see a list of available services. Select the **my-sklearn-service** that you created from the Train and deploy a machine learning model tutorial.

Power Query populates the columns automatically for you. You remember that in our schema for the service, we had a Python decorator that specified the inputs. Select **OK**.



ⓘ Note

For time series models, Power BI may not automatically detect the date format for the time column. To proceed, convert the time column to Date/Time type in Power BI before invoking Azure Machine Learning.

Selecting **OK** calls the Azure Machine Learning service. It triggers a warning on data privacy for both the data and the endpoint.



Select **Continue**. In the next screen, select **Ignore Privacy Levels checks for this file > Save**.

Once the data is scored, Power Query creates an additional column named **AzureML.my-diabetes-model**.

The screenshot shows the Power Query Editor interface with a table containing two columns: 'diabetes tab' and 'AzureML.my-sklearn-service'. The 'AzureML.my-sklearn-service' column is highlighted with a red box. The 'APPLIED STEPS' pane on the right shows the step 'Applied AzureML.my-sklearn-service'.

The data that the service returns is a **list**.

ⓘ Note

If you deployed a designer model, you see a **record**.

To get the predictions, select the double-headed arrow in the **AzureML.my-diabetes-model** column header > **Expand to New Rows**.

The screenshot shows the Power Query Editor interface with the 'AzureML.my-sklearn-service' column header highlighted with a red box. A double-headed arrow icon is located to the right of the column name.

After the expansion, you see the predictions in the **AzureML.my-diabetes-model** column.

The screenshot shows the Power Query Editor interface with the 'AzureML.my-sklearn-service' column expanded, displaying multiple rows of predictions. The entire column is highlighted with a red box. The 'APPLIED STEPS' pane on the right shows the step 'Expanded AzureML.my-sklearn-service'.

Follow these next steps to finish cleaning up your data model.

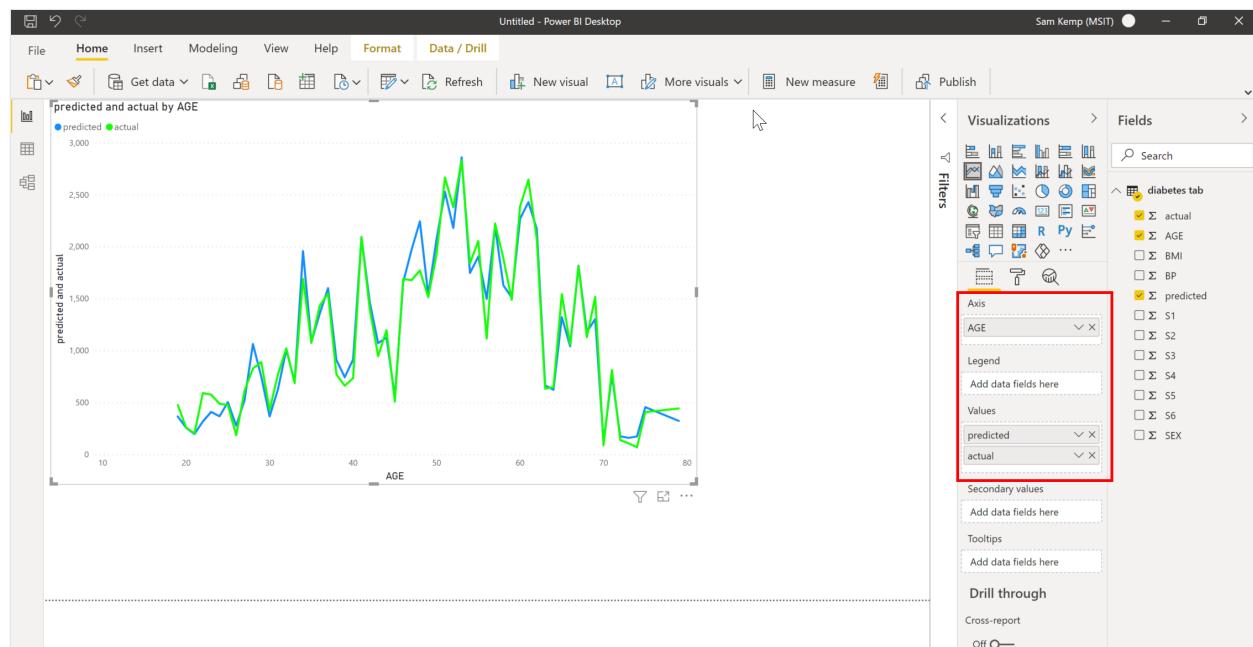
1. Rename the **AzureML.my-diabetes-model** column to **predicted**.
2. Rename the **Y** column to **actual**.
3. Change the type of the **actual** column: Select the column, then on the **Transform** ribbon select **Data Type** > **Decimal Number**.
4. Change the type of the **predicted** column: Select that column, then on the **Transform** ribbon select **Data Type** > **Decimal Number**.
5. On the **Home** ribbon, select **Close & Apply**.

Create a report with visualizations

Now you can create some visualizations to show your data.

1. In the **Visualizations** pane, select a **Line chart**.
2. With the line chart visual selected:
3. Drag the **AGE** field to the **Axis**.
4. Drag the **actual** field to **Values**.
5. Drag the **predicted** field to **Values**.

Resize the line chart to fill the page. Your report now has a single line chart with two lines, one for the predicted and one for the actual values, distributed by age.



Publish the report

You can add more visualizations if you wish. In the interest of brevity, in this tutorial we'll publish the report.

1. Save the report.
2. Select **File > Publish > Publish to Power BI**.
3. Sign in to the Power BI service.
4. Select **My Workspace**.
5. When the report is published successfully, select the **Open <MY_PBIX_FILE.pbix> in Power BI** link. The report opens the report in Power BI in your browser.

Publishing to Power BI

✓ Success!

[Open 'demo.pbix' in Power BI](#)

[Get Quick Insights](#)



Did you know?

You can create a portrait view of your report, tailored for mobile phones.
On the **View** tab, select **Mobile Layout**. [Learn more](#)

[Got it](#)

Enable datasets to refresh

In a scenario where the data source is refreshed with new data to score, you need to update your credentials so the data can be scored.

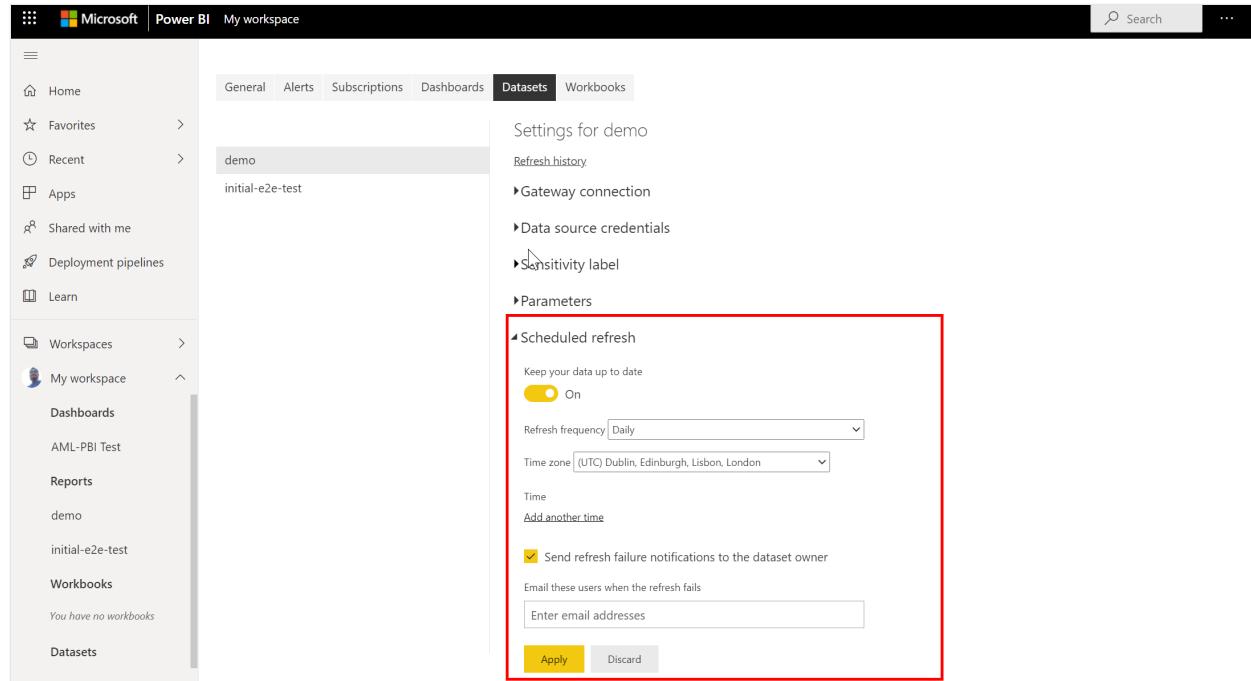
In My Workspace in the Power BI service, in the black header bar, select **More options (...)** > **Settings** > **Settings**.

The screenshot shows the Power BI service interface. The left sidebar includes 'Home', 'Favorites', 'Recent', 'Apps', 'Shared with me', and 'Deployment pipelines'. The top navigation bar has tabs for 'General', 'Alerts', 'Subscriptions', 'Dashboards', 'Datasets' (which is selected), and 'Workbooks'. On the right, there's a 'Search' bar and a 'More options' button. A context menu is open over a dataset named 'demo'. The menu items include 'Manage personal storage', 'Notifications', 'Settings' (which is highlighted with a red box), 'Download', 'Help & Support', and 'Feedback'. Below the menu, there's a note about sharing Q&A questions and a link to 'Manage embed codes'.

Select **Datasets**, expand **Data source credentials**, then select **Edit Credentials**.

The screenshot shows the 'Datasets' page for the 'demo' dataset. The 'Datasets' tab is selected in the top navigation. The dataset details show 'Settings for demo', 'Refresh history', and 'Gateway connection'. Under 'Data source credentials', there's a note: 'Your data source can't be refreshed because the credentials are invalid. Please update your credentials and try again.' Below this, there are buttons for 'Edit credentials' and 'Show in lineage view' for both 'AzureMLFunctions' and 'Web' sources. Other sections like 'Sensitivity label' and 'Parameters' are also visible.

Follow the instructions for both **azureMLFunctions** and **Web**. Make sure that you select a privacy level. You can now set a **Scheduled refresh** of the data. Select a **Refresh frequency** and **Time zone**. You can also select an email address where Power BI can send refresh failure notifications.



Select **Apply**.

ⓘ Note

When the data is refreshed, it also sends the data to your Azure Machine Learning endpoint for scoring.

Clean up resources

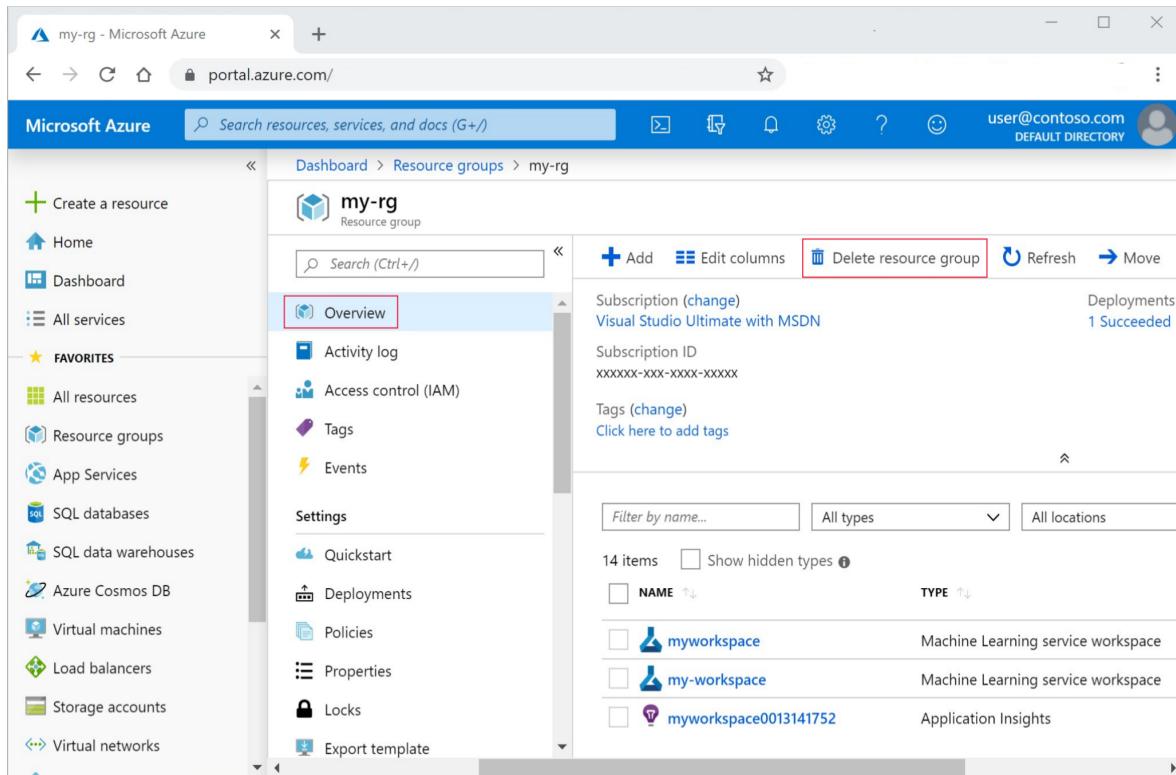
ⓘ Important

You can use the resources that you created as prerequisites for other Azure Machine Learning tutorials and how-to articles.

If you don't plan to use the resources that you created, delete them so you don't incur any charges.

1. In the Azure portal, select **Resource groups** on the far left.
2. From the list, select the resource group that you created.

3. Select Delete resource group.



The screenshot shows the Microsoft Azure portal interface. The left sidebar has a 'Resource groups' section with 'my-rg' selected. The main content area shows the 'Overview' tab for the 'my-rg' resource group. At the top right of the content area, there is a 'Delete resource group' button, which is highlighted with a red box. Below it are buttons for 'Add', 'Edit columns', 'Refresh', and 'Move'. The 'Subscription' information shows 'Visual Studio Ultimate with MSDN' and a 'Subscription ID' starting with 'xxxxxxxx-xxxx-xxxx-xxxx'. The 'Tags' section indicates '1 Succeeded' deployment. The resource list table shows 14 items, with columns for 'NAME' (sorted by type) and 'TYPE'. The items listed are:

NAME	TYPE
myworkspace	Machine Learning service workspace
my-workspace	Machine Learning service workspace
myworkspace0013141752	Application Insights

4. Enter the resource group name. Then select **Delete**.

5. In My Workspace in the Power BI service, delete the report and the related dataset. You don't need to delete Power BI Desktop or the report on your computer. Power BI Desktop is free.

Next steps

In this tutorial series, you learnt how to set up a schedule in Power BI so that new data can be scored by your scoring endpoint in Azure Machine Learning.

Refresh data from an on-premises SQL Server database

Article • 02/09/2023

In this tutorial, you explore how to refresh a Power BI dataset from a relational database that exists on premises in your local network. Specifically, this tutorial uses a sample SQL Server database, which Power BI must access through an on-premises data gateway.

In this tutorial, you complete the following steps:

- ✓ Create and publish a Power BI Desktop .pbix file that imports data from an on-premises SQL Server database.
- ✓ Configure data source and dataset settings in Power BI for SQL Server connectivity through a data gateway.
- ✓ Configure a refresh schedule to ensure your Power BI dataset has recent data.
- ✓ Do an on-demand refresh of your dataset.
- ✓ Review the refresh history to analyze the outcomes of past refresh cycles.
- ✓ Clean up resources by deleting the items you created in this tutorial.

Prerequisites

- If you don't already have one, sign up for a [free Power BI trial](#) before you begin.
- [Install Power BI Desktop](#) on a local computer.
- [Install SQL Server](#) on a local computer, and restore the [AdventureWorksDW2017 sample database from a backup](#). For more information about the AdventureWorks sample databases, see [AdventureWorks installation and configuration](#).
- [Install SQL Server Management Studio \(SSMS\)](#).
- [Install an on-premises data gateway](#) on the same local computer as SQL Server. In production, the gateway would usually be on a different computer.

Note

If you're not a gateway administrator, or don't want to install a gateway yourself, ask a gateway administrator in your organization to create the required data source definition to connect your dataset to your SQL Server database.

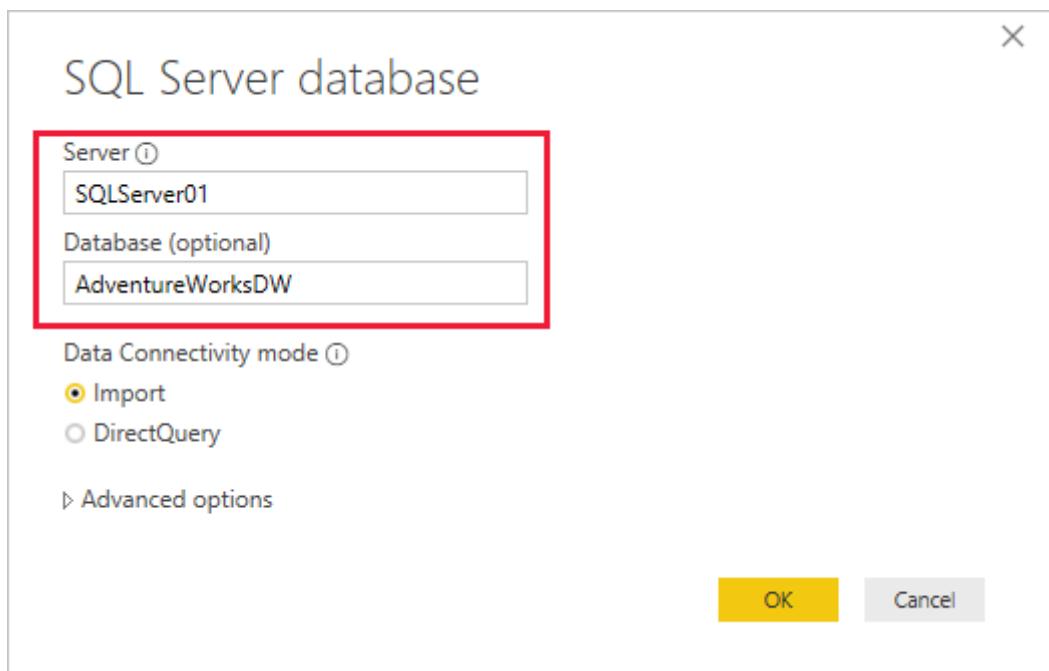
Create and publish a Power BI Desktop file

Use the following procedure to create a basic Power BI report that uses the AdventureWorksDW2017 sample database. Publish the report to the Power BI service to get a Power BI dataset, which you configure and refresh in later steps.

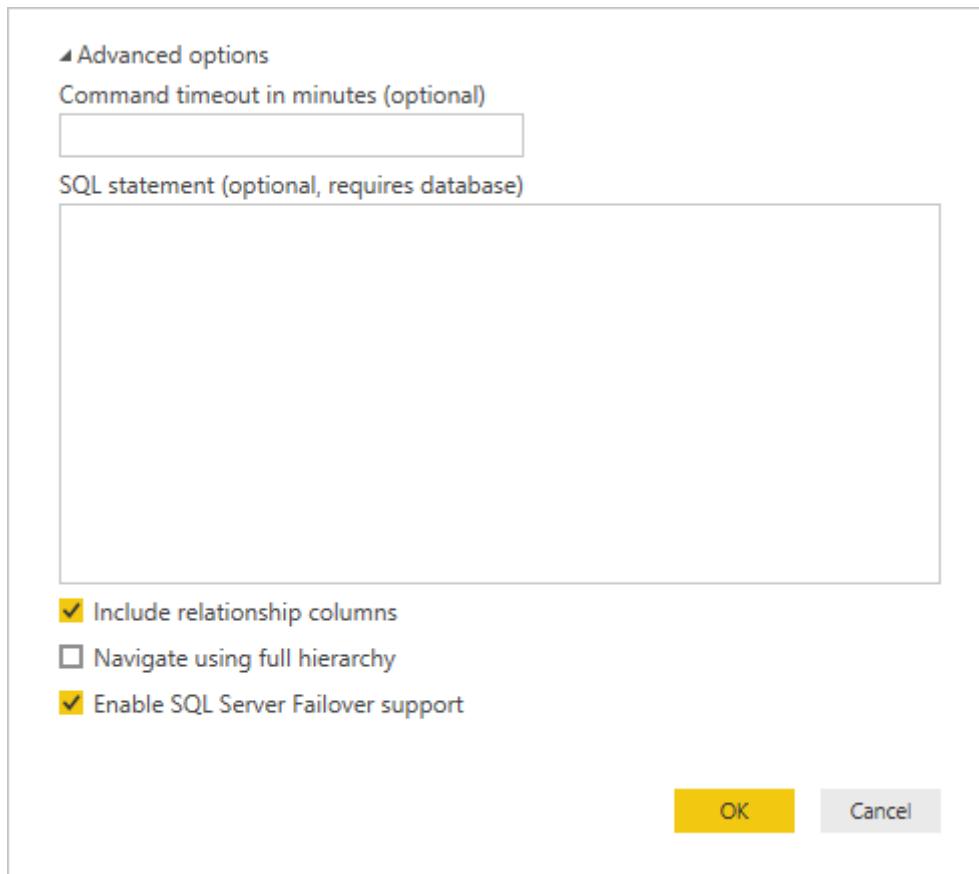
1. In Power BI Desktop, on the **Home** tab, select **Get data > SQL Server**.
2. In the **SQL Server database** dialog box, enter the **Server** and **Database (optional)** names, and make sure the **Data Connectivity mode** is set to **Import**.

! Note

If you plan to use a stored procedure, you must use **Import** as the **Data connectivity** mode.



Optionally, under **Advanced options**, you could specify a SQL statement and set other options like using [SQL Server Failover](#).



3. Select **OK**.

4. On the next screen, verify your credentials, and then select **Connect**.

① Note

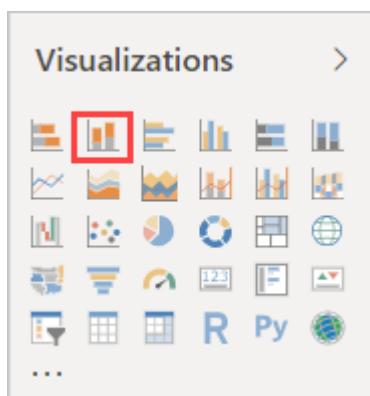
If authentication fails, make sure you selected the correct authentication method and used an account with database access. In test environments, you might use **Database** authentication with an explicit username and password. In production environments, you typically use **Windows** authentication. For more assistance, see [Troubleshoot refresh scenarios](#), or contact your database administrator.

5. If an **Encryption Support** dialog box appears, select **OK**.

6. In the **Navigator** dialog box, select the **DimProduct** table, and then select **Load**.

The screenshot shows the Power BI Desktop Navigator pane. On the left, under 'Display Options', there is a list of tables: vAssocSeqOrders, vDMPrep, vTargetMail, vTimeSeries, AdventureWorksDWBuildVersion, DatabaseLog, DimAccount, DimCurrency, DimCustomer, DimDate, DimDepartmentGroup, DimEmployee, DimGeography, DimOrganization, DimProduct, DimProductCategory, DimProductSubcategory, DimPromotion, DimReseller, and DimSalesReason. The 'DimProduct' table is selected, indicated by a checked checkbox and highlighted with a red box. On the right, a preview of the 'DimProduct' table is shown in a grid format. The columns are ProductKey, ProductAlternateKey, ProductSubcategoryKey, and Weight. The data consists of 12 rows of product keys and their corresponding values. A note at the bottom states: 'The data in the preview has been truncated due to size limits.' At the bottom right of the pane are three buttons: 'Load' (highlighted with a red box), 'Transform Data', and 'Cancel'.

7. In the Power BI Desktop Report view, in the **Visualizations** pane, select the **Stacked column chart**.



8. With the new column chart selected in the report canvas, in the **Fields** pane, select the **EnglishProductName** and **ListPrice** fields.

Fields >

Search

DimProduct

- ArabicDescription
- ChineseDescription
- Class
- Color
- Σ DaysToManufacture
- Σ DealerPrice
- EndDate
- EnglishDescription
- EnglishProductName
- FinishedGoodsFlag
- FrenchDescription
- FrenchProductName
- GermanDescription
- HebrewDescription
- JapaneseDescription
- Σ ListPrice
- ModelName

9. Drag **EndDate** from the **Fields** pane onto **Filters** on this page in the **Filters** pane, and under **Basic filtering**, select the checkbox for **(Blank)**.

EndDate
is (Blank)

Filter type ⓘ

Basic filtering

Search

Select all

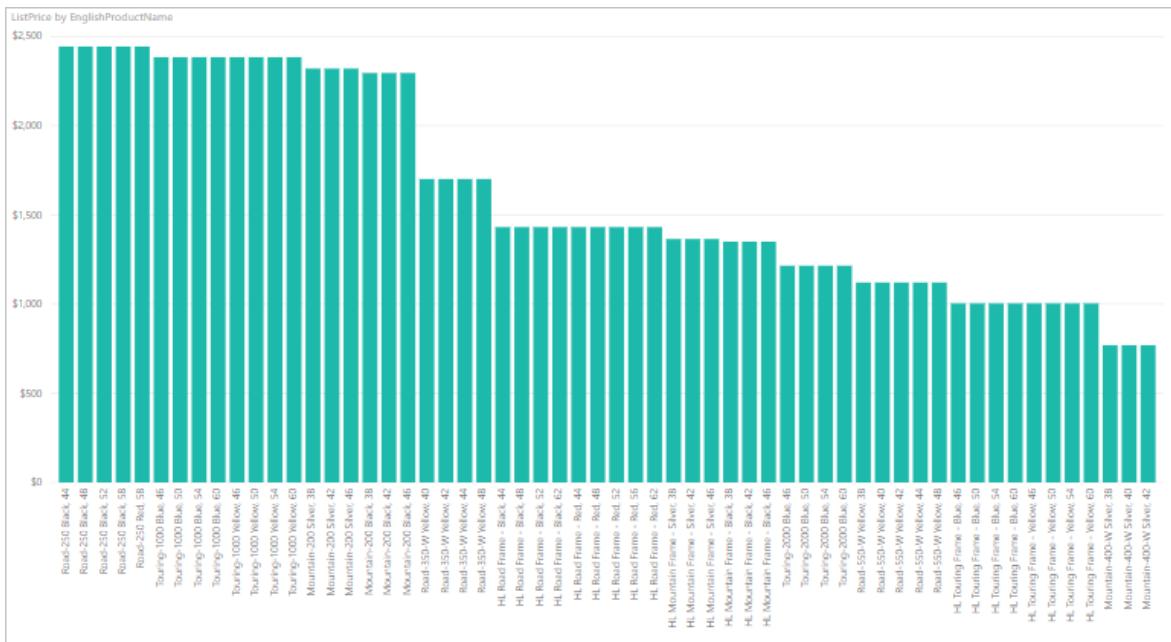
(Blank)

12/28/2007 12:00:00 AM

12/28/2008 12:00:00 AM

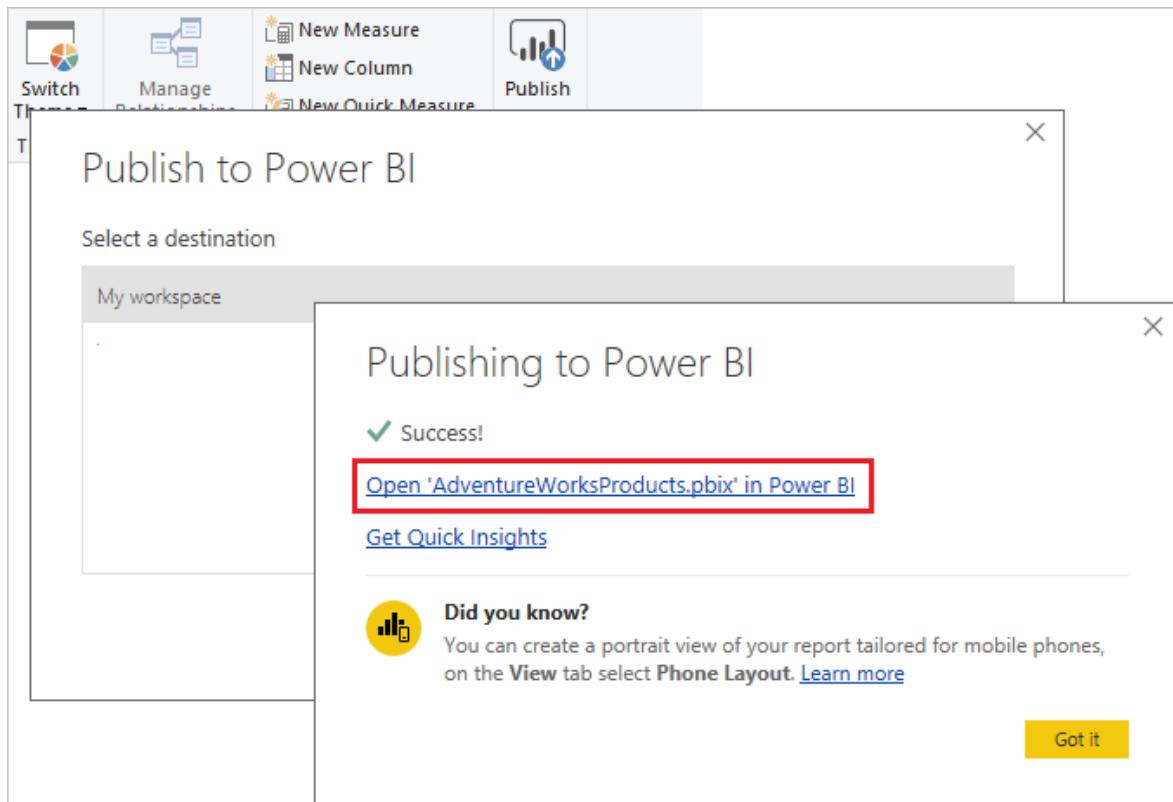
Require single selection

The visualization should now look similar to the following chart:



Notice that the **Road-250 Red** product has the same list price as the other **Road-250** products. This price will change when you later update the data and refresh the report.

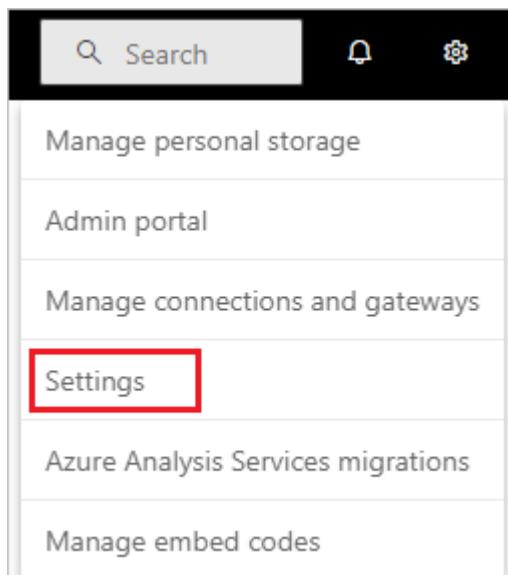
10. Save the report with the name *AdventureWorksProducts.pbix*.
 11. On the **Home** tab, select **Publish**.
 12. On the **Publish to Power BI** screen, choose **My Workspace**, and then select **Select**.
Sign in to the Power BI service if necessary.
 13. When the **Success** message appears, select **Open 'AdventureWorksProducts.pbix'** in Power BI.



Connect the dataset to the SQL Server database

In Power BI Desktop, you connected directly to your on-premises SQL Server database. In the Power BI service, you need a data gateway to act as a bridge between the cloud and your on-premises network. Follow these steps to add your on-premises SQL Server database as a data source to a gateway and connect your dataset to this data source.

1. In the Power BI service, in the upper-right corner of the screen, select the settings gear icon and then select **Settings**.



- Select the **Datasets** tab, and then select the **AdventureWorksProducts** dataset from the list of datasets.
- Expand **Gateway connection** and verify that at least one gateway is listed. If you don't see a gateway, make sure you followed the instructions to [install an on-premises data gateway](#).

General Dashboards Datasets Workbooks Alerts Subscriptions

Settings for AdventureWorksProducts

[Refresh history](#)

▲ Gateway connection
To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click apply.

Use an On-premises or VNet data gateway
 On

Gateway	Department	Contact information	Status	Actions
<input checked="" type="radio"/> Enterprise Data Gateway		Kay@KayTestOrg.onmi...	✗ Not configured correctly	

⚠ You have no personal gateways installed. [Learn more](#)

[Install now](#)

Apply Discard

- Select the arrow toggle under **Actions** to expand the data sources, and then select the **Add to gateway** link next to your data source.

▲ Gateway connection
To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click apply.

Use an On-premises or VNet data gateway
 On

Gateway	Department	Contact information	Status	Actions
<input checked="" type="radio"/> Enterprise Data Gateway		Kay@KayTestOrg.onmi...	✗ Not configured correctly	

Data sources included in this dataset:

✗ SqlServer{"server":"sqlserver01","database":"adventureworksdw2017"}	Add to gateway
---	--------------------------------

⚠ You have no personal gateways installed. [Learn more](#)

[Install now](#)

Apply Discard

- On the **New connection** screen with **On-premises** selected, complete or verify the following fields. Most fields are already filled in.
 - Gateway cluster name:** Verify or enter the gateway cluster name.
 - Connection name:** Enter a name for the new connection, such as **AdventureWorksProducts**.
 - Connection type:** Select **SQL Server** if not already selected.

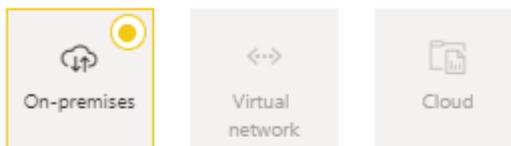
- **Server:** Verify or enter your SQL Server instance name. Must be identical to what you specified in Power BI Desktop.
- **Database:** Verify or enter your SQL Server database name, such as **AdventureWorksDW2017**. Must be identical to what you specified in Power BI Desktop.

Under **Authentication**:

- **Authentication method:** Select **Windows**, **Basic**, or **OAuth2**, usually **Windows**.
- **Username and Password:** Enter the credentials you use to connect to SQL Server.

New connection

X



Gateway cluster name *

powerbi-docs

Connection name *

AdventureWorksProducts

Connection type *

SQL Server

Server *

LAPTOP-R5

Database *

AdventureWorksDW2017

Authentication ⓘ

Authentication method *

Windows

Username

<username>

Password

Skip test connection

Single sign-on

- Use SSO via Kerberos for DirectQuery queries ⓘ
- Use SSO via Kerberos for DirectQuery and Import queries ⓘ
- Use SSO via Azure AD for DirectQuery queries ⓘ

Create

Close

6. Select **Create**.

7. Back on the **Settings** screen, expand the **Gateway connection** section, and verify that the data gateway you configured now shows a **Status** of running on the machine where you installed it. Select **Apply**.

The screenshot shows the 'Datasets' tab selected in the navigation bar. Under 'AdventureWorksProducts', the 'Gateway connection' section is visible. It lists an 'Enterprise Data Gateway' with the status 'Running on KAYU-GTW'. A red box highlights the 'Apply' button at the bottom left. A message box in the top right corner indicates that updates to the gateway connection have been applied.

Configure a refresh schedule

Now that you've connected your Power BI dataset to your SQL Server on-premises database through a data gateway, follow these steps to configure a refresh schedule. Refreshing your dataset on a scheduled basis helps ensure that your reports and dashboards have the most recent data.

1. In the left navigation pane, expand **My Workspace**.
2. In the **Datasets** section, point to the **AdventureWorksProducts** dataset, select the **Open** menu three vertical dots icon, and then select **Schedule refresh**.

Tip

Make sure you point to the **AdventureWorksProducts** dataset, not the report with the same name, which doesn't have a **Schedule refresh** option.

3. In the **Scheduled refresh** section, under **Keep your data up to date**, set refresh to **On**.
4. Under **Refresh frequency**, select **Daily** for this example, and then under **Time**, select **Add another time**.

For this example, specify **6:00 AM**, then select **Add another time** and specify **6:00 PM**.

Screenshot of the Power BI dataset settings page for "AdventureWorksProducts". The "Datasets" tab is selected. On the left, a sidebar lists datasets: "2020SU09 Blog Demo - September", "AdventureWorksProducts" (selected), "Best states for sunglass sales", "BuildDetails - 2021-03-17T160909.414", "Conditional-formatting-Best states for sunglass s", "Customer Profitability Sample", "data", "DirectQuery", "fb-comments", "Financial Sample (1)", "Opportunity Analysis Sample", "Procurement Analysis Sample", "Procurement Analysis Sample1", "QueryResult", "Retail Analysis Sample", "Retail Analysis Sample PBIX (1)", "Sales & Returns Sample v201912", "Sales and Marketing Sample PBIX", and "statecharacteristics".

The main pane shows "Settings for AdventureWorksProducts". It includes a "View dataset" link, a "Refresh history" button, and sections for "Dataset description", "Gateway connection", "Data source credentials", "Sensitivity label", and "Parameters". A red box highlights the "Scheduled refresh" section.

Scheduled refresh
Keep your data up to date
Configure a data refresh schedule to import data from the data source into the dataset. [Learn more](#)

On

Refresh frequency: Daily

Time zone: (UTC-08:00) Pacific Time (US and Canada)

Time: 6:00 AM and 6:00 PM

Add another time

Send refresh failure notifications to:

Dataset owner

These contacts:

Apply **Discard**

(!) Note

You can configure up to eight daily time slots if your dataset is on shared capacity, or 48 time slots on Power BI Premium.

5. Leave the checkbox under **Send refresh failure notifications to** set to **Dataset owner**, and select **Apply**.

Now that you've configured a refresh schedule, Power BI refreshes your dataset at the next scheduled time, within a margin of 15 minutes.

Do an on-demand refresh

To refresh the data anytime, such as to test your gateway and data source configuration, you can do an on-demand refresh by using the **Refresh Now** option in the left pane **Dataset** menu. On-demand refreshes don't affect the next scheduled refresh time.

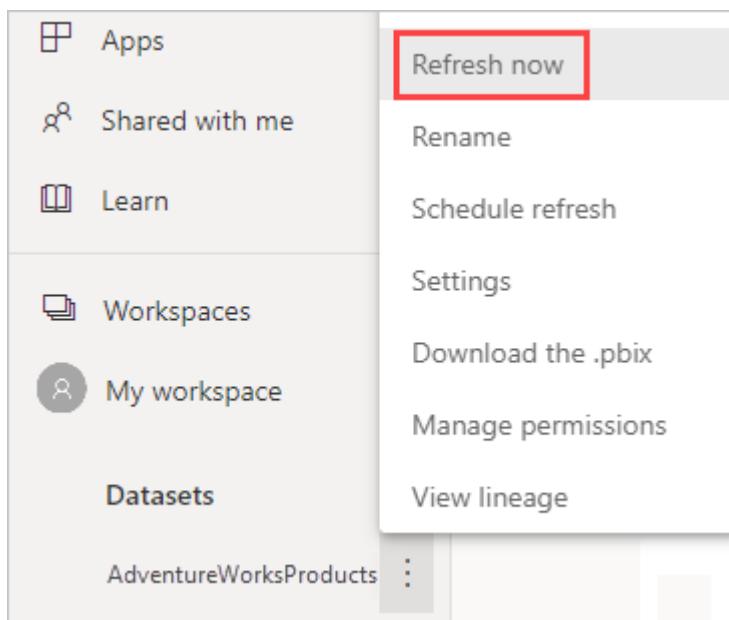
To illustrate an on-demand refresh, first change the sample data by using SSMS to update the `DimProduct` table in the AdventureWorksDW2017 database, as follows:

```
SQL

UPDATE [AdventureWorksDW2017].[dbo].[DimProduct]
SET ListPrice = 5000
WHERE EnglishProductName = 'Road-250 Red, 58'
```

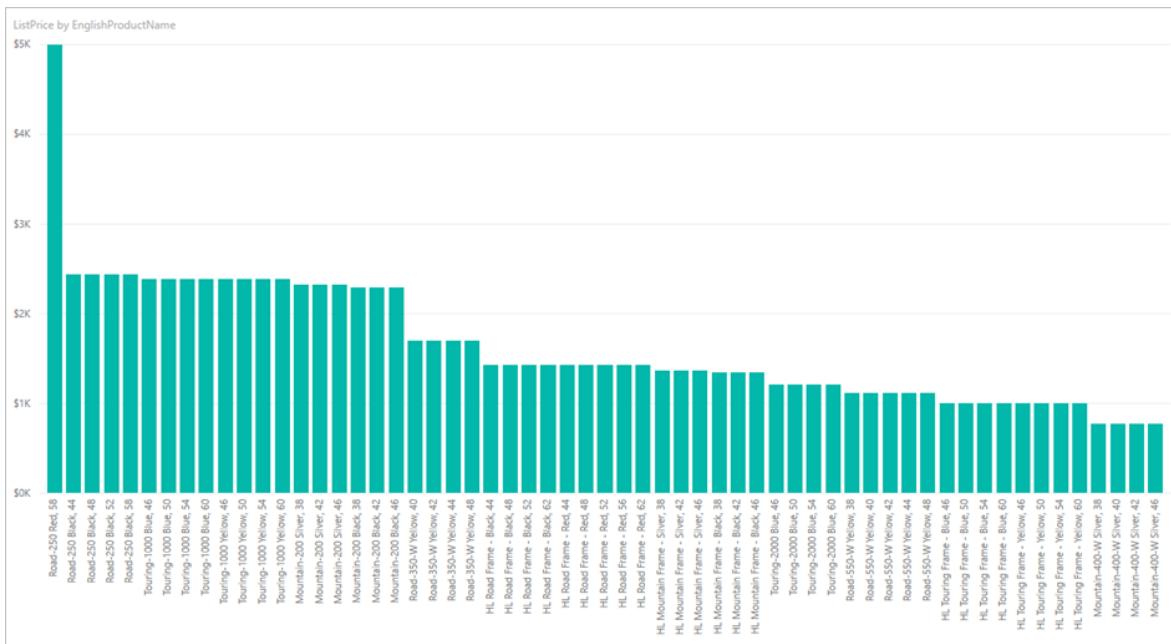
Follow these steps to make the updated data flow through the gateway connection to the dataset and into the Power BI reports:

1. In the Power BI service, expand **My Workspace** in the left navigation pane.
2. In the **Datasets** section, hover over the **AdventureWorksProducts** dataset, select the three vertical dots **Open menu** icon, and then select **Refresh now**.



A **Preparing for refresh** message appears at upper right.

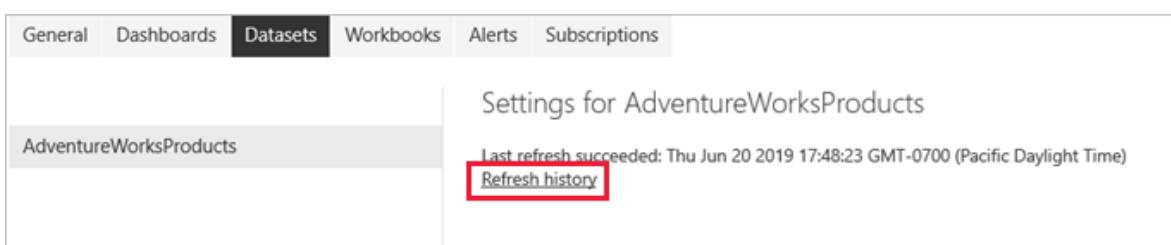
3. In the **Reports** section of **My Workspace**, select **AdventureWorksProducts**. See how the updated data flowed through into the report, and the product with the highest list price is now **Road-250 Red, 58**.



Review the refresh history

It's a good idea to periodically use the refresh history to check the outcomes of past refresh cycles. Database credentials might have expired, or the selected gateway might have been offline when a scheduled refresh was due. Follow these steps to examine the refresh history and check for issues.

1. In the upper-right corner of the Power BI screen, select the settings gear icon and then select **Settings**.
2. On the **Datasets** tab, select the dataset you want to examine, such as **AdventureWorksProducts**.
3. Select the **Refresh history** link.



4. On the **Scheduled** tab of the **Refresh history** dialog box, notice the past scheduled and on-demand refreshes with their **Start** and **End** times. A **Status of Completed** indicates that Power BI did the refreshes successfully. For failed refreshes, you can see the error message and examine error details.

Details	Type	Start	End	Status	Message
	On demand	6/20/2019, 5:48:16 PM	6/20/2019, 5:48:23 PM	Completed	
	On demand	6/20/2019, 5:41:22 PM	6/20/2019, 5:41:47 PM	Completed	

Close

➊ Note

The OneDrive tab is relevant only for datasets that are connected to Power BI Desktop files, Excel workbooks, or CSV files on OneDrive or SharePoint Online. For more information, see [Data refresh in Power BI](#).

Clean up resources

Follow these instructions to clean up the resources you created for this tutorial:

- If you don't want to use the sample data anymore, use SSMS to drop the database.
- If you don't want to use the SQL Server data source, remove the data source from your data gateway. Also consider uninstalling the data gateway, if you installed it only for this tutorial.
- Also delete the AdventureWorksProducts dataset and report that Power BI created when you published the *AdventureWorksProducts.pbix* file.

Next steps

This tutorial explored how to:

- Import data from an on-premises SQL Server database into a Power BI dataset.
- Refresh the Power BI dataset on a scheduled and on-demand basis to update the reports and dashboards that use the dataset.

Now, you can learn more about Power BI data refresh and managing data gateways and data sources.

- [Manage an on-premises data gateway](#)
- [Manage your data source - Import/scheduled refresh](#)

- Data refresh in Power BI

Tutorial: Automate configuration of template app installation using an Azure function

Article • 11/29/2021

Template apps are a great way for customers to start getting insights from their data. Template apps get them up and running quickly by connecting them to their data. The template apps provide customers with prebuilt reports that they can customize if they so desire.

Customers aren't always familiar with the details of how to connect to their data. Having to provide these details when they install a template app can be a pain point for them.

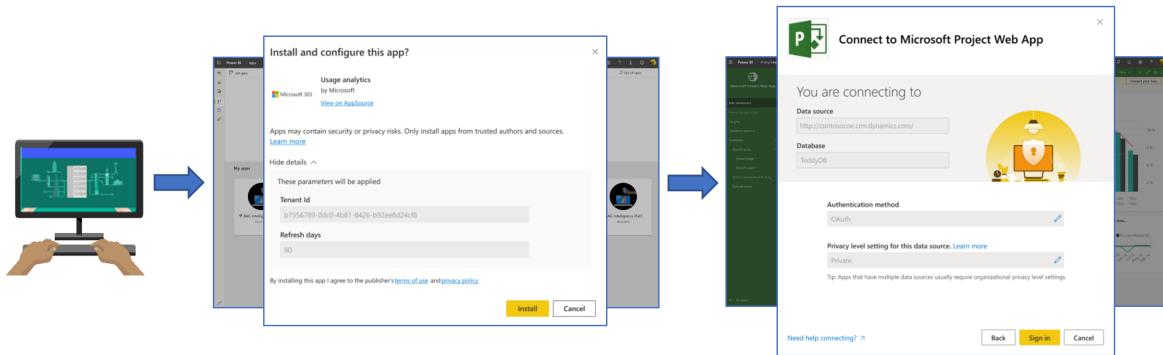
If you are a data services provider and have created a template app to help your customers get started with their data on your service, you can make it easier for them to install your template app. You can automate the configuration of your template app's parameters.

When the customer signs in to your portal, they select a special link you've prepared. This link:

- Launches the automation, which gathers the information it needs.
- Preconfigures the template app parameters.
- Redirects the customer to their Power BI account where they can install the app.

All they have to do is select **Install** and authenticate against their data source, and they're good to go!

The customer experience is illustrated here.



Customer logs into your portal and clicks the link to install your preconfigured template app.

This launches your automation which then gathers the info it needs and preconfigures the template app parameters.

The automation redirects the customer to their Power BI account where they can see the preconfigured parameters and install the template app.

Customer authenticates on the data source and starts getting insights from their own data in the prebuilt reports your template app offers

In this tutorial, you'll use an automated installation Azure Functions sample that we've created to preconfigure and install your template app. This sample has deliberately been kept simple for demonstration purposes. It encapsulates the setup of an Azure function to use Power BI APIs for installing a template app and configuring it for your users automatically.

For more information about the general automation flow and the APIs that the app uses, see [Automate configuration of a template app installation](#).

Our simple application uses an Azure function. For more information about Azure Functions, see the [Azure Functions documentation](#).

Basic flow

The following basic flow lists what the application does when the customer launches it by selecting the link in your portal.

1. The user signs in to the ISV's portal and selects the supplied link. This action initiates the flow. The ISV's portal prepares the user-specific configuration at this stage.
2. The ISV acquires an *app-only* token based on a [service principal \(app-only token\)](#) that's registered in the ISV's tenant.
3. Using [Power BI REST APIs](#), the ISV creates an *install ticket*, which contains the user-specific parameter configuration as prepared by the ISV.
4. The ISV redirects the user to Power BI by using a `POST` redirection method, which contains the install ticket.

5. The user is redirected to their Power BI account with the install ticket and is prompted to install the template app. When the user selects **Install**, the template app is installed for them.

 **Note**

While parameter values are configured by the ISV in the process of creating the install ticket, data source-related credentials are only supplied by the user in the final stages of the installation. This arrangement prevents them from being exposed to a third party and ensures a secure connection between the user and the template app data sources.

Prerequisites

- Your own Azure Active Directory (Azure AD) tenant set up. For instructions on how to set one up, see [Create an Azure AD tenant](#).
- A [service principal \(app-only token\)](#) registered in the preceding tenant.
- A parameterized [template app](#) that's ready for installation. The template app must be created in the same tenant in which you register your application in Azure AD. For more information, see [Template app tips](#) or [Create a template app in Power BI](#).
- To be able to test your automation work flow, add the service principal to the template app workspace as an Admin.
- A Power BI Pro license. If you're not signed up for Power BI Pro, [sign up for a free trial](#) before you begin.

Set up your template apps automation development environment

Before you continue setting up your application, follow the instructions in [Quickstart: Create an Azure Functions app with Azure App Configuration](#) to develop an Azure function along with an Azure app configuration. Create your app configuration as described in the article.

Register an application in Azure AD

Create a service principal as described in [Embed Power BI content with service principal and an application secret](#).

Make sure to register the application as a **server-side web application** app. You register a server-side web application to create an application secret.

Save the *application ID* (ClientID) and *application secret* (ClientSecret) for later steps.

You can go through the [Embedding setup tool](#) to quickly get started creating an app registration. If you're using the [Power BI App Registration Tool](#), select the **Embed for your customers** option.

Add the service principal to the template app workspace as an Admin, so that you will be able to test your automation work flow.

Template app preparation

After you've created your template app and it's ready for installation, save the following information for the next steps:

- *App ID, Package Key, and Owner ID* as they appear in the installation URL at the end of the [Define the properties of the template app](#) process when the app was created.

You can also get the same link by selecting **Get link** in the template app's [Release Management pane](#).

- *Parameter names* as they're defined in the template app's dataset. Parameter names are case-sensitive strings. They can also be retrieved from the **Parameter Settings** tab when you [define the properties of the template app](#) or from the dataset settings in Power BI.

Note

You can test your preconfigured installation application on your template app if the template app is ready for installation, even if it isn't publicly available on AppSource yet. For users outside your tenant to be able to use the automated installation application to install your template app, the template app must be publicly available in the [Power BI apps marketplace](#). Before you distribute your template app by using the automated installation application you're creating, be sure to publish it to [Partner Center](#).

Install and configure your template app

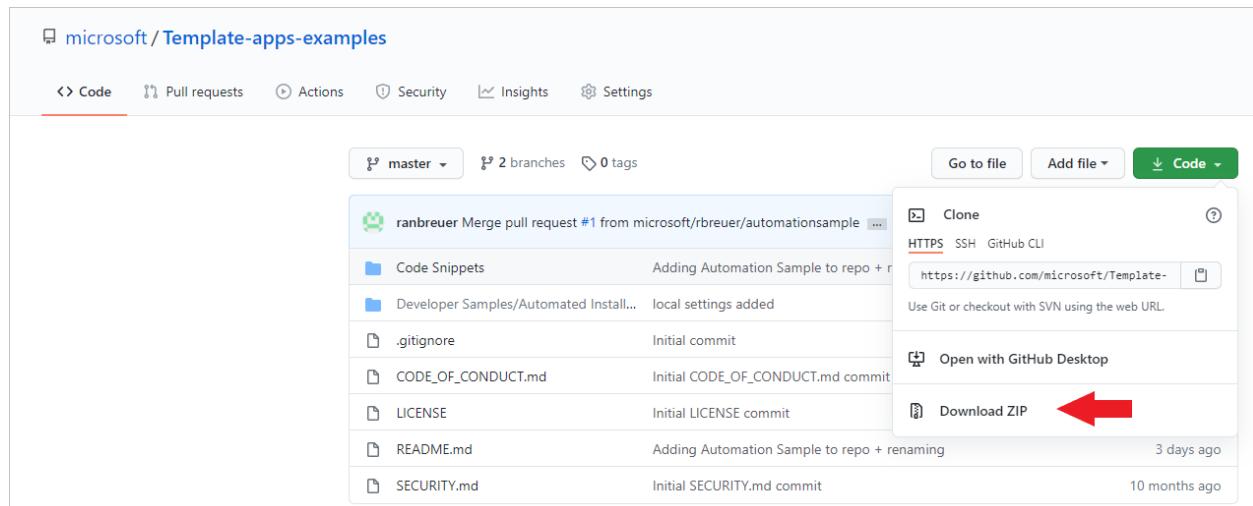
In this section, you'll use an automated installation Azure Functions sample that we created to preconfigure and install your template app. This sample has deliberately been kept simple for demonstration purposes. It allows you to use an [Azure function](#) and [Azure App Configuration](#) to easily deploy and use the automated installation API for your template apps.

Download [Visual Studio](#) (version 2017 or later)

Download [Visual Studio](#) (version 2017 or later). Make sure to download the latest NuGet package [.](#).

Download the automated installation Azure Functions sample

Download the [automated installation Azure Functions sample](#) from GitHub to get started.



Set up your Azure app configuration

To run this sample, you need to set up your Azure app configuration with the values and keys as described here. The keys are the **application ID**, the **application secret**, and your template app's **AppId**, **PackageKey**, and **OwnerId** values. See the following sections for information about how to get these values.

The keys are also defined in the **Constants.cs** file.

Configuration key	Meaning
TemplateAppInstall:Application:AppId	AppId from the installation URL
TemplateAppInstall:Application:PackageKey	PackageKey from the installation URL

Configuration key	Meaning
TemplateAppInstall:Application:OwnerId	OwnerId from the installation URL
TemplateAppInstall:ServicePrincipal:ClientId	Service principal application ID
TemplateAppInstall:ServicePrincipal:ClientSecret	Service principal application secret

The **Constants.cs** file is shown here.

```

C# Constants.cs
Developer Samples > Automated Install Azure Function > InstallTemplateAppSample > InstallTemplateApp > C# Constants.cs > ...
You, 3 days ago | 1 author (You)
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
You, 3 days ago | 1 author (You)
5 namespace InstallTemplateApp
6 {
    10 references | You, 3 days ago | 1 author (You)
7     public static class Constants
8     {
        1 reference
9         public const string RedirectUrlFormat = "{0}/Redirect?action=InstallApp&appId={1}&packageKey={2}&ownerId={3}";
10
11         // Configurable Parameters
12         1 reference
13         public const string AppIdConfigurationKey = "TemplateAppInstall:Application:AppId";
14         1 reference
15         public const string PackageKeyConfigurationKey = "TemplateAppInstall:Application:PackageKey";
16         1 reference
17         public const string OwnerIdConfigurationKey = "TemplateAppInstall:Application:OwnerId";
18         1 reference
19         public const string ClientIdConfigurationKey = "TemplateAppInstall:ServicePrincipal:ClientId";
20         1 reference
21         public const string ClientSecretConfigurationKey = "TemplateAppInstall:ServicePrincipal:ClientSecret";
22
23         // Power BI API Constants
24         1 reference
25         public const string PowerBIApiResourceUrl = "https://analysis.windows.net/powerbi/api";
26         1 reference
27         public const string PowerBIApiUrl = "https://api.powerbi.com";
28         1 reference
29         public const string PowerBIApiBaseUrl = "https://app.powerbi.com";
30         1 reference
31         public const string PowerBIApiAuthorityUrlFormat = "https://login.microsoftonline.com/{0}/";
32     }
}

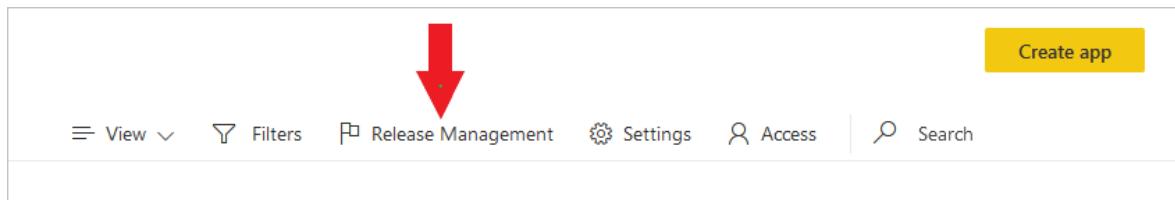
```

Get the template app properties

Fill in all relevant template app properties as they're defined when the app is created. These properties are the template app's **AppId**, **PackageKey**, and **OwnerId** values.

To get the preceding values, follow these steps:

1. Sign in to [Power BI](#).
2. Go to the application's original workspace.
3. Open the **Release Management** pane.



4. Select the app version, and get its installation link.

A screenshot of the "Release Management" screen. The left side shows a table with columns for Refresh, Endorsement, and Sensitivity, containing several rows of data. The right side displays the app lifecycle stages: Workspace, Testing, Pre-production, and Production. The "Production" stage is highlighted with a yellow circle. An "app" icon is next to the "Shark Attacks" entry. Below the entry, it says "Created on: 2/11/2019, 2:31 PM" and "Promoted on: 10/18/2020, 3:50 PM". A note states: "This is the production version. Make sure you always promote the latest version that's live on AppSource to production." A red arrow points from the bottom left towards the "Get link" button. The "Get link" button is located at the bottom right of the production entry. Other buttons include "Close" and three dots (...).

5. Copy the link to the clipboard.

A screenshot of a modal dialog box titled "Installation link". It contains the text "Shark Attacks Version 50" and a URL: "https://app.powerbi.com/Redirect?action=InstallApp&appId=3c386c19-26da-4e3c-b6aa-8cd16bf71c67&p". A "Copy" button is to the right of the URL. A red arrow points from the bottom left towards the "Copy" button. At the bottom right of the dialog is a "Close" button.

6. This installation URL holds the three URL parameters whose values you need. Use the **appId**, **packageKey**, and **ownerId** values for the application. A sample URL will be similar to what is shown here.

HTML

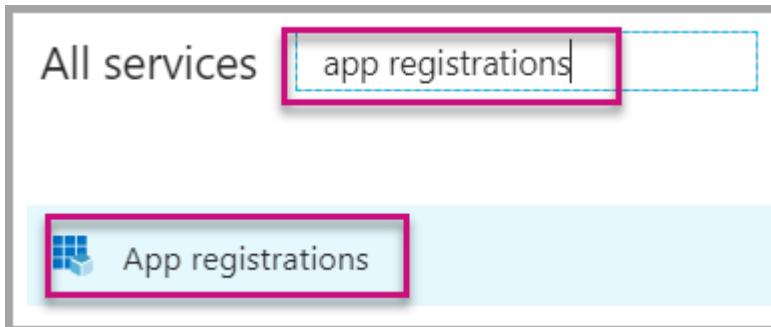
```
https://app.powerbi.com/Redirect?  
action=InstallApp&appId=3c386...16bf71c67&packageKey=b2df4b...dLpHIUnum  
2pr6k&ownerId=72f9...1db47&buildVersion=5
```

Get the application ID

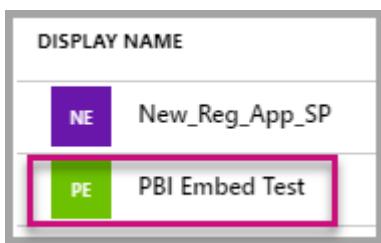
Fill in the **applicationId** information with the application ID from Azure. The **applicationId** value is used by the application to identify itself to the users from which you're requesting permissions.

To get the application ID, follow these steps:

1. Sign in to the [Azure portal](#).
2. In the left pane, select All services > App registrations.



3. Select the application that needs the **application ID**.



4. There's an application ID that's listed as a GUID. Use this application ID as the **applicationId** value for the application.

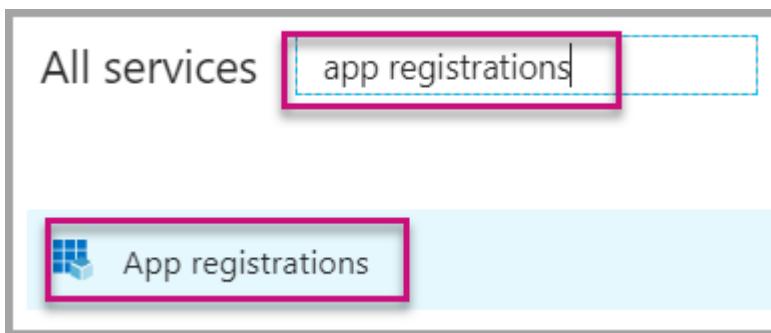
Display name	: PBI Embed Test
Application (client) ID	: 93105ba0-2cee-4b35-80ab-f1e80abdbb05
Directory (tenant) ID	: fd6d462f-fa94-4505-93bb-b1881d76a1c9
Object ID	: 5d5cfaba-574b-406a-8613-d4c9b008b234

Get the application secret

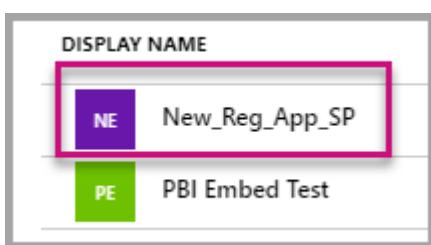
Fill in the **ApplicationSecret** information from the **Keys** section of your **App registrations** section in Azure. This attribute works when you use the [service principal](#).

To get the application secret, follow these steps:

1. Sign in to the [Azure portal](#).
2. In the left pane, select **All services > App registrations**.



3. Select the application that needs to use the **application secret**.



4. Select **Certificates and secrets** under **Manage**.
5. Select **New client secrets**.
6. Enter a name in the **Description** box, and select a duration. Then select **Save** to get the value for your application. When you close the **Keys** pane after you save the key value, the **Value** field shows only as hidden. At that point, you aren't able to retrieve the key value. If you lose the key value, create a new one in the Azure portal.

Client secrets		
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.		
+ New client secret		
DESCRIPTION	EXPIRES	VALUE
PBI Test	12/31/2299	xKWBjL5A_Lug-2A1=Vb.2 

Test your function locally

Follow the steps as described in [Run the function locally](#) to run your function.

Configure your portal to issue a `POST` request to the URL of the function. An example is `POST http://localhost:7071/api/install`. The request body should be a JSON object that describes key-value pairs. Keys are *parameter names* as defined in Power BI Desktop. Values are the desired values to be set for each parameter in the template app.

Note

In production, parameter values are deduced for each user by your portal's intended logic.

The desired flow should be:

1. The portal prepares the request, per user or session.
2. The `POST /api/install` request is issued to your Azure function. The request body consists of key-value pairs. The key is the parameter name. The value is the desired value to be set.
3. If everything is configured properly, the browser should automatically redirect to the customer's Power BI account and show the automated installation flow.
4. Upon installation, parameter values are set as configured in steps 1 and 2.

Next steps

Publish your project to Azure

To publish your project to Azure, follow the instructions in the [Azure Functions documentation](#). Then you can integrate template app automated installation APIs into your product and begin testing it in production environments.

Power BI data sources

Article • 06/09/2023

Power BI uses Power Query to connect to data sources. **Power BI data sources** are documented in the following article: [Power Query \(including Power BI\) connectors](#).

Each data source article in the Power Query documentation describes the capabilities of the data connector, such as whether DirectQuery is supported. The following image shows the **Capabilities supported** section for [Azure Data Explorer \(Kusto\)](#), where it states that DirectQuery is supported for the connector in Power BI. If DirectQuery (or any other capability) isn't listed, the capability isn't supported.

Capabilities supported

- Import
- DirectQuery (Power BI Datasets)
- Advanced options
 - Limit query result record number
 - Limit query result data size in Bytes
 - Disable result-set truncation
 - Additional set statements

Connect to Azure Data Explorer from Power Query Desktop

For a list of the connectors available in Power Query, see [connectors in Power Query](#).

For information about dataflows in Power BI, see [connect to data sources for Power BI dataflows](#).

Considerations and limitations

- Many data connectors for Power BI Desktop require Internet Explorer 10 (or newer) for authentication.
- Some data sources are available in Power BI Desktop optimized for Power BI Report Server, but aren't supported when published to Power BI Report Server. See [Power BI report data sources in Power BI Report Server](#) for the list of supported data sources.
- Power BI Desktop and the Power BI service may send multiple queries for any given query, to get schema information or the data itself, based in part on whether

data is cached. This behavior is by design, for more information, see the Power Query article that describes [why a query may run multiple times](#).

Next steps

The following articles provide more information about Power BI and connecting to data:

- [Connectors in Power Query](#)
- [Connect to data in Power BI Desktop](#)
- [Using DirectQuery in Power BI](#)
- [What is an on-premises data gateway?](#)
- [Power BI report data sources in Power BI Report Server](#)

Datasets in the Power BI service

Article • 01/23/2023

This article provides a technical explanation of Power BI datasets.

Dataset types

Power BI datasets represent a source of data that's ready for reporting and visualization. You can create Power BI datasets in the following ways:

- Connect to an existing data model that isn't hosted in Power BI.
- Upload a Power BI Desktop file that contains a model.
- Upload an Excel workbook that contains one or more Excel tables and/or a workbook data model, or upload a comma-separated values (CSV) file.
- Use the Power BI service to create a [push dataset](#).
- Use the Power BI service to create a [streaming or hybrid streaming dataset](#).

Except for streaming datasets, datasets represent data models, which use the mature modeling technologies of [Analysis Services](#).

ⓘ Note

Power BI documentation sometimes uses the terms *dataset* and *model* interchangeably. A *dataset* in the Power BI service refers to a *model* from a development perspective. In a documentation context, the terms mean much the same thing.

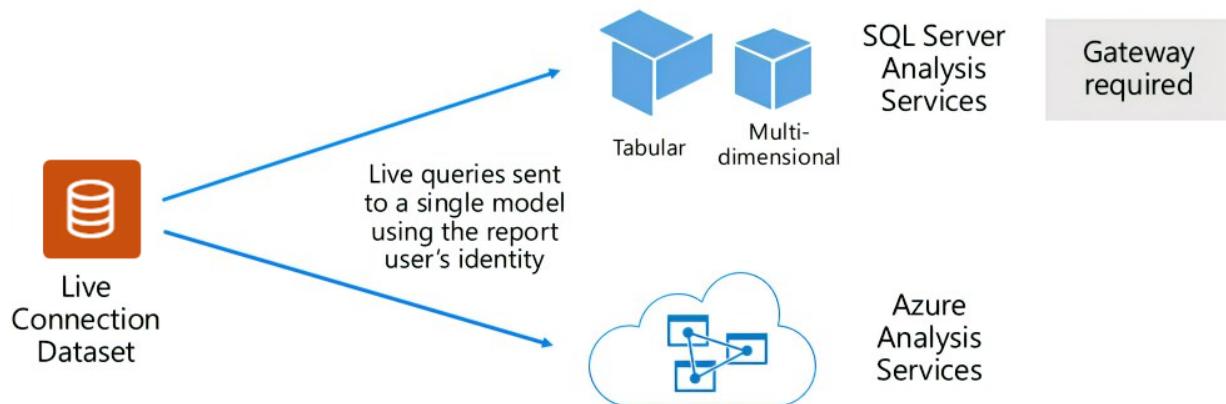
External-hosted models

There are two types of external-hosted models: SQL Server Analysis Services and [Azure Analysis Services](#).

To connect to a SQL Server Analysis Services model, you must install an [on-premises data gateway](#) either on-premises or on a virtual machine-hosted infrastructure-as-a-service (IaaS). Azure Analysis Services doesn't require a gateway.

It often makes sense to connect to Analysis Services when there are existing model investments, which typically form part of an enterprise data warehouse (EDW). Power BI can make a *live connection* to Analysis Services, and enforce data permissions by using the identity of the Power BI report user.

SQL Server Analysis Services supports both multidimensional models, or cubes, and tabular models. As the following image shows, a live connection dataset passes queries to external-hosted models.



Power BI Desktop-developed models

You can use Power BI Desktop, a client application for Power BI development, to develop a model. A Power BI Desktop model is effectively an Analysis Services tabular model.

You can develop three different types, or *modes*, of models by using Power BI Desktop: Import, DirectQuery, and Composite. You develop models by importing data from dataflows and then integrating them with external data sources. The mode depends on whether data is imported into the model, or whether it remains in the data source. For more information about the modes, see [Dataset modes in the Power BI service](#).

Row-level security

External-hosted models and Power BI desktop models can enforce row-level security (RLS) to limit the data that certain users can retrieve. For example, users assigned to a **Salespeople** security group might be able to view report data only for the sales regions they're assigned to. RLS roles are *dynamic* or *static*. Dynamic roles filter by the report user, while static roles apply the same filters for all users assigned to the role. For more information, see [Row-level security \(RLS\) with Power BI](#).

Excel workbook models

Creating datasets based on [Excel workbooks](#) or [CSV files](#) automatically creates a model. Imported Excel tables and CSV data create model tables, while Excel workbook data transposes to create a Power BI model. In all cases, file data imports into a model.

Summary

In summary:

- Power BI datasets that represent models are either hosted in the Power BI service, or are externally hosted by Analysis Services.
- Dataset models can store imported data, or issue pass-through query requests to underlying data sources, or do both.

Considerations

The following important facts and considerations apply to Power BI datasets that represent models:

- SQL Server Analysis Services-hosted models need a gateway to do live connection queries.
- To query Power BI-hosted models that import data, you must fully load them into memory.
- Power BI-hosted models that use Import need refresh to keep data current, and must use gateways when source data isn't accessible directly over the internet.
- Power BI-hosted Import models can refresh according to a schedule, or a user can trigger on-demand refresh in the Power BI service.
- Power BI-hosted models that use [DirectQuery](#) mode require connectivity to the source data. Power BI issues queries to the source data to retrieve current data. This mode must use gateways when source data isn't accessible directly over the internet.
- Models can enforce RLS rules to filter data access to certain users.
- You can use the [Datasets - Take Over In Group API](#) to take over ownership if a dataset owner leaves the organization.

To successfully deploy and manage Power BI datasets, you should understand the following factors:

- The model design itself, including its data preparation queries, relationships, and calculations.
- The following configurations that can significantly impact Power BI capacity resources:
 - Where models are hosted
 - The storage mode
 - Any dependencies on gateways
 - The size of imported data
 - Model refresh type and frequency

Next steps

- Dataset modes in the Power BI service
- Questions? [Ask the Power BI Community](#)↗
- Suggestions? [Contribute ideas to improve Power BI](#)↗

Dataset modes in the Power BI service

Article • 01/06/2023

This article provides a technical explanation of Power BI dataset modes. It applies to datasets that represent a live connection to an external-hosted Analysis Services model, and also to models developed in Power BI Desktop. The article emphasizes the rationale for each mode, and possible impacts on Power BI capacity resources.

The three dataset modes are:

- [Import](#)
- [DirectQuery](#)
- [Composite](#)

Import mode

Import mode is the most common mode used to develop datasets. This mode delivers fast performance thanks to in-memory querying. It also offers design flexibility to modelers, and support for specific Power BI service features (Q&A, Quick Insights, etc.). Because of these strengths, it's the default mode when creating a new Power BI Desktop solution.

It's important to understand that imported data is always stored to disk. When queried or refreshed, the data must be fully loaded into memory of the Power BI capacity. Once in memory, Import models can then achieve very fast query results. It's also important to understand that there's no concept of an Import model being partially loaded into memory.

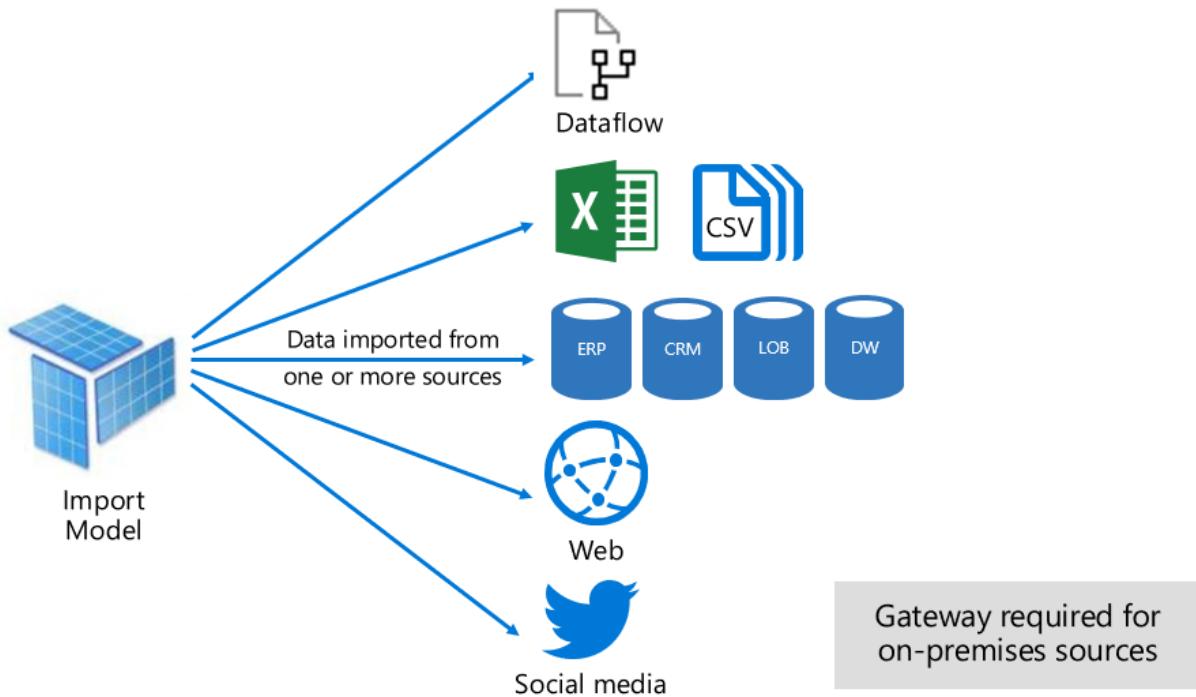
When refreshed, data is compressed and optimized and then stored to disk by the VertiPaq storage engine. When loaded from disk into memory, it's possible to see 10-times compression. So, it's reasonable to expect that 10 GB of source data can compress to about 1 GB in size. Storage size on disk can achieve a 20% reduction from the compressed size. The difference in size can be determined by comparing the Power BI Desktop file size with the Task Manager memory usage of the file.

Design flexibility can be achieved in three ways:

- Integrate data by caching data from dataflows, and external data sources, whatever the data source type or format.
- Use the entire set of [Power Query M formula language](#), referred to as *M*, functions when creating data preparation queries.

- Apply the entire set of [Data Analysis Expressions \(DAX\)](#) functions when enhancing the model with business logic. There's support for calculated columns, calculated tables, and measures.

As shown in the following image, an Import model can integrate data from any number of supported data source types.



However, while there are compelling advantages associated with Import models, there are disadvantages, too:

- The entire model must be loaded to memory before Power BI can query the model, which can place pressure on available capacity resources, especially as the number and size of Import models grow.
- Model data is only as current as the latest refresh, and so Import models need to be refreshed, usually on a scheduled basis.
- A full refresh removes all data from all tables and reloads it from the data source. This operation can be expensive in terms of time and resources for the Power BI service, and the data sources.

! Note

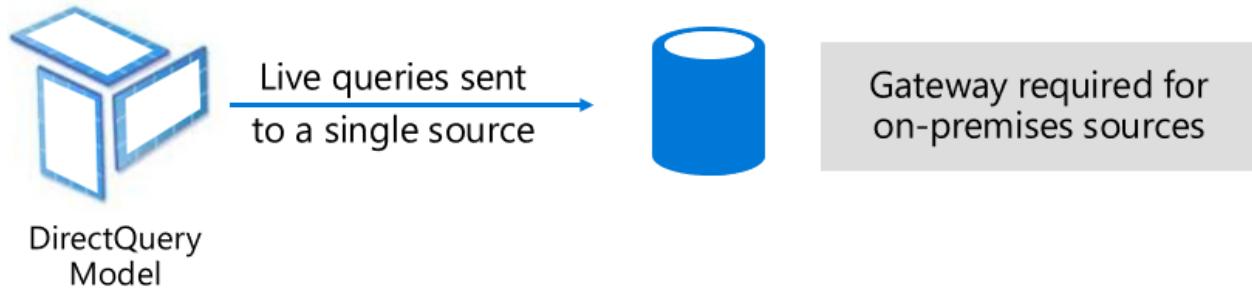
Power BI can achieve incremental refresh to avoid truncating and reloading entire tables. For more information, including supported plans and licensing, see [Incremental refresh and real-time data for datasets](#).

From a Power BI service resource perspective, Import models require:

- Sufficient memory to load the model when it's queried or refreshed.
- Processing resources and extra memory resources to refresh data.

DirectQuery mode

DirectQuery mode is an alternative to Import mode. Models developed in DirectQuery mode don't import data. Instead, they consist only of metadata defining the model structure. When the model is queried, native queries are used to retrieve data from the underlying data source.



There are two main reasons to consider developing a DirectQuery model:

- When data volumes are too large, even when [data reduction methods](#) are applied, to load into a model, or practically refresh.
- When reports and dashboards need to deliver *near real-time* data, beyond what can be achieved within scheduled refresh limits. Scheduled refresh limits are eight times a day for shared capacity, and 48 times a day for a Premium capacity.

There are several advantages associated with DirectQuery models:

- Import model size limits don't apply.
- Models don't require scheduled data refresh.
- Report users see the latest data when interacting with report filters and slicers. Also, report users can refresh the entire report to retrieve current data.
- Real-time reports can be developed by using the [Automatic page refresh](#) feature.
- Dashboard tiles, when based on DirectQuery models, can update automatically as frequently as every 15 minutes.

However, there are some limitations associated with DirectQuery models:

- Power Query/Mashup expressions can only be functions that can be transposed to native queries understood by the data source.
- DAX formulas are limited to use only functions that can be transposed to native queries understood by the data source. Calculated tables aren't supported.
- Quick Insights features aren't supported.

From a Power BI service resource perspective, DirectQuery models require:

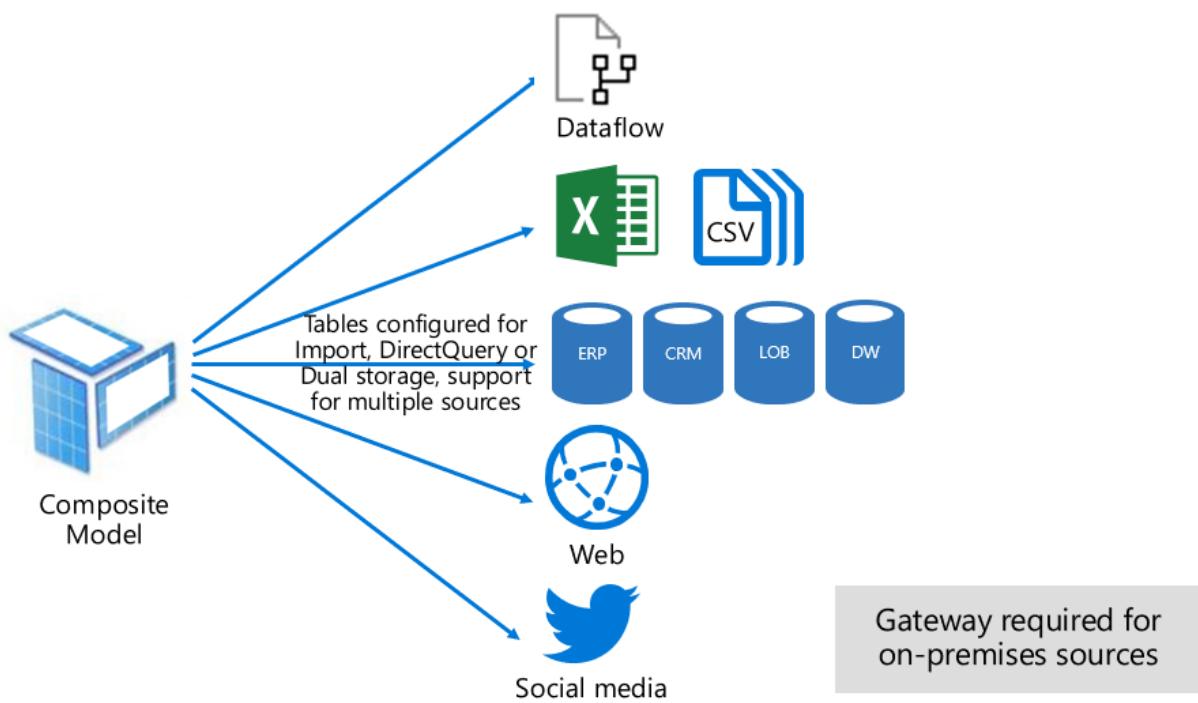
- Minimal memory to load the model (metadata only) when it's queried.
- Sometimes the Power BI service must use significant processor resources to generate and process queries sent to the data source. When this situation arises, it can affect throughput, especially when concurrent users are querying the model.

For more information, see [Use DirectQuery in Power BI Desktop](#).

Composite mode

Composite mode can mix Import and DirectQuery modes, or integrate multiple DirectQuery data sources. Models developed in Composite mode support configuring the storage mode for each model table. This mode also supports calculated tables, defined with DAX.

The table storage mode can be configured as Import, DirectQuery, or Dual. A table configured as Dual storage mode is both Import and DirectQuery, and this setting allows the Power BI service to determine the most efficient mode to use on a query-by-query basis.



Composite models strive to deliver the best of Import and DirectQuery modes. When configured appropriately, they can combine the high query performance of in-memory models with the ability to retrieve near real-time data from data sources.

For more information, see [Use composite models in Power BI Desktop](#).

Pure Import and DirectQuery tables

Data modelers who develop Composite models are likely to configure dimension-type tables in Import or Dual storage mode, and fact-type tables in DirectQuery mode. For more information about model table roles, see [Understand star schema and the importance for Power BI](#).

For example, consider a model with a **Product** dimension-type table in Dual mode, and a **Sales** fact-type table in DirectQuery mode. The **Product** table could be efficiently and quickly queried from in-memory to render a report slicer. The **Sales** table could also be queried in DirectQuery mode with the related **Product** table. The latter query could enable the generation of a single efficient native SQL query that joins **Product** and **Sales** tables, and filters by the slicer values.

Hybrid tables

Data modelers who develop Composite models can also configure fact tables as hybrid tables. A hybrid table is a table with one or multiple Import partitions and one DirectQuery partition. The advantage of a hybrid table is it could be efficiently and quickly queried from in-memory while at the same time including the latest data changes from the data source that occurred after the last import cycle, as the following visualization illustrates.

Granularity	Name	Row Count
Year	2011	295,489,717
Year	2012	297,678,498
Year	2013	295,575,442
Year	2014	292,477,875
Year	2015	297,780,469
Year	2016	294,060,081
Year	2017	300,419,682
Year	2018	296,541,108
Year	2019	292,787,420
Year	2020	299,273,979
Quarter	2021Q1	74,135,277
Month	2021Q104	24,939,498
Day	2021Q10501	820,805
Day	2021Q10502	826,885
Day	2021Q10503	821,043
Day-DirectQuery	2021Q10504-DQ	271,110
Total		3,063,898,887

The diagram illustrates the row count distribution for a hybrid table. A vertical bracket on the right side groups the rows into three categories: "Archived: Import" (years 2011-2020), "Incremental refresh: Import" (quarters/months 2021Q1-Q3), and "Real time: DirectQuery" (days Q10501-Q10504).

The easiest way to create a hybrid table is to configure an incremental refresh policy in Power BI Desktop and enable the option **Get the latest data in real time with DirectQuery (Premium only)**. When Power BI applies an incremental refresh policy that has this option enabled, it partitions the table like the partitioning scheme displayed in the previous diagram. To ensure good performance, configure your dimension-type tables in Dual storage mode so that Power BI can generate efficient native SQL queries when querying the DirectQuery partition.

ⓘ Note

Power BI supports hybrid tables only when the dataset is hosted in workspaces on Premium capacities. Accordingly, you must upload your dataset to a Premium workspace if you configure an incremental refresh policy with the option to get the latest data in real time with DirectQuery. For more information, see [Incremental refresh and real-time data for datasets](#).

It's also possible to convert an Import table to a hybrid table by adding a DirectQuery partition using Tabular Model Scripting Language (TMSL) or the Tabular Object Model (TOM) or by using a third-party tool. For example, you can partition a fact table such that the bulk of the data is left in the data warehouse while only a fraction of the most recent data is imported. This approach can help to optimize performance if the bulk of this data is historical data that is infrequently accessed. A hybrid table can have multiple Import partitions, but only one DirectQuery partition.

Next steps

- [Storage mode in Power BI Desktop](#)
- [Using DirectQuery in Power BI](#)
- [Use composite models in Power BI Desktop](#)
- More questions? [Try asking the Power BI Community](#) ↗

Power BI data source prerequisites

Article • 02/08/2023

For data sources, Power BI supports specific provider versions and data source versions, and certain objects. For more information about available Power BI data sources, see [Data sources](#).

The following table describes Power BI data source requirements.

Data source	Provider	Minimum provider version	Minimum data source version	Supported data source objects	Download link
SQL Server	ADO.net (built into .NET Framework)	.NET Framework 3.5 (only)	SQL Server 2005+	Tables/Views, Scalar functions, Table functions	Included in .NET Framework 3.5 or above
Access	Microsoft Access Database Engine (ACE)	ACE 2010 SP1	No restriction	Tables/Views	Download link
Excel (.xls files only) (see note 1)	Microsoft Access Database Engine (ACE)	ACE 2010 SP1	No restriction	Tables, Sheets	Download link
Oracle (see note 2)	ODP.NET	ODAC 11.2 Release 5 (11.2.0.3.20)	9.x+	Tables/Views	Download link
MySQL	Connector/Net	6.6.5	5.1	Tables/Views, Scalar functions	Download link
PostgreSQL	NPGSQL ADO.NET provider (Shipped with Power BI Desktop)	4.0.10	9.4	Tables/Views	Download link
Teradata	.NET Data Provider for Teradata	14+	12+	Tables/Views	Download link
SAP Sybase SQL Anywhere	iAnywhere.Data.SQLAnywhere for .NET 3.5	16+	16+	Tables/Views	Download link

 **Note**

Excel files that have an .xlsx extension do not require a separate provider installation.

 **Note**

The Oracle providers also require Oracle client software (version 8.1.7+).

Using enhanced dataset metadata

Article • 01/18/2023

When Power BI Desktop creates reports, it also creates dataset metadata in the corresponding PBIX and PBIT files. Previously, the metadata was stored in a format that was specific to Power BI Desktop. The metadata used base-64 encoded M expressions and data sources. Power BI made assumptions about how that metadata was stored.

With the release of the *enhanced dataset metadata* feature, many of these limitations are removed. PBIX files are automatically upgraded to enhanced metadata upon opening the file. With enhanced dataset metadata, metadata created by Power BI Desktop uses a format similar to what is used for Analysis Services tabular models, based on the [Tabular Object Model](#).

The enhanced dataset metadata feature is strategic and foundational. Future Power BI functionality will be built upon its metadata. These other capabilities stand to benefit from enhanced dataset metadata:

- [XMLA read/write](#) for management of Power BI datasets.
- Migration of Analysis Services workloads to Power BI to benefit from next-generation features.

Upgrade

Your reports are automatically upgraded to the enhanced metadata format when you open them in the latest version of Power BI Desktop. If the report was saved with unapplied query changes, or there was an error during the auto-upgrade, there's a warning on the report canvas that you still need to upgrade. Selecting **Upgrade report** applies any pending changes and upgrades the data model to the new format.

Exclude table from report refresh

Once a data model has been upgraded to the enhanced metadata format, some metadata that was previously only used in Power BI Desktop is now respected in the Power BI service as well. This metadata includes the **Include in Report Refresh** option. For upgraded models, if the **Include in Report Refresh** option is unselected in the Power Query Editor, then that table isn't refreshed when the report or dataset is refreshed in Power BI Desktop or the Power BI service. Reports already published in the Power BI service that aren't yet upgraded to the new enhanced metadata formal need to be upgraded in Power BI Desktop before this new behavior takes effect.

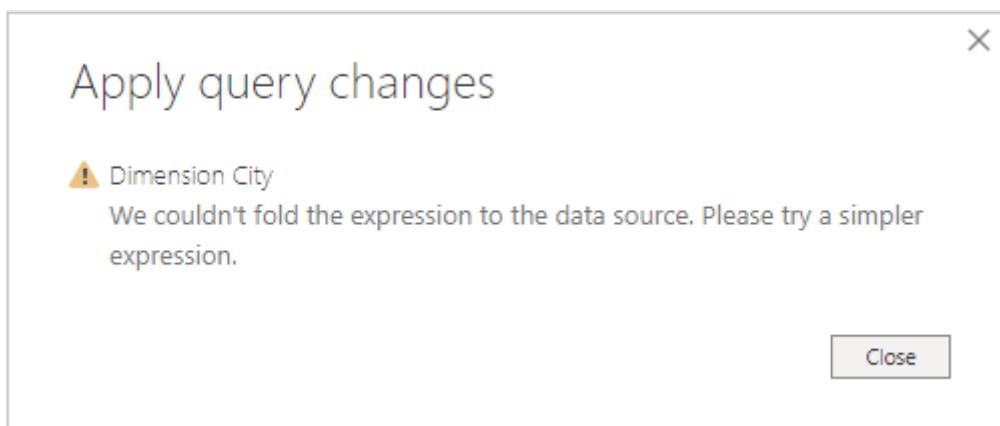
Considerations and limitations

Before enhanced metadata support, for SQL Server, Oracle, Teradata, and legacy HANA connections, Power BI Desktop added a native query to the data model. This query is used by the Power BI service data models. With enhanced metadata support, the Power BI service data model regenerates the native query at runtime. It doesn't use the query that Power BI Desktop created. In most cases, this retrieval resolves itself correctly, but some transformations don't work without reading underlying data. You might see some errors in reports that previously worked. For example, an error might say:

- Unable to convert an M query in table 'Dimension City' into a native source query.
Try again later or contact support. If you contact support, provide these details.

You can fix your queries in three different places in Power BI Desktop:

- When you apply changes or do a refresh.
- In a warning bar in the Power Query Editor informing you that the expression couldn't be folded to the data source.



- When you run evaluations when you open a report to check if you have unsupported queries. Running these evaluations can result in performance implications.

Certain character combinations in M expressions that would be unsupported in the Tabular Object Model (TOM) are also unsupported in the enhanced dataset metadata environment.

Next steps

You can do all sorts of things with Power BI Desktop. For more information on its capabilities, check out the following resources:

- [What is Power BI Desktop?](#)

- What's new in Power BI?
- Query overview with Power BI Desktop
- Data types in Power BI Desktop
- Shape and combine data with Power BI Desktop
- Common query tasks in Power BI Desktop

Work with multidimensional models in Power BI

Article • 01/19/2023

You can connect to multidimensional models in Power BI, and create reports that visualize all sorts of data within the model. With multidimensional models, Power BI applies rules to how it processes data, based on which column is defined as the *default member*.

With multidimensional models, Power BI handles data from the model based on where the column that contains the **Default Member** is used. The **DefaultMember** property value for an attribute hierarchy is set in CSDL (Conceptual Schema Definition Language) for a particular column in a multidimensional model. For more information about the default member, see [Attribute properties - Define a default member](#). When a data analysis expression (DAX) query is executed, the default member specified in the model is applied automatically.

This article describes how Power BI behaves under various circumstances when working with multidimensional models, based on where the default member is found.

Work with filter cards

When you create a filter card on a field with a default member, the default member field value is selected automatically in the filter card. The result is that all visuals affected by the filter card retain their default models in the database. The values in such filter cards reflect that default member.

If the default member is removed, deselecting the value clears it for all visuals to which the filter card applies, and the values displayed don't reflect the default member.

For example, imagine we have a *Currency* column and a default member set to *USD*:

- In this example case, if we have a card that shows *Total Sales*, the value will have the default member applied and the sales that correspond to *USD*.
- If we drag *Currency* to the filter card pane, we see *USD* as the default value selected. The value of *Total Sales* remains the same, since the default member is applied.
- However, if we deselect the *USD* value from the filter card, the default member for *Currency* is cleared, and now *Total Sales* reflects all currencies.

- When we select another value in the filter card (let's say we select *EURO*), along the default member, the *Total Sales* reflects the filter *Currency IN {USD, EURO}*.

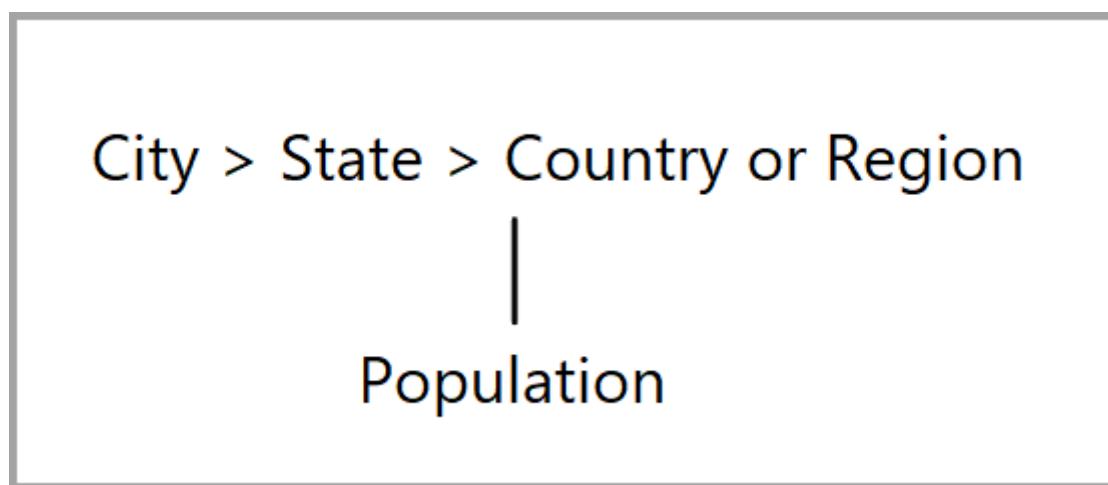
Group visuals

In Power BI, whenever you group a visual on a column that has a default member, Power BI clears the default member for that column and its attribute relationship path. This behavior ensures the visual displays all values, instead of just the default values.

Attribute relationship paths (ARPs)

Attribute relationship paths (ARPs) provide default members with powerful capabilities, but also introduce a certain amount of complexity. When ARPs are encountered, Power BI follows the path of ARPs to clear other default members for other columns to provide consistent, and precise handling of data for visuals.

Let's look at an example to clarify the behavior. Consider the following configuration of ARPs:



Now let's imagine that the following default members are set for these columns:

- City > Seattle
- State > WA
- Country/Region > US
- Population > Large

Now let's examine what happens when each column is used in Power BI. When visuals group on the following columns, here are the results:

- **City** - Power BI displays all the cities by clearing all the default members for *City, State, Country/Region* but preserves the default member for *Population*; Power BI

cleared the entire ARP for *City*.

(!) Note

Population isn't in the ARP path of *City*, it's solely related to *State* and thus Power BI doesn't clear it.

- **State** - Power BI displays all the *States* by clearing all default members for *City*, *State*, *Country/Region* and *Population*.
- **Country/Region** - Power BI displays all the countries/regions by clearing all default members for *City*, *State* and *Country/Region*, but preserves the default member for *Population*.
- **City and State** - Power BI clears all default members for all columns.

Groups displayed in the visual have their entire ARP path cleared.

If a group isn't displayed in the visual, but is part of the ARP path of another grouped-on column, the following applies:

- Not all branches of the ARP path are cleared automatically.
- That group is still filtered by that uncleared default member.

Slicers and filter cards

When you work with slicers or filter cards, the following behavior occurs:

- When a slicer or filter card is loaded with data, Power BI groups on the column in the visual, so the display behavior is the same as described in the previous section.

Since slicers and filter cards are often used to interact with other visuals, the logic of clearing default members for the affected visuals occurs as explained in the following table.

For this table, we use the same example data from earlier in this article:

Visual with groups	Filter card selection	Expected result in the visual
City	City = Default Member (Seattle)	Only the DM value of City shows up
City	City = Portland	Only Portland shows up
City	City - All	All cities, but from states that have Large population (due to the DM on Population)
City	City - All, Population - All	All cities
State	City = Default Member (Seattle)	WA, due to the default of Seattle
State	City - All	All states
State	Population - Default Member (Large)	Only states with large population
State	City = Default Member (Seattle) Country - All	WA In this case the clearing of the whole ARP path for Country (Country - State - City) due to Country - All But we will keep only City = Default The end result will be: All states, All countries but Cities = Seattle
State	Country - All Population - Default Member (Large)	Clear Country, State, City + Keep Population DM States that have the default population

The following rules apply to the way Power BI behaves in these circumstances.

Power BI clears a default member for a specified column, if:

- Power BI groups on that column.
- Power BI groups on a column related to that column (anywhere in the ARP, up or down).
- Power BI filters on a column that's in the ARP (up or down).
- The column has a filter card with *ALL* stated.
- The column has a filter card with any value selected (Power BI receives a filter for the column).

Power BI doesn't clear a default member for a specified column, if:

- The column has a filter card with default stated, and Power BI is grouping on a column in its ARP.
- The column is above another column in the ARP, and Power BI has a filter card for that other column in default state.

Next steps

This article described the behavior of Power BI when working with default members in multidimensional models. You might also be interested in the following articles:

- [Show items with no data in Power BI](#)
- [Data sources in Power BI Desktop](#)

DirectQuery in Power BI

Article • 06/21/2023

In Power BI Desktop or the Power BI service, you can connect to many different data sources in different ways. You can *import* data to Power BI, which is the most common way to get data. You can also connect directly to some data in its original source repository, which is called *DirectQuery*. This article primarily discusses DirectQuery capabilities.

This article describes:

- The different Power BI data connectivity options.
- Guidance about when to use DirectQuery rather than import.
- Limitations and implications of using DirectQuery.
- Recommendations for successfully using DirectQuery.
- How to diagnose DirectQuery performance issues.

The article focuses on the DirectQuery workflow when you create a report in Power BI Desktop, but also covers connecting through DirectQuery in the Power BI service.

Note

DirectQuery is also a feature of SQL Server Analysis Services. That feature shares many details with Direct Query in Power BI, but there are also important differences. This article primarily covers DirectQuery with Power BI, not SQL Server Analysis Services.

For more information about using DirectQuery with SQL Server Analysis Services, see [Use DirectQuery for Power BI datasets and Analysis Services \(preview\)](#). You can also download the PDF [DirectQuery in SQL Server 2016 Analysis Services](#).

Power BI data connectivity modes

Power BI connects to a large number of varied data sources, such as:

- Online services like Salesforce and Dynamics 365.
- Databases like SQL Server, Access, and Amazon Redshift.
- Simple files in Excel, JSON, and other formats.
- Other data sources like Spark, websites, and Microsoft Exchange.

You can import data from these sources into Power BI. For some sources, you can also connect using DirectQuery. For a summary of the sources that support DirectQuery, see [Data sources supported by DirectQuery](#). DirectQuery-enabled sources are primarily sources that can deliver good interactive query performance.

You should import data into Power BI wherever possible. Importing takes advantage of the high-performance query engine of Power BI, and provides a highly interactive, fully featured experience.

If you can't meet your goals by importing data, for example if the data changes frequently and reports must reflect the latest data, consider using DirectQuery. DirectQuery is feasible only when the underlying data source can provide interactive query results in less than five seconds for a typical aggregate query, and can handle the generated query load. Carefully consider the limitations and implications of using DirectQuery.

Power BI import and DirectQuery capabilities evolve over time. Changes that provide more flexibility when using imported data let you import more often, and eliminate some of the drawbacks of using DirectQuery. Regardless of improvements, the performance of the underlying data source is a major consideration when using DirectQuery. If an underlying data source is slow, using DirectQuery for that source remains unfeasible.

The following sections cover the three options for connecting to data: import, DirectQuery, and live connection. The remainder of the article focuses on DirectQuery.

Import connections

When you connect to a data source like SQL Server and import data in Power BI Desktop, the following results occur:

- When you initially **Get Data**, each set of tables you select defines a query that returns a set of data. You can edit those queries before loading the data, for example to apply filters, aggregate the data, or join different tables.
- Upon load, all the data defined by the queries imports into the Power BI cache.
- Building a visual within Power BI Desktop queries the cached data. The Power BI store ensures the query is fast, and that all changes to the visual reflect immediately.
- Visuals don't reflect changes to the underlying data in the data store. You need to reimport to refresh the data.

- Publishing the report to the Power BI service as a *.pbix* file creates and uploads a dataset that includes the imported data. You can then schedule data refresh, for example reimport the data every day. Depending on the location of the original data source, it might be necessary to configure an on-premises data gateway for the refresh.
- Opening an existing report or authoring a new report in the Power BI service queries the imported data again, ensuring interactivity.
- You can pin visuals or entire report pages as dashboard tiles in the Power BI service. The tiles automatically refresh whenever the underlying dataset refreshes.

DirectQuery connections

When you use DirectQuery to connect to a data source in Power BI Desktop, the following results occur:

- You use **Get Data** to select the source. For relational sources, you can still select a set of tables that define a query that logically returns a set of data. For multidimensional sources like SAP Business Warehouse (SAP BW), you select only the source.
- Upon load, no data is imported into the Power BI store. Instead, when you build a visual, Power BI Desktop sends queries to the underlying data source to retrieve the necessary data. The time it takes to refresh the visual depends on the performance of the underlying data source.
- Any changes to the underlying data aren't immediately reflected in existing visuals. It's still necessary to refresh. Power BI Desktop resends the necessary queries for each visual, and updates the visual as necessary.
- Publishing the report to the Power BI service creates and uploads a dataset, the same as for import. However, that dataset includes no data.
- Opening an existing report or authoring a new report in the Power BI service queries the underlying data source to retrieve the necessary data. Depending upon the location of the original data source, it might be necessary to configure an on-premises data gateway to get the data.
- You can pin visuals or entire report pages as dashboard tiles. To ensure that opening a dashboard is fast, the tiles automatically refresh on a schedule, for example every hour. You can control refresh frequency depending on how frequently the data changes and the importance of seeing the latest data.

- When you open a dashboard, the tiles reflect the data at the time of the last refresh, not necessarily the latest changes made to the underlying source. You can refresh an open dashboard to ensure that it's current.

Live connections

When you connect to SQL Server Analysis Services, you can choose to import the data or use a *live connection* to the selected data model. Using a live connection is similar to DirectQuery. No data is imported, and the underlying data source is queried to refresh visuals.

For example, when you use import to connect to SQL Server Analysis Services, you define a query against the external SQL Server Analysis Services source, and import the data. If you connect live, you don't define a query, and the entire external model shows in the field list.

This situation also applies when you connect to the following sources, except there's no option to import the data:

- Power BI datasets, for example connecting to a Power BI dataset that's already published to the service, to author a new report over it.
- Microsoft Dataverse.

When you publish SQL Server Analysis Services reports that use live connections, the behavior in the Power BI service is similar to DirectQuery reports in the following ways:

- Opening an existing report or authoring a new report in the Power BI service queries the underlying SQL Server Analysis Services source, possibly requiring an on-premises data gateway.
- Dashboard tiles automatically refresh on a schedule, such as every hour.

A live connection also differs from DirectQuery in several ways. For example, live connections always pass the identity of the user opening the report to the underlying SQL Server Analysis Services source.

DirectQuery use cases

Connecting with DirectQuery can be useful in the following scenarios. In several of these cases, leaving the data in its original source location is necessary or beneficial.

DirectQuery in Power BI offers the greatest benefits in the following scenarios:

- The data changes frequently, and you need near real-time reporting.
- You need to handle large data without having to pre-aggregate.
- The underlying source defines and applies security rules.
- Data sovereignty restrictions apply.
- The source is a multidimensional source containing measures, such as SAP BW.

Data changes frequently, and you need near real-time reporting

You can refresh models with imported data at most once per hour, more frequently with Power BI Pro or Power BI Premium subscriptions. If the data is continually changing, and it's necessary for reports to show the latest data, using import with scheduled refresh might not meet your needs. You can stream data directly into Power BI, although there are limits on the data volumes supported for this case.

Using DirectQuery means that opening or refreshing a report or dashboard always shows the latest data in the source. The dashboard tiles can also be updated more frequently, as often as every 15 minutes.

Data is very large

If the data is very large, it's not feasible to import all of it. DirectQuery requires no large transfer of data, because it queries data in place. However, large data might also make the performance of queries against that underlying source too slow.

You don't always have to import full detailed data. The Power Query Editor makes it easy to pre-aggregate data during import. Technically, it's possible to import exactly the aggregate data you need for each visual. While DirectQuery is the simplest approach to large data, importing aggregate data might offer a solution if the underlying data source is too slow for DirectQuery.

These details relate to using Power BI alone. For more information about using large models in Power BI, see [large datasets in Power BI Premium](#). There's no restriction on how frequently the data can be refreshed.

The underlying source defines security rules

When you import data, Power BI connects to the data source by using the current user's Power BI Desktop credentials, or the credentials configured for scheduled refresh from the Power BI service. In publishing and sharing reports that have imported data, you

must be careful to share only with users allowed to see the data, or you must define row-level security as part of the dataset.

DirectQuery lets a report viewer's credentials pass through to the underlying source, which applies security rules. DirectQuery supports single sign-on (SSO) to Azure SQL data sources, and through a data gateway to on-premises SQL servers. For more information, see [Overview of single sign-on \(SSO\) for gateways in Power BI](#).

Data sovereignty restrictions apply

Some organizations have policies around data sovereignty, meaning that data can't leave the organization premises. This data presents issues for solutions based on data import. With DirectQuery, the data remains in the underlying source location. However, even with DirectQuery, the Power BI service keeps some caches of data at the visual level, because of scheduled refresh of tiles.

The underlying data source uses measures

An underlying data source such as SAP HANA or SAP BW contains *measures*. Measures mean that imported data is already at a certain level of aggregation, as defined by the query. A visual that asks for data at a higher-level aggregate, such as **TotalSales** by **Year**, further aggregates the aggregate value. This aggregation is fine for additive measures, such as **Sum** and **Min**, but can be an issue for non-additive measures, such as **Average** and **DistinctCount**.

Easily getting the correct aggregate data needed for a visual directly from the source requires sending queries per visual, as in DirectQuery. When you connect to SAP BW, choosing DirectQuery allows this treatment of measures. For more information, see [DirectQuery and SAP BW](#).

Currently DirectQuery over SAP HANA treats data the same as a relational source, and produces behavior similar to import. For more information, see [DirectQuery and SAP HANA](#).

DirectQuery limitations

Using DirectQuery has some potentially negative implications. Some of these limitations differ slightly depending on the exact source you use. The following sections list general implications of using DirectQuery, and limitations related to performance, security, transformations, modeling, and reporting.

General implications

Some general implications and limitations of using DirectQuery follow:

- **If data changes, you must refresh to show the latest data.** Given the use of caches, there's no guarantee that visuals always show the latest data. For example, a visual might show transactions in the past day. A slicer change might refresh the visual to show transactions for the past two days, including recent, newly arrived transactions. But returning the slicer to its original value could result in it again showing the cached previous value. Select **Refresh** to clear any caches and refresh all the visuals on the page to show the latest data.
- **If data changes, there's no guarantee of consistency between visuals.** Different visuals, whether on the same page or on different pages, might be refreshed at different times. If the data in the underlying source is changing, there's no guarantee that each visual shows the data at the same point in time.

Given that more than one query might be required for a single visual, for example, to obtain the details and the totals, even consistency within a single visual isn't guaranteed. To guarantee this consistency would require the overhead of refreshing all visuals whenever any visual refreshed, along with using costly features like snapshot isolation in the underlying data source.

You can mitigate this issue to a large extent by selecting **Refresh** to refresh all of the visuals on the page. Even for import mode, there's a similar problem of maintaining consistency when you import data from more than one table.

- **You must refresh in Power BI Desktop to reflect schema changes.** After a report is published, **Refresh** in the Power BI service refreshes the visuals in the report. But if the underlying source schema changes, the Power BI service doesn't automatically update the available fields list. If tables or columns are removed from the underlying source, it might result in query failure upon refresh. To update the fields in the model to reflect the changes, you must open the report in Power BI Desktop and choose **Refresh**.
- **A limit of 1 million rows can return on any query.** There's a fixed limit of 1 million rows that can return in any single query to the underlying source. This limit generally has no practical implications, and visuals won't display that many points. However, the limit can occur in cases where Power BI doesn't fully optimize the queries sent, and requests some intermediate result that exceeds the limit.

The limit can also occur while building a visual, on the path to a more reasonable final state. For example, including **Customer** and **TotalSalesQuantity** could hit this

limit if there are more than 1 million customers, until you apply some filter. The error that returns is: **The resultset of a query to external data source has exceeded the maximum allowed size of '1000000' rows.**

ⓘ Note

Premium capacities let you exceed the one-million row limit. For more information, see [max intermediate row set count](#).

- You can't change a model from import to DirectQuery mode. You can switch a model from DirectQuery mode to import mode if you import all the necessary data. It's not possible to switch back to DirectQuery mode, primarily because of the feature set that DirectQuery mode doesn't support. For multidimensional sources like SAP BW, you can't switch from DirectQuery to import mode either, because of the different treatment of external measures.

Performance and load implications

When you use DirectQuery, the overall experience depends on the performance of the underlying data source. If refreshing each visual, for example after changing a slicer value, takes less than five seconds, the experience is reasonable, although might feel sluggish compared to the immediate response with imported data. If the slowness of the source causes individual visuals to take longer than tens of seconds to refresh, the experience becomes unreasonably poor. Queries might even time out.

Along with the performance of the underlying source, the load placed on the source also impacts performance. Each user who opens a shared report, and each dashboard tile that refreshes, sends at least one query per visual to the underlying source. The source must be able to handle such a query load while maintaining reasonable performance.

Security implications

Unless the underlying data source uses SSO, a DirectQuery report always uses the same fixed credentials to connect to the source once it's published to the Power BI service. Immediately after you publish a DirectQuery report, you must configure the credentials of the user to use. Until you configure the credentials, trying to open the report in the Power BI service results in an error.

Once you provide the user credentials, Power BI uses those credentials for whoever opens the report, the same as for imported data. Every user sees the same data, unless

row-level security is defined as part of the report. You must pay the same attention to sharing the report as for imported data, even if there are security rules defined in the underlying source.

- Connecting to Power BI datasets and Analysis Services in DirectQuery mode always uses SSO, so the security is similar to live connections to Analysis Services.
- Alternate credentials aren't supported when making DirectQuery connections to SQL Server from Power BI Desktop. You can use your current Windows credentials or database credentials.
- You can use multiple data sources in a DirectQuery model by using [composite models](#). When you use multiple data sources, it's important to understand the [security implications](#) of how data moves back and forth between the underlying data sources.

Data transformation limitations

DirectQuery limits the data transformations you can apply within Power Query Editor. With imported data, you can easily apply a sophisticated set of transformations to clean and reshape the data before using it to create visuals. For example, you can parse JSON documents, or pivot data from a column to a row form. These transformations are more limited in DirectQuery.

When you connect to an online analytical processing (OLAP) source like SAP BW, you can't define any transformations, and the entire external model is taken from the source. For relational sources like SQL Server, you can still define a set of transformations per query, but those transformations are limited for performance reasons.

Any transformations must be applied on every query to the underlying source, rather than once on data refresh. Transformations must be able to reasonably translate into a single native query. If you use a transformation that's too complex, you get an error that either it must be deleted or the connection model switched to import.

Also, the **Get Data** dialog or Power Query Editor use subselects within the queries they generate and send to retrieve data for a visual. Queries defined in Power Query Editor must be valid within this context. In particular, it's not possible to use a query with common table expressions, nor one that invokes stored procedures.

Modeling limitations

The term *modeling* in this context means the act of refining and enriching raw data as part of authoring a report using the data. Examples of modeling include:

- Defining relationships between tables.
- Adding new calculations, like calculated columns and measures.
- Renaming and hiding columns and measures.
- Defining hierarchies.
- Defining column formatting, default summarization, and sort order.
- Grouping or clustering values.

You can still make many of these model enrichments when you use DirectQuery, and use the principle of enriching the raw data to improve later consumption. However, some modeling capabilities aren't available or are limited with DirectQuery. The limitations are applied to avoid performance issues.

The following limitations are common to all DirectQuery sources. More limitations might apply to individual sources.

- **No built-in date hierarchy:** With imported data, every date/datetime column also has a built-in date hierarchy available by default. For example, if you import a table of sales orders that includes a column **OrderDate**, and you use **OrderDate** in a visual, you can choose the appropriate date level to use, such as year, month, or day. This built-in date hierarchy isn't available with DirectQuery. If there's a **Date** table available in the underlying source, as is common in many data warehouses, you can use the Data Analysis Expressions (DAX) time-intelligence functions as usual.
- **Date/time support only to the seconds level:** For datasets that use time columns, Power BI issues queries to the underlying DirectQuery source only up to the seconds detail level, not milliseconds. Remove milliseconds data from your source columns.
- **Limitations in calculated columns:** Calculated columns can only be intra-row, that is they can refer only to values of other columns of the same table, without using any aggregate functions. Also, the allowed DAX scalar functions, such as `LEFT()`, are limited to those functions that can be pushed to the underlying source. The functions vary depending upon the exact capabilities of the source. Functions that aren't supported aren't listed in autocomplete when authoring the DAX query for a calculated column, and result in an error if used.
- **No support for parent-child DAX functions:** When in DirectQuery mode, it's not possible to use the family of `DAX PATH()` functions that usually handle parent-child structures, such as charts of accounts or employee hierarchies.
- **No clustering:** When you use DirectQuery, you can't use the clustering capability to automatically find groups.

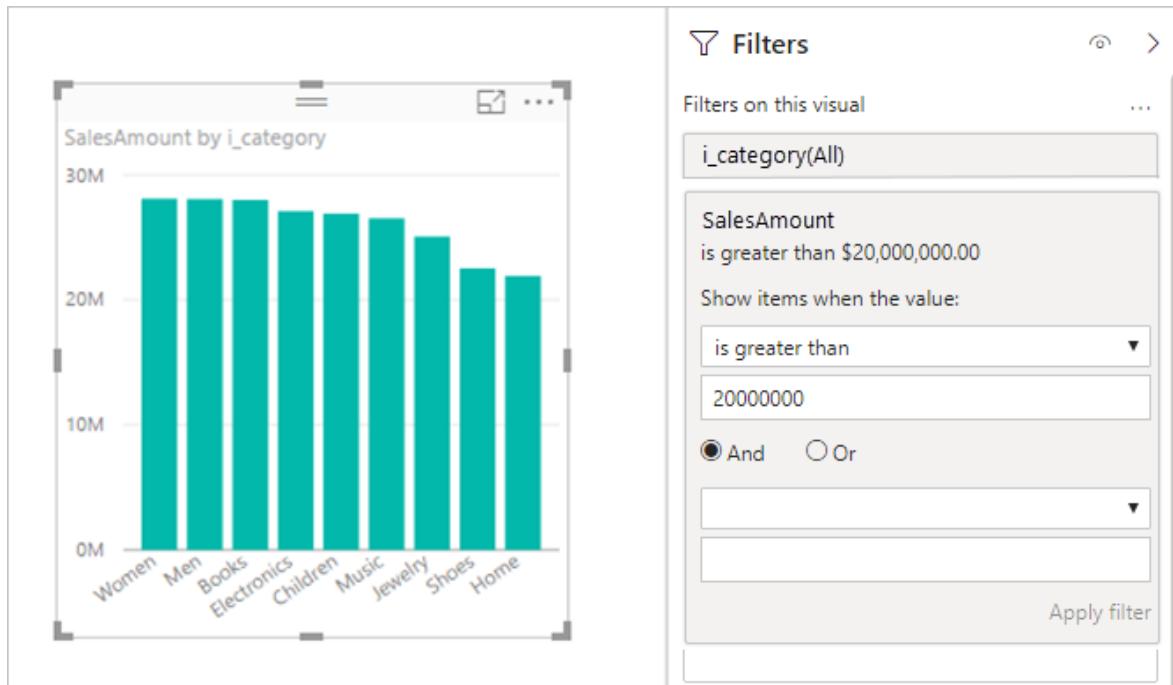
Reporting limitations

Almost all reporting capabilities are supported for DirectQuery models. As long as the underlying source offers a suitable level of performance, you can use the same set of visualizations as for imported data.

One general limitation is that the maximum length of data in a text column for DirectQuery datasets is 32,764 characters. Reporting on longer texts results in an error.

The following Power BI reporting capabilities can cause performance issues in DirectQuery-based reports:

- **Measure filters:** Visuals that use measures or aggregates of columns can contain filters in those measures. For example, the following graphic shows **SalesAmount** by **Category**, but only for categories with more than **20M** of sales.



This approach causes two queries to be sent to the underlying source:

- The first query retrieves the categories that meet the condition **SalesAmount** greater than 20 million.
- The second query retrieves the necessary data for the visual, which includes the categories that met the **WHERE** condition.

This approach generally works well if there are hundreds or thousands of categories, as in this example. Performance can degrade if the number of categories is much larger. The query fails if there are more than a million categories.

- **TopN filters:** You can define advanced filters to filter on only the top or bottom `N` values ranked by some measure. For example, filters can include the top 10 categories. This approach again sends two queries to the underlying source. However, the first query returns all categories from the underlying source, and then the `TopN` are determined based on the returned results. Depending on the cardinality of the column involved, this approach can lead to performance issues or query failures because of the one-million row limit on query results.
- **Median:** Any aggregation, such as `Sum` or `Count Distinct`, is pushed to the underlying source. However, usually the `median` aggregate isn't supported by the underlying source. For `median`, the detail data is retrieved from the underlying source, and the median is calculated from the returned results. This approach is reasonable for calculating the median over a relatively small number of results. Performance issues or query failures can arise if the cardinality is large because of the one-million row limit. For example, querying for **Median Country/Region Population** might be reasonable, but **Median Sales Price** might not be reasonable.
- **Advanced text filters like 'contains':** Advanced filtering on a text column allows filters like `contains` and `begins with`. These filters can result in degraded performance for some data sources. In particular, don't use the default `contains` filter if you need an exact match. Although the results might be the same depending on the actual data, the performance might be drastically different because of indexes.
- **Multi-select slicers:** By default, slicers only allow making a single selection. Allowing multi-selection in filters can cause performance issues. For example, if the user selects 10 products of interest, each new selection results in queries being sent to the source. Although the user can select the next item before the query completes, this approach results in extra load on the underlying source.
- **Totals on table visuals:** By default, tables and matrices display totals and subtotals. In many cases, getting the values for such totals requires sending separate queries to the underlying source. This requirement applies whenever you use `DistinctCount` aggregation, or in all cases that use DirectQuery over SAP BW or SAP HANA. You can switch off such totals by using the **Format** pane.

DirectQuery recommendations

This section provides high-level guidance on how to successfully use DirectQuery, given its implications.

Underlying data source performance

Validate that simple visuals refresh within five seconds, to provide a reasonable interactive experience. If visuals take longer than 30 seconds to refresh, it's likely that further issues following report publication will make the solution unworkable.

If queries are slow, examine the queries sent to the underlying source, and the reason for the slow performance. For more information, see [Performance diagnostics](#).

This article doesn't cover the wide range of database optimization recommendations across the full set of potential underlying sources. The following standard database practices apply to most situations:

- For better performance, base relationships on integer columns rather than joining columns of other data types.
- Create the appropriate indexes. Index creation generally means using column store indexes in sources that support them, for example SQL Server.
- Update any necessary statistics in the source.

Model design

When you define the model, follow this guidance:

- **Avoid complex queries in Power Query Editor.** Power Query Editor translates a complex query into a single SQL query. The single query appears in the subselect of every query sent to that table. If that query is complex, it might result in performance issues on every query sent. You can get the actual SQL query for a set of steps by right-clicking the last step under **Applied steps** in Power Query Editor and choosing **View Native Query**.
- **Keep measures simple.** At least initially, limit measures to simple aggregates. If the measures operate in a satisfactory manner, you can define more complex measures, but pay attention to performance.
- **Avoid relationships on calculated columns.** In databases where you need to do multi-column joins, Power BI doesn't allow basing relationships on multiple columns as the primary key or foreign key. The common workaround is to concatenate the columns by using a calculated column, and base the join on that column.

This workaround is reasonable for imported data, but for DirectQuery it results in a join on an expression. That result usually prevents using any indexes, and leads to

poor performance. The only workaround is to actually materialize the multiple columns into a single column in the underlying data source.

- **Avoid relationships on 'uniqueidentifier' columns.** Power BI doesn't natively support a `uniqueidentifier` datatype. Defining a relationship between `uniqueidentifier` columns results in a query with a join that involves a cast. Again, this approach commonly leads to poor performance. The only workaround is to materialize columns of an alternative type in the underlying data source.
- **Hide the 'to' column on relationships.** The `to` column on relationships is commonly the primary key on the `to` table. That column should be hidden, but if hidden, it doesn't appear in the field list and can't be used in visuals. Often the columns on which relationships are based are actually *system columns*, for example surrogate keys in a data warehouse. It's still best to hide such columns.

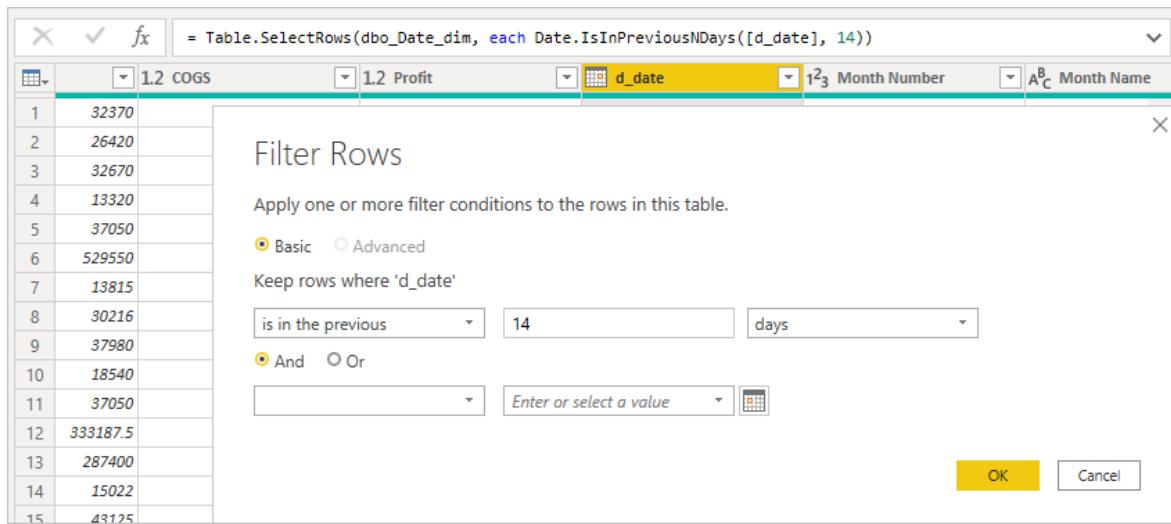
If the column has meaning, introduce a calculated column that's visible and that has a simple expression of being equal to the primary key, for example:

SQL

```
ProductKey_PK  (Destination of a relationship, hidden)
ProductKey (= [ProductKey_PK], visible)
ProductName
...
```

- **Examine all calculated columns and data type changes.** You can use calculated tables when you use DirectQuery with [composite models](#). These capabilities aren't necessarily harmful, but they result in queries that contain expressions rather than simple references to columns. Those queries might result in indexes not being used.
- **Avoid bidirectional cross filtering on relationships.** Using bidirectional cross filtering can lead to query statements that don't perform well. For more information about bidirectional cross filtering, see [Enable bidirectional cross-filtering for DirectQuery in Power BI Desktop](#), or download the [Bidirectional cross-filtering](#) white paper. The examples in the paper are for SQL Server Analysis Services, but the fundamental points also apply to Power BI.
- **Experiment with setting *Assume referential integrity*.** The *Assume referential integrity* setting on relationships enables queries to use `INNER JOIN` rather than `OUTER JOIN` statements. This guidance generally improves query performance, although it depends on the specifics of the data source.

- **Don't use the relative date filtering in Power Query Editor.** It's possible to define relative date filtering in Power Query Editor. For example, you can filter to the rows where the date is in the last 14 days.



However, this filter translates into a filter based on a fixed date, such as the time the query was authored, as you can see in the native query.

```
select [__].[d_date_sk], [__].[d_date_id], [__].[d_date], ... [__].[d_current_year]
from [dbo].[Date_dim] as [__]
where convert(date, [__].[d_date]) >= convert(datetime2, '2017-03-15 00:00:00') and
convert(date, [__].[d_date]) < convert(datetime2, '2017-03-29 00:00:00')
```

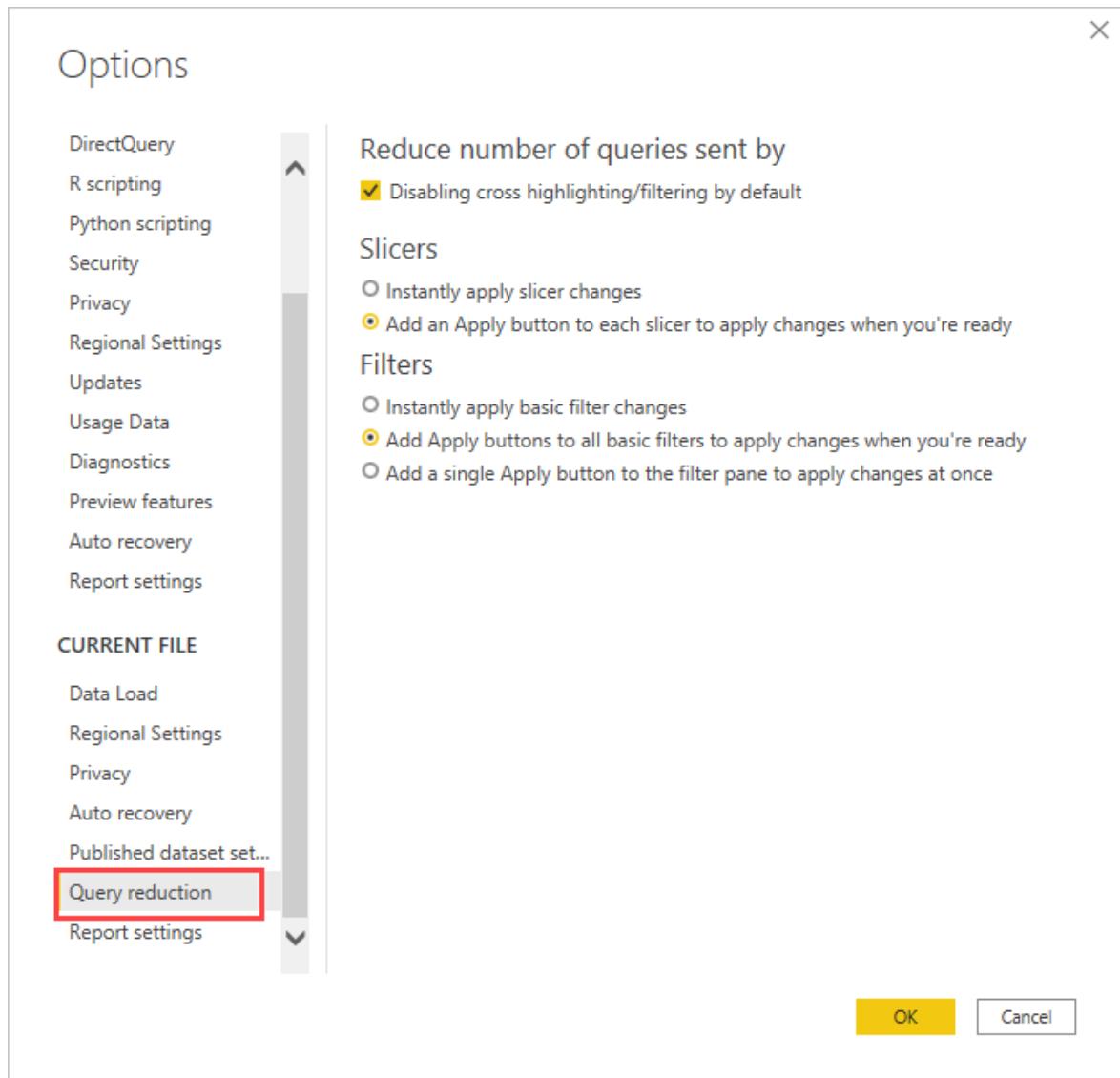
This data is probably not what you want. To ensure the filter is applied based on the date at the time the report runs, apply the date filter in the report. You can create a calculated column that calculates the number of days ago by using the `DAX DATE()` function, and use that calculated column in the filter.

Report design

When you create a report that uses a DirectQuery connection, follow this guidance:

- **Consider using query reduction options:** Power BI provides report options to send fewer queries, and to disable certain interactions that cause a poor experience if the resulting queries take a long time to run. These options apply when you interact with your report in Power BI Desktop, and also apply when users consume the report in the Power BI service.

To access these options in Power BI Desktop, go to **File > Options and settings > Options** and select **Query reduction**.



Selections on the **Query reduction** screen let you show an **Apply** button for slicers or filter selections. No queries are sent until you select the **Apply** button on the filter or slicer. The queries then use your selections to filter the data. This button lets you make several slicer and filter selections before you apply them.

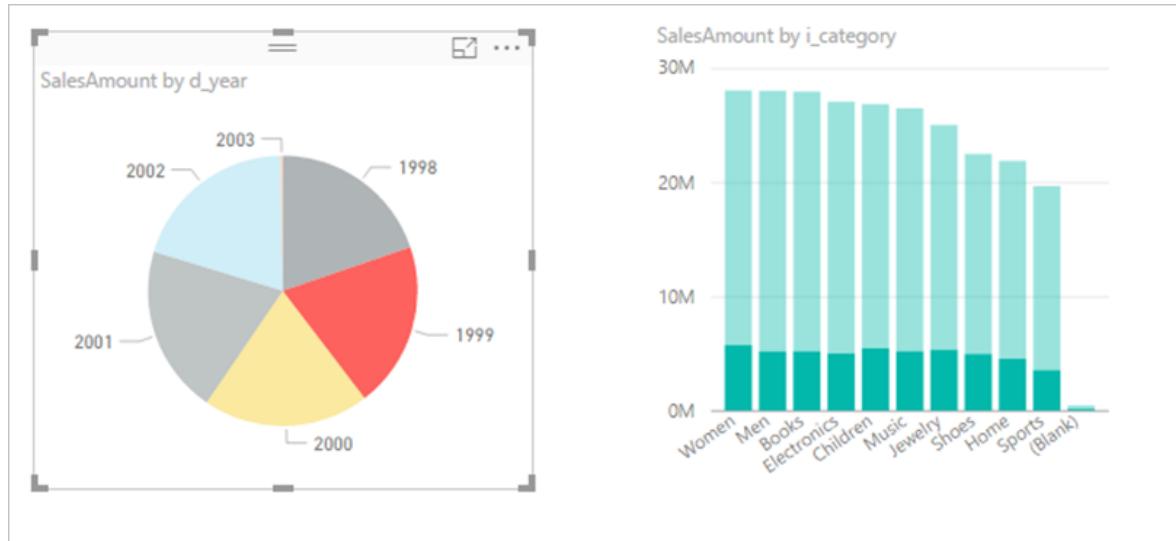
- **Apply filters first:** Always apply any applicable filters at the start of building a visual. For example, rather than drag in **TotalSalesAmount** and **ProductName**, and then filter to a particular year, apply the filter on **Year** at the beginning.

Each step of building a visual sends a query. Although it's possible to make another change before the first query completes, this approach still leaves unnecessary load on the underlying source. Applying filters early generally makes those intermediate queries less costly. Failing to apply filters early can result in hitting the one-million row limit.

- **Limit the number of visuals on a page:** When you open a page or change a page level slicer or filter, all the visuals on the page refresh. There's a limit on the number of parallel queries. As the number of visuals increases, some visuals refresh serially, which increases the time it takes to refresh the page. Therefore, it's best to

limit the number of visuals on a single page, and instead have more, simpler pages.

- **Consider switching off interaction between visuals:** By default, visualizations on a report page can be used to cross-filter and cross-highlight the other visualizations on the page. For example, if you select **1999** on the pie chart, the column chart is cross-highlighted to show the sales by category for **1999**.



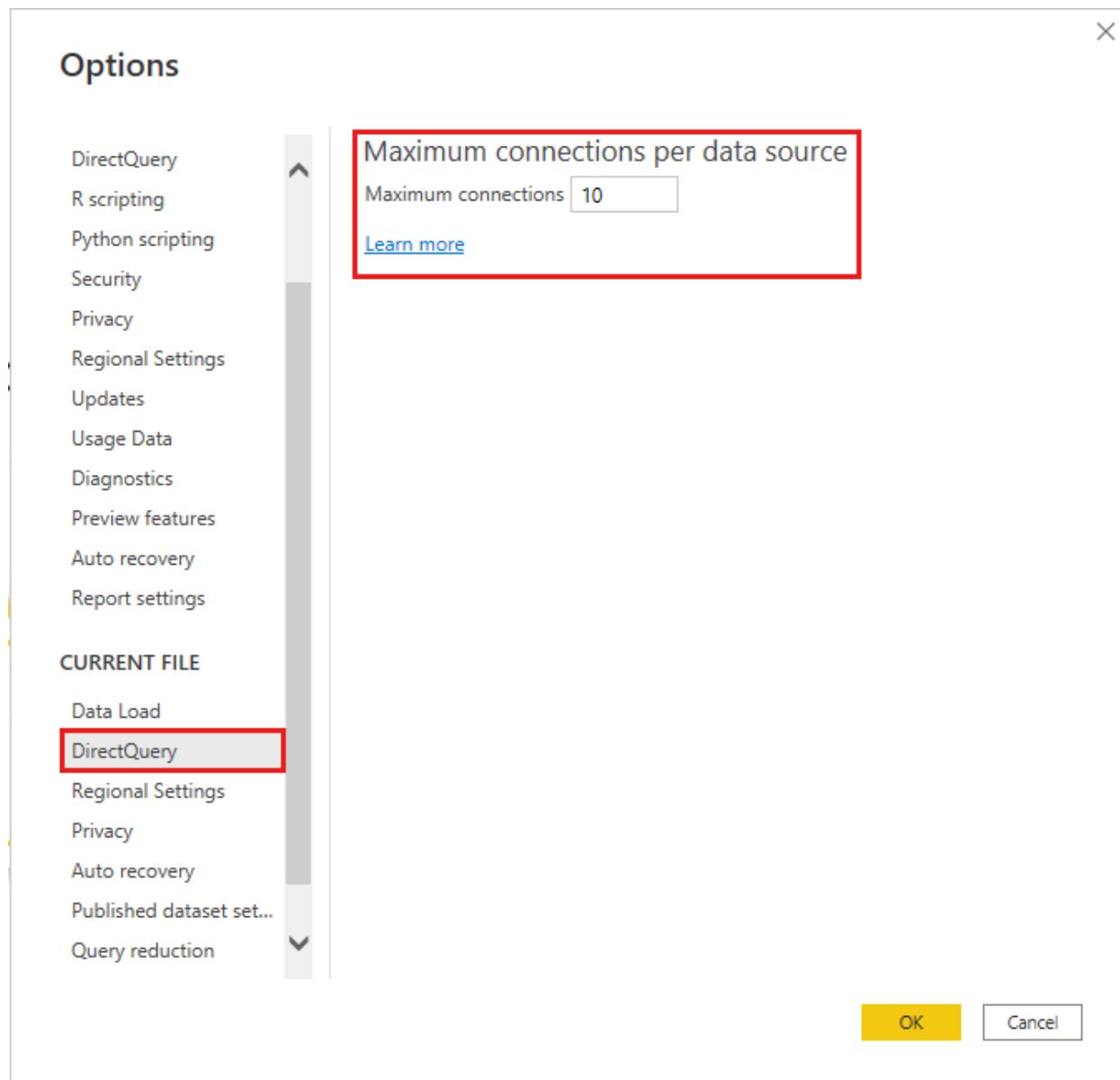
Cross-filtering and cross-highlighting in DirectQuery require queries to be submitted to the underlying source. You should switch off this interaction if the time taken to respond to users' selections is unreasonably long.

You can use the **Query reduction** settings to disable cross-highlighting throughout your report, or on a case-by-case basis. For more information, see [How visuals cross-filter each other in a Power BI report](#).

Maximum number of connections

You can set the maximum number of connections DirectQuery opens for each underlying data source, which controls the number of queries concurrently sent to each data source.

DirectQuery opens a default maximum number of 10 concurrent connections. To change the maximum number for the current file in Power BI Desktop, go to **File > Options and Settings > Options**, and select **DirectQuery** in the **Current File** section of the left pane.



The setting is enabled only when there's at least one DirectQuery source in the current report. The value applies to all DirectQuery sources, and to any new DirectQuery sources added to that report.

Increasing **Maximum connections per data source** allows sending more queries, up to the maximum number specified, to the underlying data source. This approach is useful when many 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. A higher limit results in more load on the underlying source, so the setting isn't guaranteed to improve overall performance.

Once you publish a report to the Power BI service, the maximum number of concurrent queries also depends on fixed limits set on the target environment where the report is published. Power BI, Power BI Premium, and Power BI Report Server impose different limits. The table below lists the upper limits of the active connections per data source for each Power BI environment. These limits apply to cloud data sources and on-premises data sources such as SQL Server, Oracle, and Teradata.

Environment	Upper limit per data source
Power BI Pro	10 active connections
Power BI Premium	30 active connections
Power BI Report Server	10 active connections

 **Note**

The maximum number of DirectQuery connections setting applies to all DirectQuery sources when you enable **enhanced metadata**, which is the default setting for all models created in Power BI Desktop.

DirectQuery in the Power BI service

All DirectQuery data sources are supported from Power BI Desktop, and some sources are also available directly from within the Power BI service. A business user can use Power BI to connect to their data in Salesforce, for example, and immediately get a dashboard, without using Power BI Desktop.

Only the following two DirectQuery-enabled sources are available directly in the Power BI service:

- Spark
- Azure Synapse Analytics (formerly SQL Data Warehouse)

Even for these two sources, it's still best to start DirectQuery use within Power BI Desktop. While it's easy to initially make the connection in the Power BI service, there are limitations on further enhancing the resulting report. For example, in the service it's not possible to create any calculations, or use many analytical features, or refresh the metadata to reflect changes to the underlying schema.

The performance of a DirectQuery report in the Power BI service depends on the degree of load placed on the underlying data source. The load depends on:

- The number of users that share the report and dashboard.
- The complexity of the report.
- Whether the report defines row-level security.

Report behavior in the Power BI service

When you open a report in the Power BI service, all the visuals on the currently visible page refresh. Each visual requires at least one query to the underlying data source. Some visuals might require more than one query. For example, a visual might show aggregate values from two different fact tables, or contain a more complex measure, or contain totals of a non-additive measure like **Count Distinct**. Moving to a new page refreshes those visuals. Refreshing sends a new set of queries to the underlying source.

Every user interaction on the report might result in visuals being refreshed. For example, selecting a different value on a slicer requires sending a new set of queries to refresh all of the affected visuals. The same is true for selecting a visual to cross-highlight other visuals, or changing a filter. Similarly, creating or editing a report requires queries to be sent for each step on the path to produce the final visual.

There's some caching of results. The refresh of a visual is instantaneous if the exact same results were recently obtained. If row-level security is defined, these caches aren't shared across users.

Using DirectQuery imposes some important limitations in some of the capabilities the Power BI service offers for published reports:

- **Quick insights aren't supported:** Power BI quick insights search different subsets of your dataset while applying a set of sophisticated algorithms to discover potentially interesting insights. Because quick insights require high-performance queries, this feature isn't available on datasets that use DirectQuery.
- **Using Explore in Excel results in poor performance:** You can explore a dataset by using the **Explore in Excel** capability, which lets you create pivot tables and pivot charts in Excel. This capability is supported for datasets that use DirectQuery, but performance is slower than creating visuals in Power BI. If using Excel is important for your scenarios, account for this issue in deciding whether to use DirectQuery.
- **Excel doesn't show hierarchies:** For example, when you use [Analyze in Excel](#), Excel doesn't show any hierarchies defined in Azure Analysis Services models or Power BI datasets that use DirectQuery.

Dashboard refresh

In the Power BI service, you can pin individual visuals or entire pages to dashboards as tiles. Tiles that are based on DirectQuery datasets refresh automatically by sending queries to the underlying data sources on a schedule. By default, datasets refresh every hour, but you can configure refresh between weekly and every 15 minutes as part of dataset settings.

If no row-level security is defined in the model, each tile is refreshed once, and the results are shared across all users. If you use row-level security, each tile requires separate queries per user to be sent to the underlying source.

There can be a large multiplier effect. A dashboard with 10 tiles, shared with 100 users, created on a dataset using DirectQuery with row-level security, results in at least 1000 queries being sent to the underlying data source for every refresh. Give careful consideration to the use of row-level security and the configuration of the refresh schedule.

Query timeouts

A timeout of four minutes applies to individual queries in the Power BI service. Queries that take longer than four minutes fail. This limit is intended to prevent issues caused by overly long execution times. You should use DirectQuery only for sources that can provide interactive query performance.

Performance diagnostics

This section describes how to diagnose performance issues, or how to get more detailed information to optimize your reports.

Start diagnosing performance issues in Power BI Desktop, rather than in the Power BI service. Performance issues are often based on the performance of the underlying source. You can more easily identify and diagnose issues in the more isolated Power BI Desktop environment.

This approach initially eliminates certain components, such as the Power BI gateway. If the performance issues don't occur in Power BI Desktop, you can investigate the specifics of the report in the Power BI service.

The Power BI Desktop [Performance analyzer](#) is a useful tool for identifying issues. Try to isolate any issues to one visual, rather than many visuals on a page. If a single visual on a Power BI Desktop page is sluggish, use the [Performance analyzer](#) to analyze the queries that Power BI Desktop sends to the underlying source.

You can also view traces and diagnostic information that some underlying data sources emit. Even if there are no traces from the source, the trace file might contain useful details of how a query runs and how you can improve it. You can use the following process to view the queries Power BI sends and their execution times.

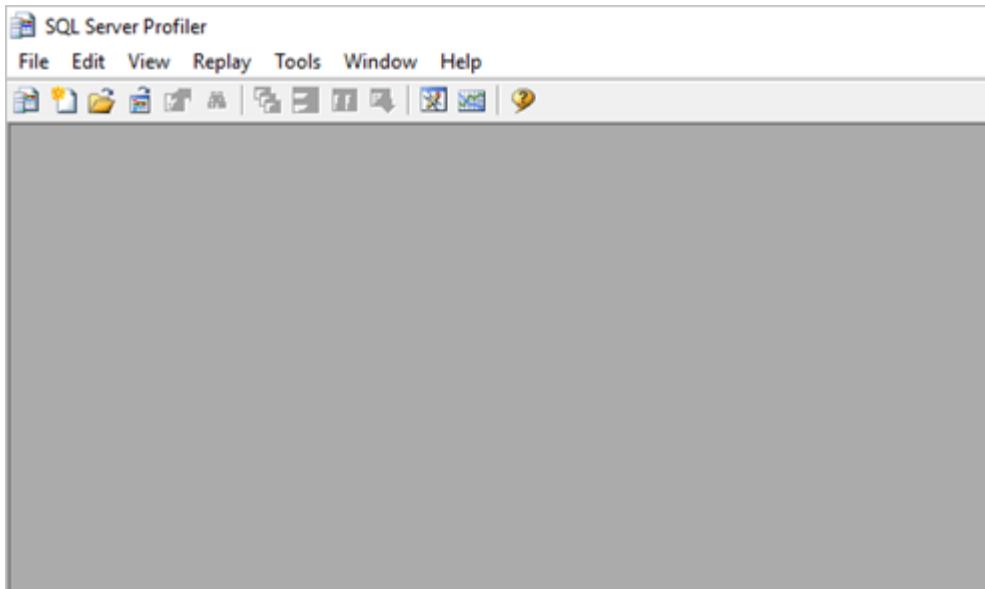
Use SQL Server Profiler to see queries

By default, Power BI Desktop logs events during a given session to a trace file called *FlightRecorderCurrent.trc*. The trace file is in the Power BI Desktop folder for the current user, in a folder called *AnalysisServicesWorkspaces*.

For some DirectQuery sources, this trace file includes all queries sent to the underlying data source. The following data sources send queries to the log:

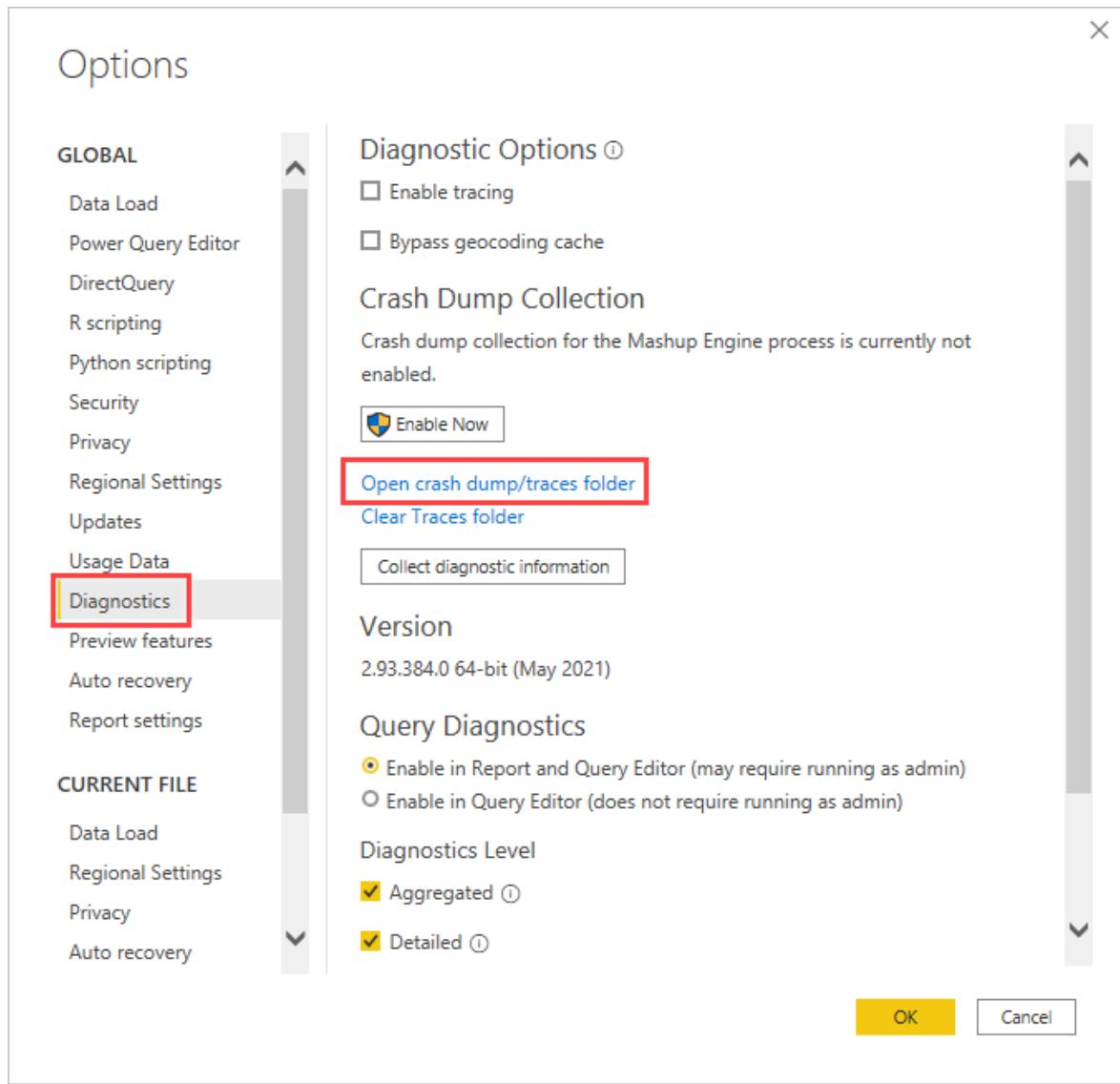
- SQL Server
- Azure SQL Database
- Azure Synapse Analytics (formerly SQL Data Warehouse)
- Oracle
- Teradata
- SAP HANA

You can read the trace files by using the *SQL Server Profiler*, part of the free download [SQL Server Management Studio](#).



To open the trace file for the current session:

1. During a Power BI Desktop session, select **File > Options and settings > Options**, and then select **Diagnostics**.
2. Under Crash Dump Collection, select **Open crash dump/traces folder**.



The *Power BI Desktop\Traces* folder opens.

3. Navigate to the parent folder and then to the *AnalysisServicesWorkspaces* folder, which contains one workspace folder for every open instance of Power BI Desktop. These folders are named with an integer suffix, such as *AnalysisServicesWorkspace2058279583*. The workspace folder is deleted when the associated Power BI Desktop session ends.

Inside the workspace folder for the current Power BI session, the *\Data* folder contains the *FlightRecorderCurrent.trc* trace file. Make a note of the location.

4. Open SQL Server Profiler, and select **File > Open > Trace File**.
5. Navigate to or enter the path to the trace file for the current Power BI session, and open *FlightRecorderCurrent.trc*.

SQL Server Profiler displays all events from the current session. The following screenshot highlights a group of events for a query. Each query group has the following events:

- A **Query Begin** and **Query End** event, which represent the start and end of a DAX query generated by changing a visual or filter in the Power BI UI, or from filtering or transforming data in the Power Query Editor.
- One or more pairs of **DirectQuery Begin** and **DirectQuery End** events, which represent queries sent to the underlying data source as part of evaluating the DAX query.

SQL Server Profiler: [C:\...\Data\FlightRecordsCurrent.trc]

File Edit View Replay Tools Window Help

EventClass EventSubClass CurrentTime StartTime C Database NTName NTD Obj ApplicationName SessionID NTCm SP TextData ServerName R RequestP ActivityID RequestID EndTime Duration C L S Error E

Command End	12 - Batch	2022-11-17 ...	6 3b8... RED... R... 9824 Power BI... ED11... RED... 14 <batch Transaction="T..." LAPTOP... <Prop... F54FEE88-B-E4C1-4790... CCC1A... 2022-11-17 17:14:3... 5 0 13 1
Command Begin	12 - Batch	2022-11-17 ...	81 3b8... RED... R... 9824 Power BI... A0CF... RED... 301 <batch Transaction="T..." LAPTOP... <Prop... 793BF0D8-B267-45B6... 30A9D... 2022-11-17 17:14:3... 4 0 13 1
Command End	12 - Batch	2022-11-17 ...	6 3b8... RED... R... 9824 Power BI... A0CF... RED... 301 <batch Transaction="T..." LAPTOP... <Prop... F94FEE88-B-E4C1-4790... 2DE9A... 2022-11-17 17:14:3... 8 0 13 1
Command Begin	12 - Batch	2022-11-17 ...	24 3b8... RED... R... 9824 Power BI... A0AC... RED... 311 <batch Transaction="T..." LAPTOP... <Prop... E27E209A-B433-4E8E... 83388... 2022-11-17 17:14:3... 5 0 13 1
Command End	12 - Batch	2022-11-17 ...	84 3b8... RED... R... 9824 Power BI... A0AC... RED... 311 <batch Transaction="T..." LAPTOP... <Prop... E27E209A-B433-4E8E... 83388... 2022-11-17 17:14:3... 5 0 13 1
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 DEFINE VAR _DSOCO... <Prop... D710944C-40CF-4420... 57905... 2022-11-17 17:14:3... 0 0 1 0
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (62) [...] LAPTOP... <Prop... D710944C-40CF-4420... 57905... 2022-11-17 17:14:3... 0 0 1 0
DirectQuery End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 DEFINE VAR _DSOCO... <Prop... D710944C-40CF-4420... 57905... 2022-11-17 17:14:3... 58 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 51 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT COUNT_BIG([...])... LAPTOP... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT COUNT_BIG([...])... LAPTOP... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 50 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 51 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 47 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 DEFINE VAR _DSOCO... <Prop... 40BDAA6-A08E-40CC... A02A7... 2022-11-17 17:14:3... 276 0 0 1
Command Begin	12 - Batch	2022-11-17 ...	88 3b8... RED... R... 9824 Power BI... 0490... RED... 328 <batch Transaction="T..." LAPTOP... <Prop... FC1D045B-5DA2-49F6... 8850F... 2022-11-17 17:14:3... 3 0 13 1
Command End	12 - Batch	2022-11-17 ...	88 3b8... RED... R... 9824 Power BI... 0490... RED... 328 <batch Transaction="T..." LAPTOP... <Prop... FC1D045B-5DA2-49F6... 8850F... 2022-11-17 17:14:3... 5 0 13 1
Command Begin	12 - Batch	2022-11-17 ...	91 3b8... RED... R... 9824 Power BI... 0490... RED... 328 <batch Transaction="T..." LAPTOP... <Prop... D053118A-C33-4E31... B0381... 2022-11-17 17:14:3... 4 0 13 1
Command End	12 - Batch	2022-11-17 ...	91 3b8... RED... R... 9824 Power BI... 0490... RED... 328 <batch Transaction="T..." LAPTOP... <Prop... D053118A-C33-4E31... B0381... 2022-11-17 17:14:3... 16 0 1 0
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... A446... RED... 318 <batch Transaction="T..." LAPTOP... <Prop... 2E53511B-21FC-407E... 0580B... 2022-11-17 17:14:3... 52 16 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... A446... RED... 318 <batch Transaction="T..." LAPTOP... <Prop... 2E53511B-21FC-407E... 0580B... 2022-11-17 17:14:3... 69 16 0 1
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT SUM([1])... [DB0... LAPTOP... <Prop... 2644A56F-626D-40E9... D056F... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT SUM([1])... [DB0... LAPTOP... <Prop... 2644A56F-626D-40E9... D056F... 0 0 1 0
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT SUM([1])... [DB0... LAPTOP... <Prop... 2644A56F-626D-40E9... D056F... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 2644A56F-626D-40E9... D056F... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 2644A56F-626D-40E9... D056F... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 DEFINE VAR _DSOCO... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:14:3... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT SUM([1])... [DB0... LAPTOP... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:14:3... 55 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 499600C-0154-49B8... 7430B... 0 0 1 0
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001)... LAPTOP... <Prop... 499600C-0154-49B8... 7430B... 49 0 0 1
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 DEFINE VAR _DSOCO... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:14:3... 116 0 0 1

```

DEFINE _DSOCORE =
    SUMMARIZE(
        ProductCategory,
        'SalesLT.ProductCategory'[ModifiedDate],
        'SalesLT.ProductCategory'[Name]
    )
VAR __DSOPRIMARYBASE = 
    SUMMARIZE(_DSOCORE, 'SalesLT.ProductCategory'[ModifiedDate])

```

Done.

Ln 137, Col 1 | Rows: 229 | Connections: 0

Multiple DAX queries can run in parallel, so events from different groups can be interleaved. You can use the **ActivityID** value to determine which events belong to the same group.

The following columns are also of interest:

- **TextData:** The textual detail of the event. For **Query Begin** and **Query End** events, the detail is the DAX query. For **DirectQuery Begin** and **DirectQuery End** events, the detail is the SQL query sent to the underlying source. The **TextData** for the currently selected event also appears in the pane at the bottom of the screen.
- **EndTime:** The time when the event completed.
- **Duration:** The duration, in milliseconds, it took to run the DAX or SQL query.
- **Error:** Whether an error occurred, in which case the event also displays in red.

To capture a trace to help diagnose a potential performance issue:

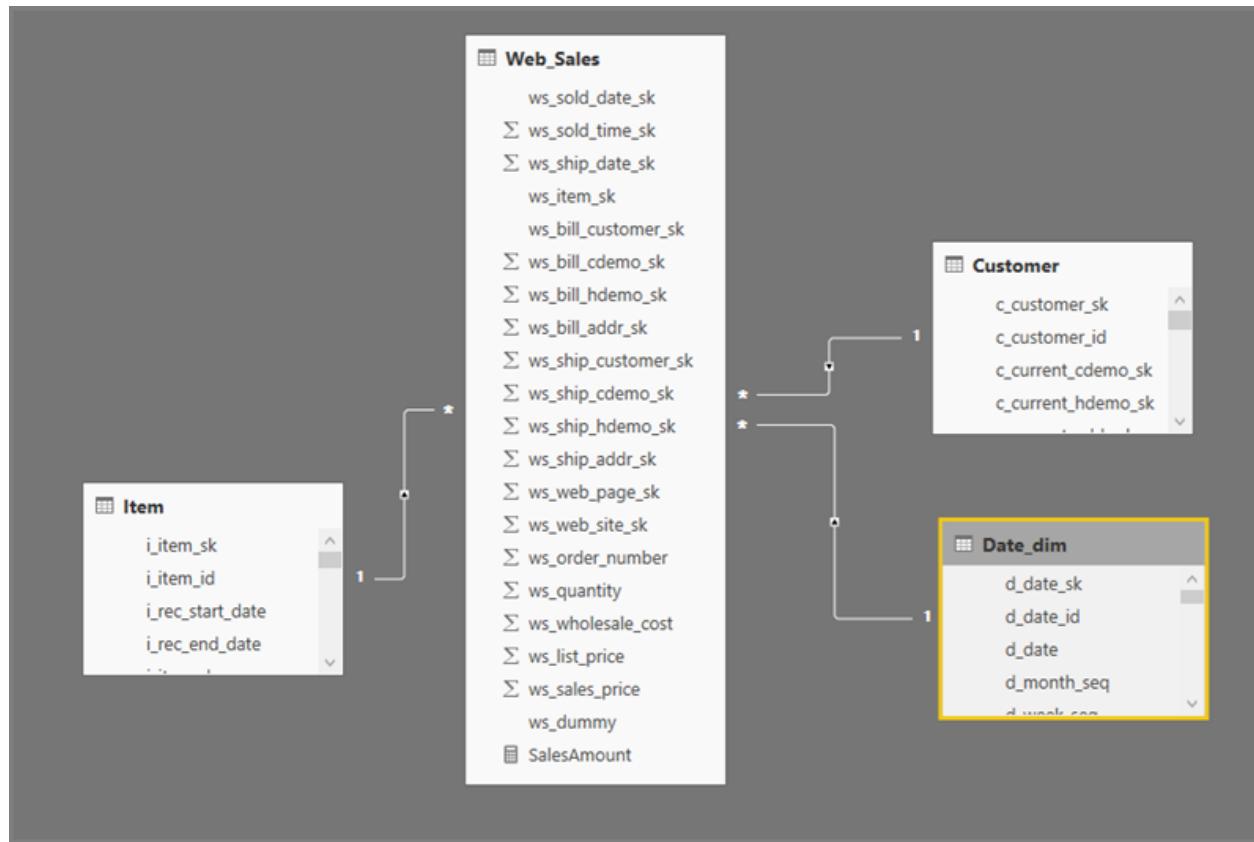
1. Open a single Power BI Desktop session, to avoid the confusion of multiple workspace folders.
2. Do the set of actions of interest in Power BI Desktop. Include a few more actions, to ensure that the events of interest are flushed into the trace file.

3. Open SQL Server Profiler and examine the trace. Remember that closing Power BI Desktop deletes the trace file. Also, further actions in Power BI Desktop don't immediately appear. You must close and reopen the trace file to see new events.

Keep individual sessions reasonably small, perhaps 10 seconds of actions, not hundreds. This approach makes it easier to interpret the trace file. There's also a limit on the size of the trace file. For long sessions, there's a chance of early events being dropped.

Understand the format of queries

The general format of Power BI Desktop queries uses subselects for each table they reference. The Power Query Editor query defines the subselect queries. For example, assume you have the following [TPC-DS](#) tables in SQL Server:

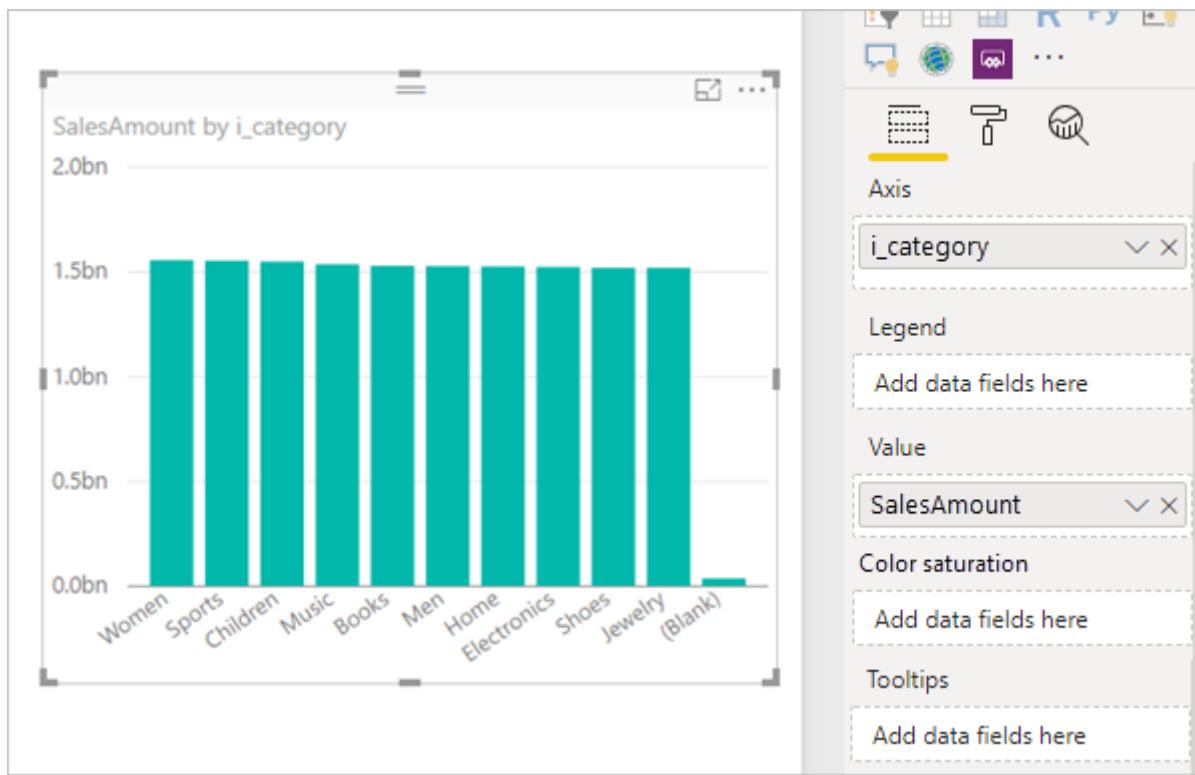


Running the following query:

```
SQL

SalesAmount (SUMX(Web_Sales, [ws_sales_price]*[ws_quantity]))
by Item[i_category]
for Date_dim[d_year] = 2000
```

Results in the following visual in Power BI:



Refreshing that visual produces the SQL query in the following image. There are three subselect queries for `Web_Sales`, `Item`, and `Date_dim`, which each return all the columns on the respective table, even though the visual references only four columns.

```

SELECT
TOP (1000001) [c13],SUM([a0])
AS [a0]
FROM
(
SELECT [t0].[i_category] AS [c13],[t1].[ws_quantity] AS [c40],[t1].[ws_sales_price] AS
[c43],[t3].[d_year] AS [c72],
([t1].[ws_sales_price] * [t1].[ws_quantity])
AS [a0]
FROM
(
(select [$Table].[ws_sold_date_sk] as [ws_sold_date_sk],
[$Table].[ws_sold_time_sk] as [ws_sold_time_sk],
[$Table].[ws_ship_date_sk] as [ws_ship_date_sk],
[$Table].[ws_item_sk] as [ws_item_sk],
[$Table].[ws_bill_customer_sk] as [ws_bill_customer_sk],
[$Table].[ws_bill_cdemo_sk] as [ws_bill_cdemo_sk],
[$Table].[ws_bill_hdemo_sk] as [ws_bill_hdemo_sk],
... 8 other columns
[$Table].[ws_quantity] as [ws_quantity],
[$Table].[ws_wholesale_cost] as [ws_wholesale_cost],
[$Table].[ws_list_price] as [ws_list_price],
[$Table].[ws_sales_price] as [ws_sales_price],
[$Table].[ws_dummy] as [ws_dummy]
from [dbo].[Web_Sales] as [$Table]) AS [t1]
left outer join

(select [$Table].[i_item_sk] as [i_item_sk],
[$Table].[i_item_id] as [i_item_id],
[$Table].[i_rec_start_date] as [i_rec_start_date],
[$Table].[i_rec_end_date] as [i_rec_end_date],

```

```

[$Table].[i_item_desc] as [i_item_desc],
[$Table].[i_current_price] as [i_current_price],
[$Table].[i_wholesale_cost] as [i_wholesale_cost],
[$Table].[i_brand_id] as [i_brand_id],
[$Table].[i_brand] as [i_brand],
[$Table].[i_class_id] as [i_class_id],
[$Table].[i_class] as [i_class],
[$Table].[i_category_id] as [i_category_id],
[$Table].[i_category] as [i_category],
... 8 other columns
[$Table].[i_product_name] as [i_product_name],
[$Table].[i_dummy] as [i_dummy]
from [dbo].[Item] as [$Table]) AS [t0] on
(
[t1].[ws_item_sk] = [t0].[i_item_sk]
)
)

left outer join
(select [$Table].[d_date_sk] as [d_date_sk],
[$Table].[d_date_id] as [d_date_id],
[$Table].[d_date] as [d_date],
[$Table].[d_month_seq] as [d_month_seq],
[$Table].[d_week_seq] as [d_week_seq],
[$Table].[d_quarter_seq] as [d_quarter_seq],
[$Table].[d_year] as [d_year],
[$Table].[d_dow] as [d_dow],
[$Table].[d_moy] as [d_moy],
[$Table].[d_dom] as [d_dom],
[$Table].[d_qoy] as [d_qoy],
... 16 other columns
[$Table].[d_current_year] as [d_current_year],
[$Table].[d_dummy] as [d_dummy]
from [dbo].[Date_dim] as [$Table]) AS [t3] on
(
[t1].[ws_sold_date_sk] = [t3].[d_date_sk]
)
)
AS [t0]
WHERE
(
[c72] = 2000
)
GROUP BY [c13]

```

Power Query Editor defines the exact subselect queries. This use of subselect queries hasn't been shown to affect performance for the data sources DirectQuery supports. Data sources like SQL Server optimize away the references to the other columns.

Power BI uses this pattern because the analyst provides the SQL query directly. Power BI uses the query as provided, without any attempt to rewrite it.

Next steps

For more information about DirectQuery in Power BI, see:

- [Use DataQuery in Power BI Desktop](#)

This article described aspects of DirectQuery that are common across all data sources.

See the following articles for details about specific sources:

- [DirectQuery and SAP HANA](#)
- [DirectQuery and SAP BW](#)
- [Use DirectQuery for Power BI datasets and Analysis Services](#)

Live connection and DirectQuery comparison

Article • 12/12/2022

[Live connection](#) is a way of connecting a Power BI report to a published Power BI dataset. [DirectQuery](#) is a method you can use to connect your dataset to data. This article describes the main differences between these concepts.

Live connection

Live connection is a method that lets you build a report in Power BI Desktop without having to build a dataset for it. When you create your report in Power BI Desktop, you can connect it to a dataset that already exists. A *live connection* allows you to rely on existing data, which can be updated without accessing the report.

Using *live connection* you can connect your report to one of the following data sources:

- A dataset that already exists in Power BI service
- An Azure Analysis Services (AAS) database
- An on-premises instance of SQL Server Analysis Services (SSAS)

DirectQuery

A Power BI dataset can have data copied into it during a refresh operation, in what's called [import mode](#). Or, the dataset can dynamically request data from a data source it's connected to using a method called [DirectQuery](#).

When using *DirectQuery*, your report uses Data Analysis Expression (DAX) queries to get data. After the dataset receives the report's DAX query, it generates another set of queries that are run on your data source, to get the required data. If for example your data source is an SQL Server database, Power BI will generate SQL queries to get the data it needs. Other data sources may generate queries in other query languages.

DirectQuery is useful when:

- You're working against data sources with a large volume of data
- You want to use 'near real-time' data

You can also use *DirectQuery* with Analysis Services, as described in [Using DirectQuery for Power BI datasets and Analysis Services](#).

Next steps

[Dataset modes in the Power BI service](#)

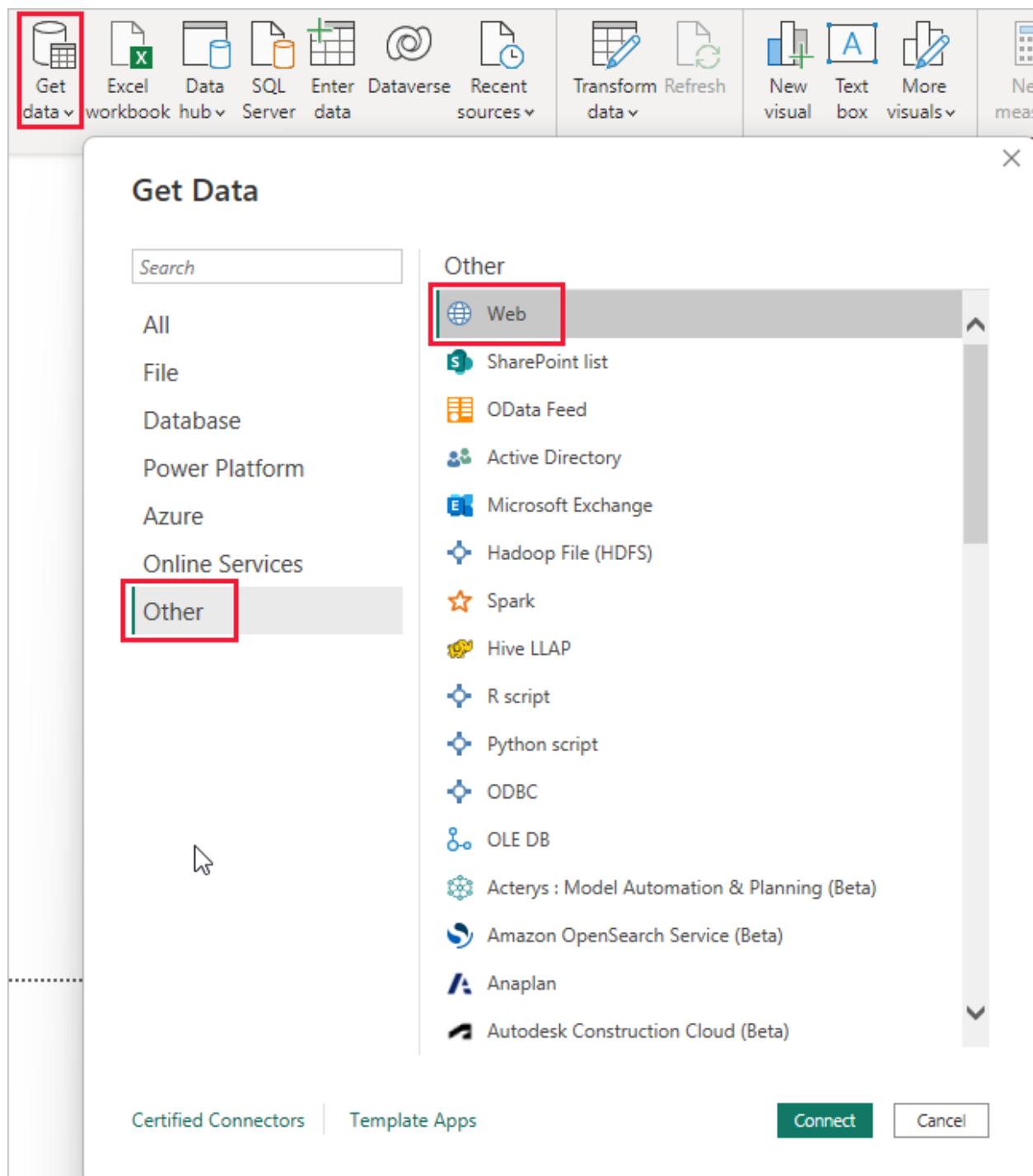
[Connect to datasets in the Power BI service from Power BI Desktop](#)

Connect to data sources in Power BI Desktop

Article • 01/12/2023

With Power BI Desktop, you can easily connect to the ever expanding world of data. If you don't have Power BI Desktop, you can [download](#) and install it.

There are *all sorts* of data sources available in Power BI Desktop. The following image shows how to connect to data, by selecting **Get data > Other > Web**.



Example of connecting to data

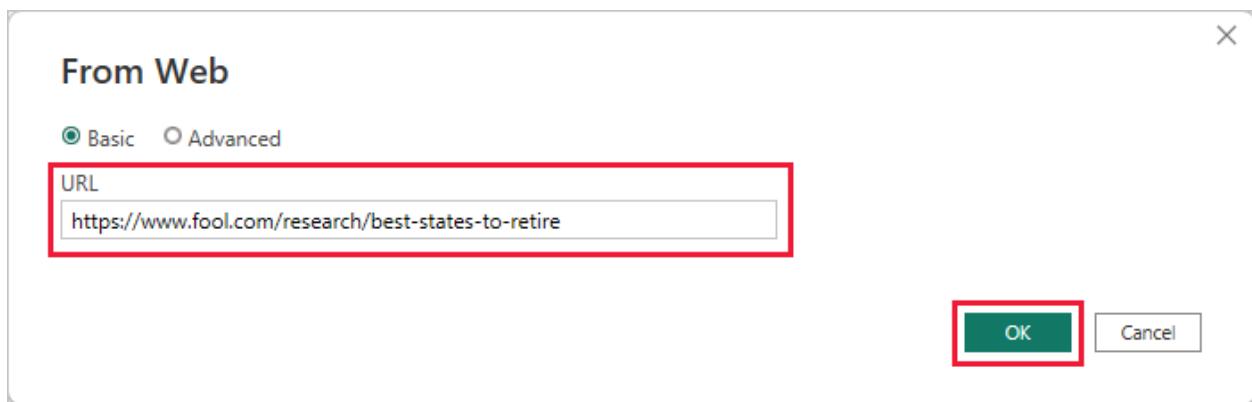
For this example, we'll connect to a **Web** data source.

Imagine you're retiring. You want to live where there's lots of sunshine, preferable taxes, and good health care. Or... perhaps you're a data analyst, and you want that information to help your customers, as in, help your raincoat manufacturing client target sales where it rains a *lot*.

Either way, you find a Web resource that has interesting data about those topics, and more:

<https://www.fool.com/research/best-states-to-retire> ↗

Select **Get data > Other > Web**. In **From Web**, enter the address.



When you select **OK**, the **Query** functionality of Power BI Desktop goes to work. Power BI Desktop contacts the Web resource, and the **Navigator** window returns the results of what it found on that Web page. In this case, it found a table. We're interested in that table, so we select it from the list. The **Navigator** window displays a preview.

The screenshot shows the Microsoft Power BI Navigator window. On the left, there's a sidebar titled "Navigator" with a search bar and a "Display Options" dropdown. Below it, there are three main sections: "HTML Tables [9]", "Suggested Tables [3]", and "Text [2]". Under "HTML Tables", "Table 1" is selected and highlighted with a red box. The main area displays a table titled "Table 1" with columns: State, Overall ranking, Quality of life, Housing cost, and Health care. The table lists 21 US states from Maine at rank 1 to Indiana at rank 21. At the bottom of the window, there are three buttons: "Load", "Transform Data" (which is also highlighted with a red box), and "Cancel".

State	Overall ranking	Quality of life	Housing cost	Health care
Maine	1	77.8	57.4	58
Vermont	2	70.6	58	49.4
New Hampshire	3	59.4	75.3	75.3
Kentucky	4	59.3	82.2	67.6
West Virginia	5	64.3	70.6	48.9
Iowa	6	50.2	75.7	55.2
Wisconsin	7	44.2	70.6	57.5
Nebraska	8	42.1	70.6	70.5
Rhode Island	9	61.3	70.6	70.5
Wyoming	10	47.7	61.6	57.5
Oregon	11	59.4	38.6	55.2
Virginia	12	36.4	78.9	57
Ohio	13	46.4	73.8	70.5
Pennsylvania	14	52.7	67.2	57.5
Delaware	15	32.3	70.6	70.5
Michigan	16	49.2	70.6	70.5
Mississippi	17	71.6	78.9	57
Montana	18	57.5	48.9	57
Connecticut	19	41.5	70.6	70.5
North Carolina	20	51.6	63.7	70.5
Indiana	21	50.6	72.9	57

At this point, you can edit the query before loading the table, by selecting **Transform Data** from the bottom of the window, or just load the table.

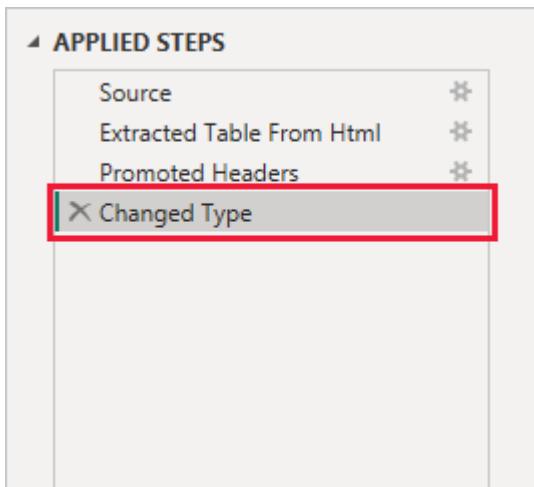
Select **Transform Data** to load the table and launch Power Query Editor. The **Query Settings** pane is displayed. If it's not, select **View** from the ribbon, then choose **Query Settings** to display the **Query Settings** pane. Here's what the editor looks like.

The screenshot shows the Power Query Editor interface. The 'Query Settings' pane is open on the right, with the 'APPLIED STEPS' section highlighted by a red box. The steps listed are: Source, Extracted Table From Html, Promoted Headers, and Changed Type. The 'Changed Type' step is currently selected. The main area displays a table with columns: State, Overall ranking, Quality of life, and Housing cost. The 'Quality of life' column contains text values like '77.8', '70.6', etc., which are being converted to numbers.

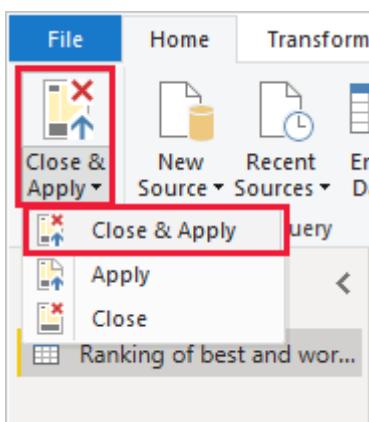
All those scores are text rather than numbers, and we need them to be numbers. No problem. Just right-click the column header, and select **Change Type > Whole Number** to change them. To choose more than one column, first select a column then choose **Shift**, select other adjacent columns, and then right-click a column header to change all selected columns. Use **Ctrl** to choose columns that aren't adjacent.

The screenshot shows the Power Query Editor with a context menu open over the 'Quality of life' column header. The menu options include Copy, Remove, Remove Other Columns, Duplicate Column, Add Column From Examples..., Remove Duplicates, Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., Split Column, Group By..., Fill, Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, Rename..., and Text. The 'Change Type' option is highlighted with a red box, and its submenu is open, showing options like Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone, Duration, Text, and True/False. The 'Whole Number' option is also highlighted with a red box.

In **Query Settings**, the **APPLIED STEPS** reflect any changes that were made. As you make more changes to the data, Power Query Editor records those changes in the **APPLIED STEPS** section, which you can adjust, revisit, rearrange, or delete as necessary.



Other changes to the table can still be made after it's loaded, but for now these changes are enough. When you're done, select **Close & Apply** from the **Home** ribbon, and Power BI Desktop applies the changes and closes Power Query Editor.



With the data model loaded, in **Report** view in Power BI Desktop, you can begin creating visualizations by dragging fields onto the canvas.

A screenshot of the Microsoft Power BI Desktop application in 'Report' view. On the left, there's a map of North America with several blue dots representing data points. The map includes labels for 'NORTH AMERICA', 'Pacific Ocean', and 'Atlantic Ocean'. On the right, there's a 'Fields' pane showing a list of fields from 'Table 1': Crime rate rate, Healthcare cost..., Housing cost, Non-housing c..., Overall ranking, Public health/C..., Quality of life, Sales taxes, State, Total score, and Weather. The 'State' field is checked with a green checkmark. There's also a 'Visualizations' pane with various chart and map icons.

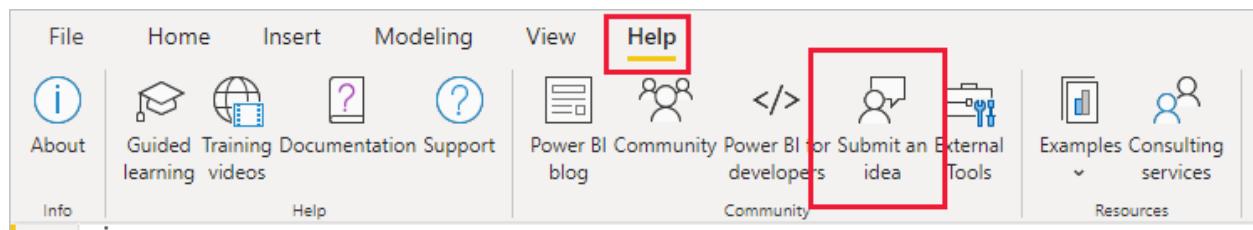
Of course, this model is simple, with a single data connection. Most Power BI Desktop reports have connections to different data sources, shaped to meet your needs, with relationships that produce a rich data model.

Next steps

There are all sorts of things you can do with Power BI Desktop. For more information on its capabilities, check out the following resources:

- [What is Power BI Desktop?](#)
- [Query overview in Power BI Desktop](#)
- [Data sources in Power BI Desktop](#)
- [Shape and combine data in Power BI Desktop](#)
- [Perform common query tasks in Power BI Desktop](#)

Want to give us feedback? Great! Use the **Submit an Idea** menu item in Power BI Desktop or visit [Community Feedback](#). We look forward to hearing from you!



Connect to cloud data sources in the Power BI service (Preview)

Article • 09/05/2023

With Power BI, you can share cloud connections for datasets and paginated reports, datamarts and dataflows, as well as Power Query Online experiences in *Get data*, enabling you to create multiple connection objects to the same cloud data source. For example, you can create separate connections to the same data source, with different credentials or privacy settings, and share the connections with others, alleviating the need for those users to manage their own separate cloud connections.

Types of data connections

The following table shows how various types of connections map to the two primary connection types: data gateway connections, and direct cloud connections. The new capability described in this article is **Shareable cloud connections**.

Data gateway connections	Direct cloud connections
Connections using a personal data gateway	Personal cloud connections
Connections using an enterprise or VNET data gateway	Shareable cloud connections (new)

Advantages of shareable cloud connections

Connections using a personal cloud connection come with several limitations. For example, with a personal cloud connection you can only create a single personal cloud connection object to a given data source. All datasets that connect to the data source use the same personal cloud connection object, so if you change the credentials of the personal cloud connection, all datasets using that personal cloud connection are affected. Often that's not a desired outcome.

Another limitation of personal cloud connection is that they can't be shared with others, so other users can't bind their datasets and paginated reports to the personal cloud connection you own; users must maintain their own personal cloud connections.

Shareable connections have no such limitations, and provide for more streamlined, more flexible connection management, including the following:

- **Support multiple connections to the same data source** - support for multiple connections on the same data source is particularly useful when you want to use different connection settings for different datasets, and other artifacts. It's also useful when you want to assign individual artifacts their own separate connections, to ensure their connection settings are isolated from each other.
- **You can share these connections with other users** - with shareable connections you can assign other users *Owner* permissions, enabling them to manage all aspects of the connection configuration, including credentials. You can provide other users with *Resharing* permissions so they can use and reshare the connection with others. You can also provide *User* permissions, enabling them to use the connection to bind their artifacts to the data source.
- **Lower the overhead of maintaining data connections and credentials** - when combined with the data source and gateway management experience, you can centralize data source connection management for gateway and cloud connections. Such centralization and management is already common for enterprise and VNET data gateways, for which a gateway administrator creates, shares, and maintains the connections. With shareable connections, you can now extend such centralized connection management to cloud data sources as well.

Compare shareable cloud connection to other connections

By default, when you create a Power BI Desktop report that connects to a cloud data source, then upload it into a workspace in the Power BI service, Power BI creates a personal cloud connection and binds it to your dataset, for which you must provide credentials. If an existing personal cloud connection is available, you likely provided the credentials previously.

In contrast, if you have access to at least one shareable cloud connection to the same data source, you can use the shareable cloud connection, which has already been configured for you by its owner, instead of having to use your only available personal cloud connection for the data source.

To use the shareable cloud connection, on the **Datasets** settings page, under **Gateway and cloud connections**, find **Cloud connections** and can select the shareable cloud connection you want to use for the connection, then select **Apply**. The following screenshot shows the settings.

General Dashboards Datasets Workbooks Dataflows

AdventureWorks

Settings for AdventureWorks

[View dataset](#)

[Refresh history](#)

▷ Dataset description

△ Gateway and cloud connections

You don't need a gateway for this dataset, because all of its data sources are in the cloud, but you can use a gateway for enhanced control over how you connect. [Learn more](#)

Gateway connections

Use an On-premises or VNet data gateway

Off

Cloud connections

Data sources included in this dataset:

SqlServer("server":"demopm.database.windows.net", "data base":"adventureworksdw2020")	Maps to:
Personal Cloud Connect	
AdventureWorksDW on DemoPM	
Personal Cloud Connection	
Create a connection	

[Apply](#) [Discard](#)

▷ Data source credentials

○ Failed to test the connection to your data source. Please retry your credentials. [Learn more](#)

AdventureWorksDW2020-demopm.database.windows.net △ [Edit credentials](#) [Show in lineage view](#)

▷ Sensitivity label

▷ Parameters

Create a new shareable cloud connection

You can create a new shareable cloud connection directly from the **Dataset** settings page. Under **Gateway connections** > **Cloud connections**, select the **Maps to** dropdown and then select **Create a connection**.

Cloud connections

Data sources included in this dataset:

SqlServer("server":"demopm.database.windows.net", "data base":"adventureworksdw2020")	Maps to:
Personal Cloud Connect	
AdventureWorksDW on DemoPM	
Personal Cloud Connection	
Create a connection	

[Apply](#) [Discard](#)

A pane appears called **New connection** and automatically populates the configuration parameters.

New connection

Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".

On-premises Virtual network Cloud

Connection name *
SSO - AdventureWorks

Connection type *
SQL Server

Server *
demopm.database.windows.net

Database *
AdventureWorksDW2020

Authentication

Authentication method *
OAuth 2.0

Single sign-on

Use SSO via Azure AD for DirectQuery queries (i)

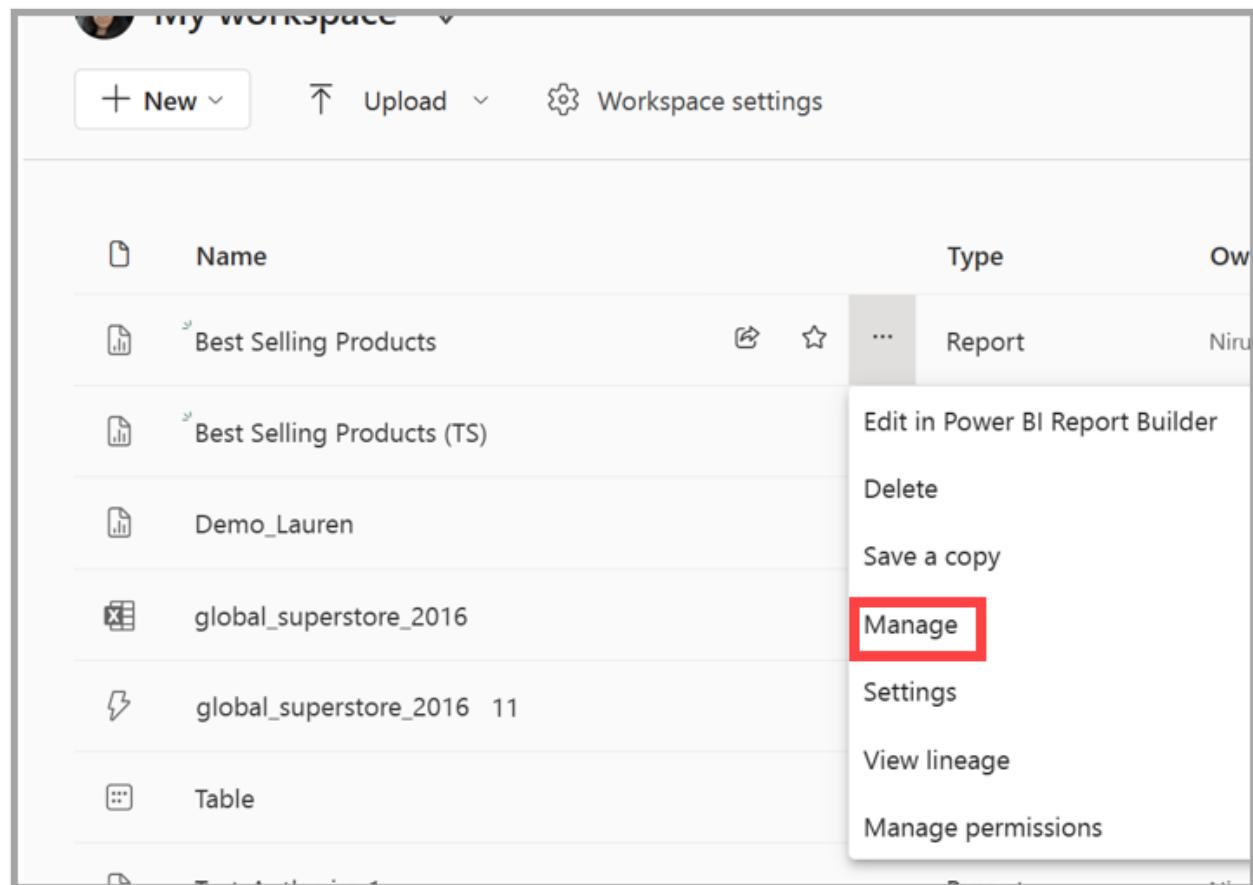
General

Create **Close**

Enabling the creation of new connections makes it easy to create separate shareable cloud connections for individual datasets, if needed. You can also display the connection management page from anywhere in the Power BI service by selecting the **Settings** gear in the upper right corner of the Power BI service, then select **Manage connections and gateways**.

Using shareable cloud connections with paginated reports

When you share your paginated report in the Power BI service, you can update the cloud connections from within the report itself. To modify the cloud connections for your paginated report, navigate to your workspace in the Power BI service, select the **More** button (ellipses) and then select **Manage**.



The screenshot shows the 'My workspace' interface in the Power BI service. At the top, there are buttons for '+ New', 'Upload', and 'Workspace settings'. Below is a list of items in the workspace:

Name	Type	Ow...
Best Selling Products	Report	Niru...
Best Selling Products (TS)		
Demo_Lauren		
global_superstore_2016		
global_superstore_2016 11		
Table		

A context menu is open over the 'global_superstore_2016' item, listing options: 'Edit in Power BI Report Builder', 'Delete', 'Save a copy', 'Manage' (which is highlighted with a red box), 'Settings', 'View lineage', and 'Manage permissions'.

Selecting **Manage** presents a page with several tabs. Select the **Reports** tab from the top row, then you can update the connection from within the **Cloud connections** area, as shown in the following screenshot.

The screenshot shows the 'Reports' tab selected in the top navigation bar. On the left, a sidebar lists datasets: 'Best Selling Products', 'Best Selling Products (TS)', 'Demo_Lauren', and 'Test_Authoring1'. The main area displays 'Settings for Best Selling Products'. Under 'Gateway and cloud connections', it says 'You don't need a gateway for this report, because all of its data sources are in the cloud, but you can use a gateway for enhanced control over how you connect.' Below this is a 'Gateway connections' section with a toggle switch set to 'Off'. The 'Cloud connections' section is highlighted with a red box. It shows a data source entry: 'SqlServer("server": "rsadventureworks.database.windows.net", "database": "rsadventureworks")' with a green checkmark. To its right is a dropdown menu titled 'Maps to:' containing 'Personal Cloud Connecti...' (selected), 'RSAAdventureWorks(TS)', 'RSAAdventureWorksCloudDS', 'Personal Cloud Connection', and 'Create a connection'. At the bottom of this section are 'Apply' and 'Discard' buttons. A link 'Data source credentials' is also present.

Limitations and considerations

- Shareable cloud connections also share your credentials - when you allow others to user your shareable cloud connections, it's important to understand that you're letting others connect their own datasets, paginated reports, and other artifacts to the corresponding data sources by using the connection details and credentials you provided. Make sure you only share connections (and their credentials) that you're authorized to share.

Next steps

For more information about creating shareable cloud connections:

[Create and share cloud data sources in the Power BI service \(Preview\)](#)

You can do all sorts of things with the Power BI service and Power BI Desktop. For more information on its capabilities, check out the following resources:

- [What is Power BI Desktop?](#)
- [Query overview with Power BI Desktop](#)
- [Data types in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)

Create and share cloud data sources in the Power BI service (Preview)

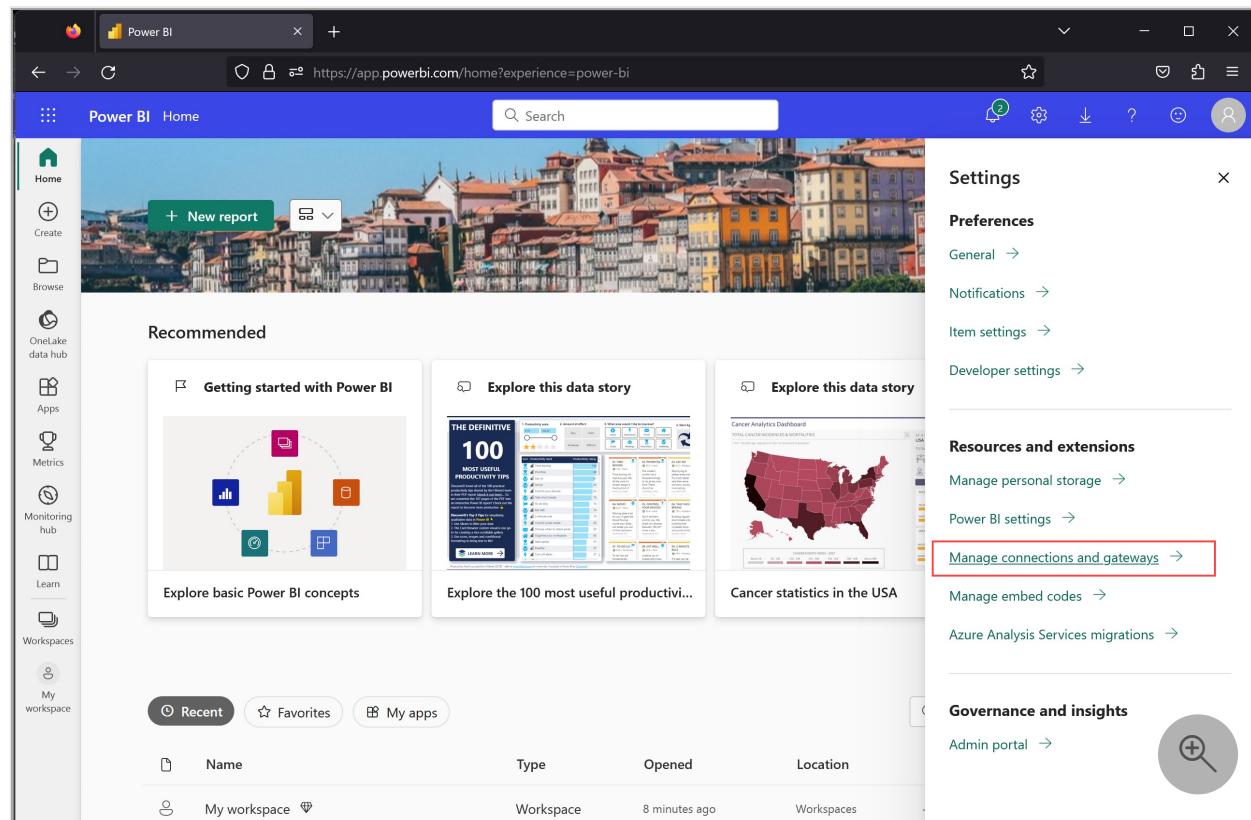
Article • 09/05/2023

With Power BI, you can create, share, and manage cloud connections for datasets and paginated reports, datamarts and dataflows, as well as Power Query Online experiences in *Get data*, all within the Power BI service user experience.

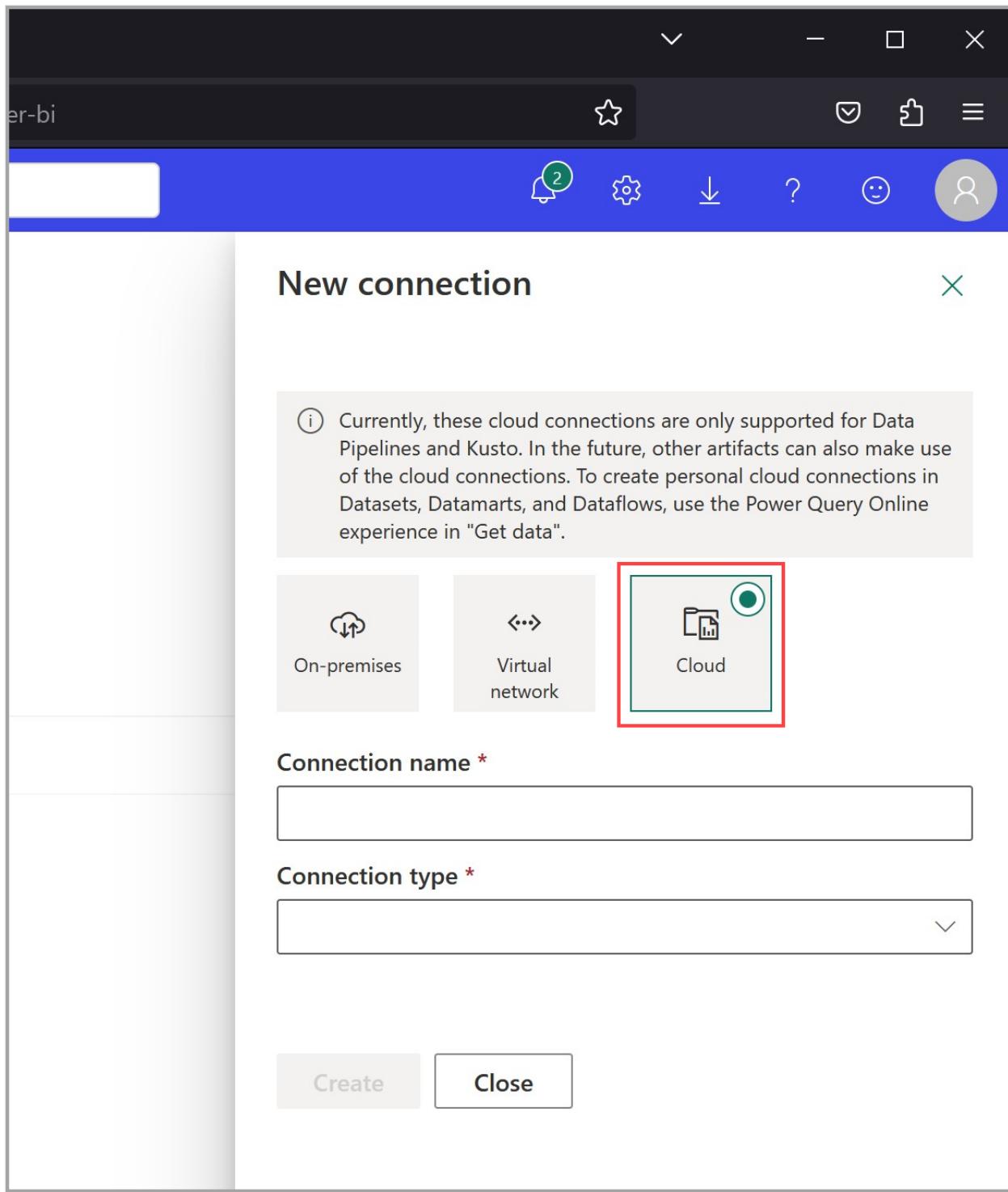
This article shows you how to create a shareable cloud connection, and then shows you how to share that connection with others. Creating and sharing shareable cloud connections have many advantages, as described in [advantages of shareable cloud connections](#).

Create a shareable cloud connection

To create a shareable cloud connection, go to the Power BI service, select the **Settings** gear icon, and from the pane that appears select **Manage connections and gateways**.



In the window that appears, select **New connection** and from the pane that appears, select **Cloud**.



Enter a name for the new connection, select the appropriate **connection type** from the drop-down list, and provide the connection details for your data source. Once you've filled in the information, select **Create**.

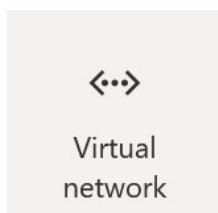
New connection

X

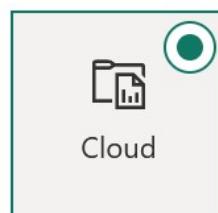
(i) Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



On-premises



Virtual
network



Cloud

Connection name *

AdventureWorksDW

Connection type *

SQL Server

Server *

base.windows.net

Database *

AdventureWorksDW20202

Authentication

Authentication method *

Basic

A screenshot of a web-based form for creating a new connection. The form has a red border and contains the following fields:

- Username ***: A text input field containing "serveradmin".
- Password ***: A password input field showing a series of black dots.
- Create**: A green button.
- Close**: A white button.

In the bottom right corner of the dialog box, there is a small circular icon with a plus sign and a magnifying glass.

With your connection created, you're ready to share it with others.

Share a shareable cloud connection

To share a shareable cloud connection that you've already created, go to your **Connections** settings in the Power BI service, select the **More** menu (the ellipses) for the connection you want to share, and select **Manage users**.

The screenshot shows the Power BI Data (preview) page. On the left, there's a vertical sidebar with icons for Home, Create, Browse, OneLake data hub, Apps, Metrics, Monitoring hub, Learn, and a recent items section. The main area has a blue header with the title "Power BI My workspace" and a search bar. Below the header, there are links for Settings, Manage users, Remove, and Get help. The main content area is titled "Data (preview)" and shows tabs for Connections, On-premises data gateways, and Virtual network data gateways. A sub-section titled "Cloud and data gateway connections for artifacts" includes a link to learn more about supported connections. A table lists connections, with one row for "AdventureWorksDW" highlighted and a context menu open over it. The context menu, which is red-bordered, contains options: "...", "SQL Server", "Settings" (which is highlighted), "Manage users", "Remove", and another "Manage users" button.

The **Manage users** window appears, where you can search users by name or by their email address, and then grant them the permission level you want them to have. You must at least grant *User* permission to allow users to connect their artifacts to the connection's data source.

The screenshot shows the 'Manage users' interface for the 'AdventureWorksDW' data source. A user named 'KU' (Owner) is listed under 'Shared with'. On the right, a red box highlights three permission levels:

- User: Allows the user to use the data source.
- User with resharing: Allows the user to use the data source and reshare with others.
- Owner: Allows the user to use the data source, manage data source configurations and credentials.

Once you've found the user and assigned permission, select **Share** at the bottom of the **Manage users** window to apply your selections.

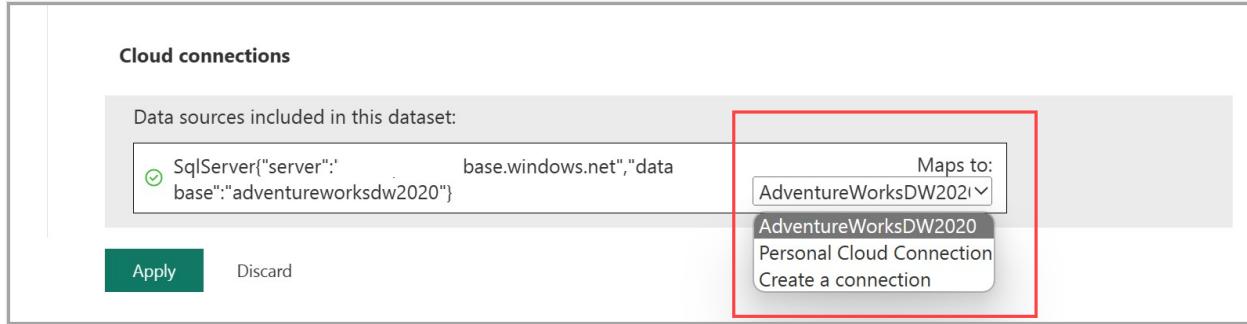
Assign a shared cloud connection to a dataset

Once you've created a shareable cloud connection, you can assign it to a dataset.

Open the settings for the dataset to which you want the shareable connection to apply, and expand the **Gateway and cloud connections** section. You'll notice that the connection is mapped to a *Personal Cloud Connection* by default.

The screenshot shows the 'Gateway and cloud connections' section of a dataset settings pane. It lists data sources included in the dataset and their mappings. A red box highlights the 'Maps to:' dropdown for a specific connection.

From the **Maps to** drop down, select the name of the shareable connection you created and want to use, then select **Apply**.



That's it, you've now assigned your shareable cloud connection to the dataset.

If you haven't created a shareable cloud connection yet when you're using this screen, you can select the **Create a connection** option from the drop-down to be taken to the **Manage connections and gateways** experience, and all the connection details from the data source for which you selected the **Create a connection** drop-down are prepopulated in the **Create new cloud connection** form.

Next steps

For important information about shareable cloud connections, including limitations and considerations, read the following article:

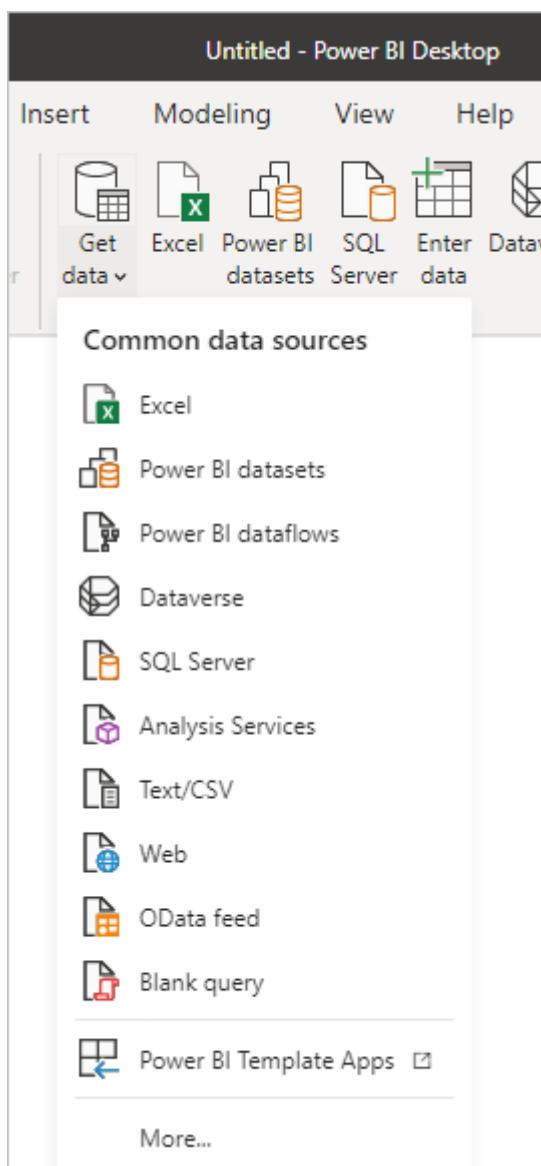
[Connect to cloud data sources in the Power BI service \(Preview\)](#)

Data sources in Power BI Desktop

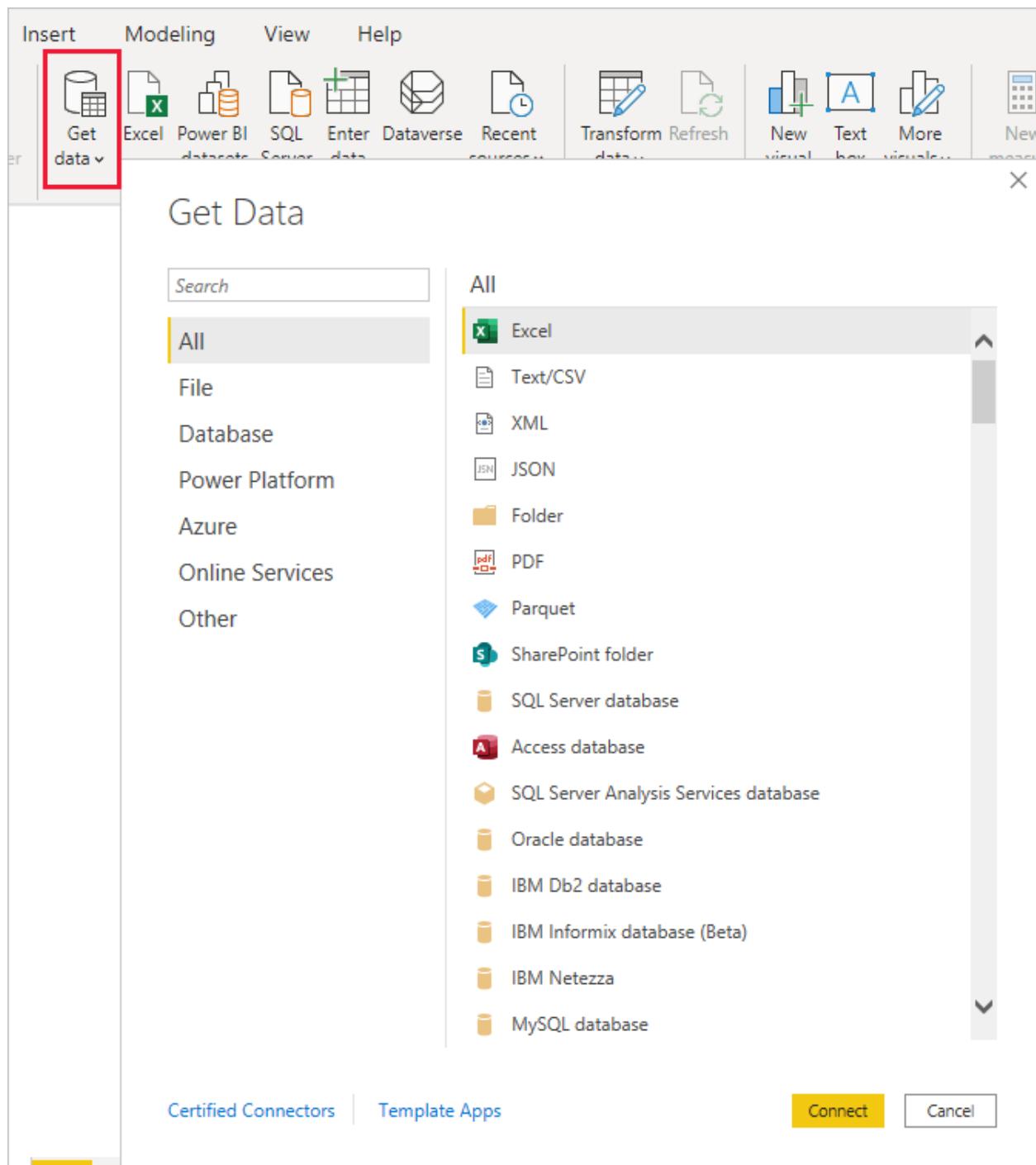
Article • 05/23/2023

With Power BI Desktop, you can connect to data from many different sources. For a full list of available data sources, see [Power BI data sources](#).

To see available data sources, in the **Home** group of the Power BI Desktop ribbon, select the **Get data** button label or down arrow to open the **Common data sources** list. If the data source you want isn't listed under **Common data sources**, select **More** to open the **Get Data** dialog box.



Or, open the **Get Data** dialog box directly by selecting the **Get Data** icon itself.



① Note

The Power BI team is continually expanding the data sources available to Power BI Desktop and the Power BI service. As such, you'll often see early versions of work-in-progress data sources marked as **Beta** or **Preview**. Any data source marked as **Beta** or **Preview** has limited support and functionality, and it shouldn't be used in production environments. Additionally, any data source marked as **Beta** or **Preview** for Power BI Desktop may not be available for use in the Power BI service or other Microsoft services until the data source becomes generally available (GA).

Data sources

The **Get Data** dialog box organizes data types in the following categories:

- All
- File
- Database
- Power Platform
- Azure
- Online Services
- Fabric
- Other

The **All** category includes all data connection types from all categories.

File data sources

The **File** category provides the following data connections:

- Excel Workbook
- Text/CSV
- XML
- JSON
- Folder
- PDF
- Parquet
- SharePoint folder

Database data sources

The **Database** category provides the following data connections:

- SQL Server database
- Access database
- SQL Server Analysis Services database
- Oracle database
- IBM Db2 database
- IBM Informix database (Beta)
- IBM Netezza
- MySQL database
- PostgreSQL database
- Sybase database

- Teradata database
- SAP HANA database
- SAP Business Warehouse Application Server
- SAP Business Warehouse Message Server
- Amazon Redshift
- Impala
- Google BigQuery
- Google BigQuery (Azure AD)(Beta)
- Vertica
- Snowflake
- Essbase
- Actian (Beta)
- Amazon Athena
- AtScale cubes
- BI Connector
- Data Virtuality LDW
- Denodo
- Dremio Software
- Dremio Cloud
- Exasol
- Indexima
- InterSystems IRIS (Beta)
- Jethro (Beta)
- Kyligence
- Linkar PICK Style / MultiValue Databases (Beta)
- MariaDB
- MarkLogic
- MongoDB Atlas SQL (Beta)
- TIBCO® Data Virtualization

Note

Some database connectors require that you enable them by selecting **File > Options and settings > Options** then selecting **Preview Features** and enabling the connector. If you don't see some of the connectors mentioned above and want to use them, check your **Preview Features** settings. Also note that any data source marked as **Beta** or **Preview** has limited support and functionality, and shouldn't be used in production environments.

Power Platform data sources

The **Power Platform** category provides the following data connections:

- Power BI datasets
- Datacharts (preview)
- Lakehouses (Preview)
- Warehouses (Preview)
- Power BI dataflows (Legacy)
- Common Data Service (Legacy)
- Dataverse
- Dataflows

Azure data sources

The **Azure** category provides the following data connections:

- Azure SQL Database
- Azure Synapse Analytics SQL
- Azure Analysis Services database
- Azure Database for PostgreSQL
- Azure Blob Storage
- Azure Table Storage
- Azure Cosmos DB v1
- Azure Data Explorer (Kusto)
- Azure Data Lake Storage Gen2
- Azure Data Lake Storage Gen1
- Azure HDInsight (HDFS)
- Azure HDInsight Spark
- HDInsight Interactive Query
- Azure Cost Management
- Azure Cosmos DB v2 (Beta)
- Azure Databricks
- Azure Synapse Analytics workspace (Beta)
- Azure Time Series Insights (Beta)

Online Services data sources

The **Online Services** category provides the following data connections:

- SharePoint Online List
- Microsoft Exchange Online
- Dynamics 365 Online (legacy)

- Dynamics 365 (Dataverse)
- Dynamics NAV
- Dynamics 365 Business Central
- Dynamics 365 Business Central (on-premises)
- Azure DevOps (Boards only)
- Azure DevOps Server (Boards only)
- Salesforce Objects
- Salesforce Reports
- Google Analytics
- Adobe Analytics
- appFigures (Beta)
- Data.World - Get Dataset (Beta)
- GitHub (Beta)
- LinkedIn Sales Navigator (Beta)
- Marketo (Beta)
- Mixpanel (Beta)
- Planview Enterprise One - PRM (Beta)
- QuickBooks Online (Beta)
- Smartsheet
- SparkPost (Beta)
- SweetIQ (Beta)
- Planview Enterprise Architecture
- Zendesk (Beta)
- Asana (Beta)
- Assemble Views
- Autodesk Construction Cloud
- Automation Anywhere
- Automy Data Analytics (Beta)
- CData Connect Cloud
- Dynamics 365 Customer Insights (Beta)
- Databricks
- Digital Construction Works Insights
- Emigo Data Source
- Entersoft Business Suite (Beta)
- eWay-CRM
- FactSet Analytics
- Palantir Foundry
- Funnel
- Hexagon PPM Smart® API
- Industrial App Store

- Intune Data Warehouse (Beta)
- Planview Projectplace
- Product Insights (Beta)
- Profisee (Beta)
- Quickbase
- SoftOne BI (Beta)
- Planview IdeaPlace
- TeamDesk (Beta)
- Webtrends Analytics (Beta)
- Witivio (Beta)
- Viva Insights
- Zoho Creator

Fabric

The **Fabric** category provides the following data connections:

- Warehouse in Microsoft Fabric
- Power BI datasets
- Datamarts (preview)
- KQL Database in Microsoft Fabric
- Lakehouse (Beta)
- Dataflows
- Warehouse (Beta)

Other data sources

The **Other** category provides the following data connections:

- Web
- SharePoint list
- OData Feed
- Active Directory
- Microsoft Exchange
- Hadoop File (HDFS)
- Spark
- Hive LLAP
- R script
- Python script
- ODBC
- OLE DB

- Acterys : Model Automation & Planning (Beta)
- Amazon OpenSearch Service (Beta)
- Anaplan
- Solver
- BitSight Security Ratings
- BQE Core
- Bloomberg Data and Analytics
- Celonis EMS (Beta)
- Cherwell (Beta)
- CloudBluePSA (Beta)
- Cognite Data Fusion
- Delta Sharing
- Eduframe (Beta)
- EQuIS (Beta)
- FactSet RMS (Beta)
- FHIR
- Google Sheets
- Information Grid (Beta)
- Jamf Pro (Beta)
- Kognitwin
- MicroStrategy for Power BI
- OpenSearch Project (Beta)
- Paxata
- QubolePresto (Beta)
- Roamler (Beta)
- SIS-CC SDMX (Beta)
- Shortcuts Business Insights (Beta)
- SingleStore Direct Query Connector 1.0 (Beta)
- Siteimprove
- Socialbakers Metrics 1.1.0 (Beta)
- Starburst Enterprise
- SumTotal
- SurveyMonkey
- Microsoft Teams Personal Analytics (Beta)
- Tenforce (Smart)List
- Usercube (Beta)
- Vena
- Vessel Insight
- Wrike (Beta)
- Zucchetti HR Infinity (Beta)

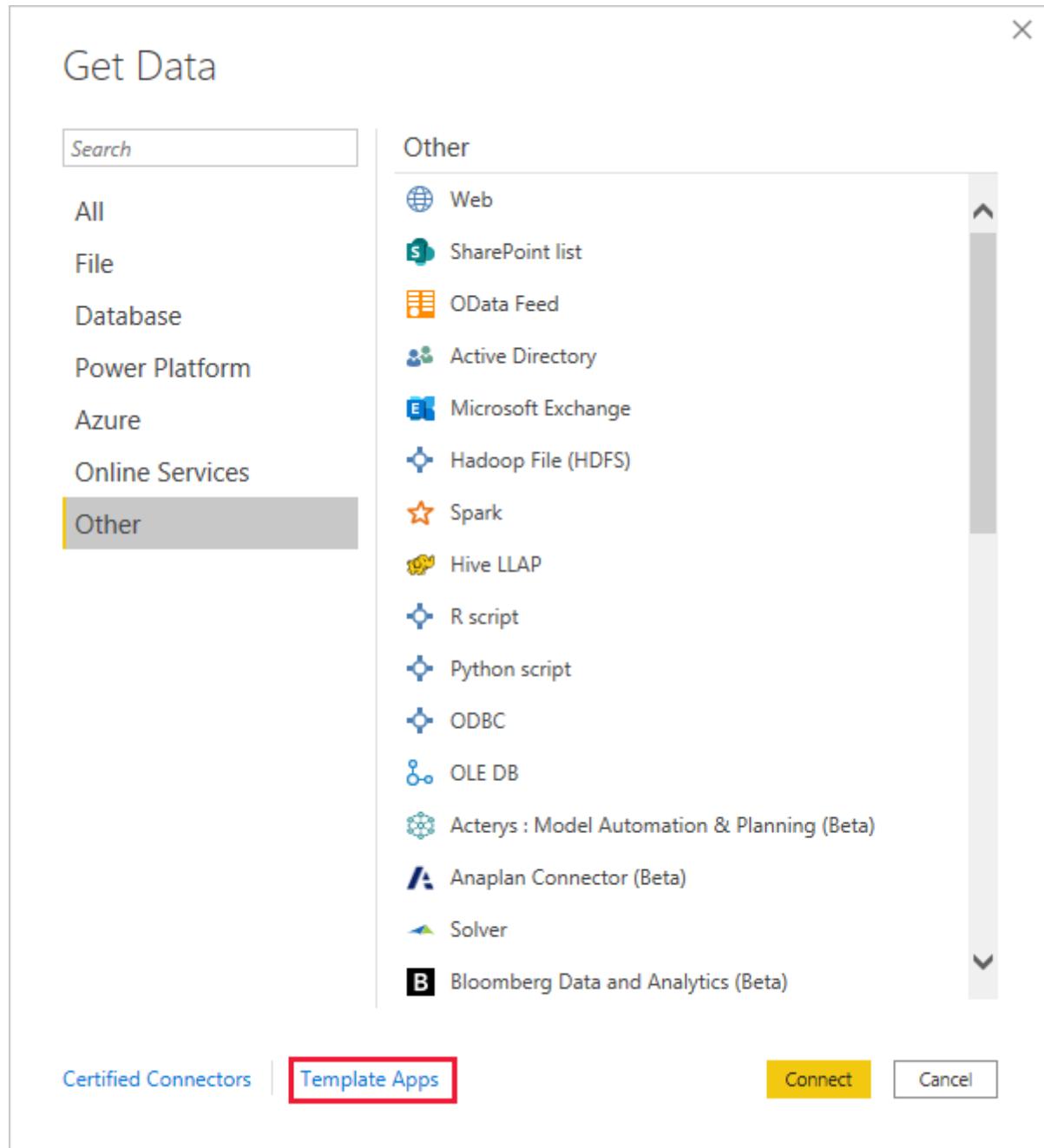
- Blank Query

① Note

At this time, it's not possible to connect to custom data sources secured using Azure Active Directory.

Template apps

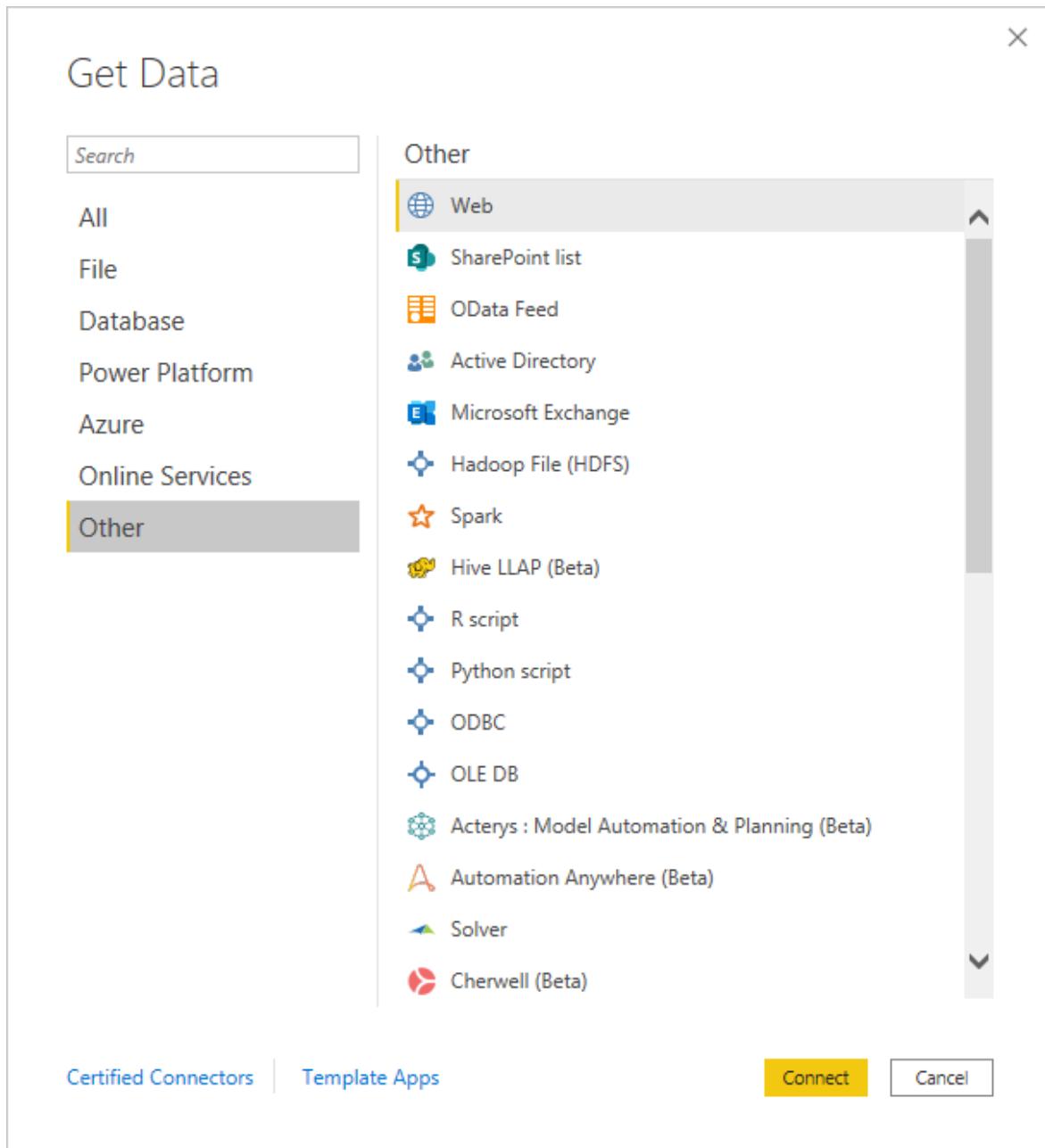
You can find template apps for your organization by selecting the **Template Apps** link near the bottom of the **Get Data** window.



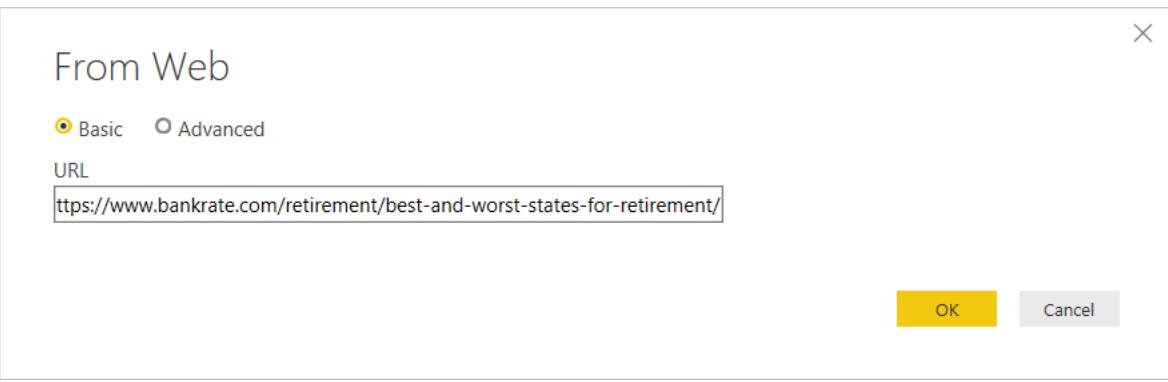
Available Template Apps may vary based on your organization.

Connect to a data source

1. To connect to a data source, select the data source from the **Get Data** window and select **Connect**. The following screenshot shows **Web** selected from the **Other** data connection category.



2. A connection window appears. Enter the URL or resource connection information, and then select **OK**. The following screenshot shows a URL entered in the **From Web** connection dialog box.



3. Depending on the data connection, you might be prompted to provide credentials or other information. After you provide all required information, Power BI Desktop connects to the data source and presents the available data sources in the **Navigator** dialog box.

4. Select the tables and other data that you want to load. To load the data, select the **Load** button at the bottom of the **Navigator** pane. To transform or edit the query in Power Query Editor before loading the data, select the **Transform Data** button.

That's all there is to connecting to data sources in Power BI Desktop! Try connecting to data from our growing list of data sources, and check back often. We continue to add to this list all the time.

Use PBIDS files to get data

PBIDS files are Power BI Desktop files that have a specific structure and a *.PBIDS* extension to identify them as Power BI data source files.

You can create a PBIDS file to streamline the **Get Data** experience for new or beginner report creators in your organization. If you create the PBIDS file from existing reports, it's easier for beginning report authors to build new reports from the same data.

When an author opens a PBIDS file, Power BI Desktop prompts the user for credentials to authenticate and connect to the data source that the file specifies. The **Navigator** dialog box appears, and the user must select the tables from that data source to load into the model. Users may also need to select the database(s) and connection mode if none was specified in the PBIDS file.

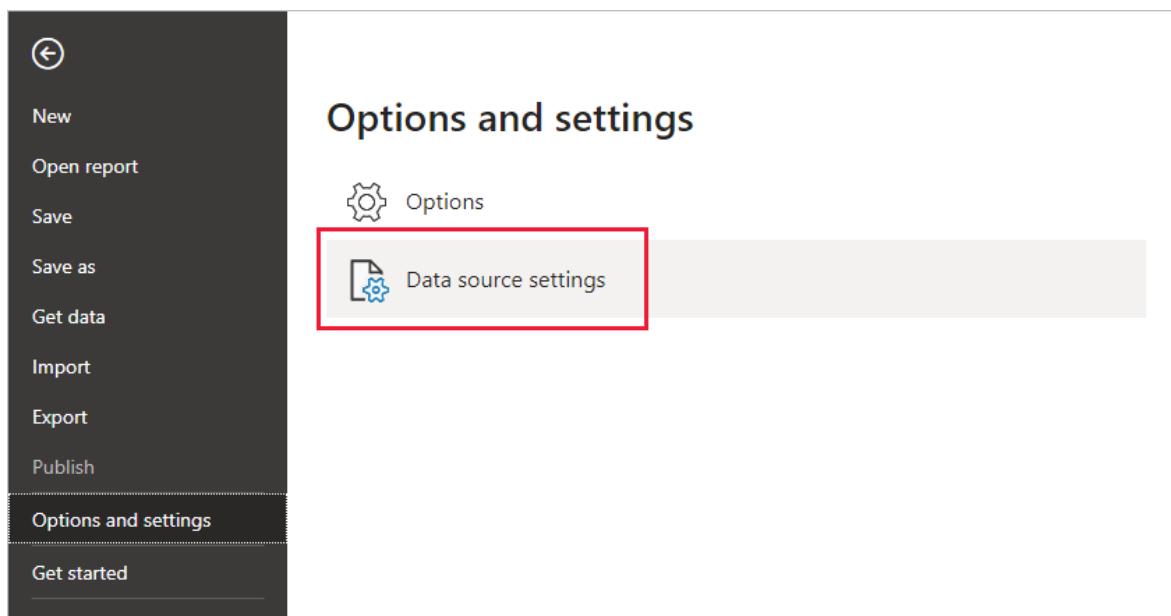
From that point forward, the user can begin building visualizations or select **Recent Sources** to load a new set of tables into the model.

Currently, PBIDS files only support a single data source in one file. Specifying more than one data source results in an error.

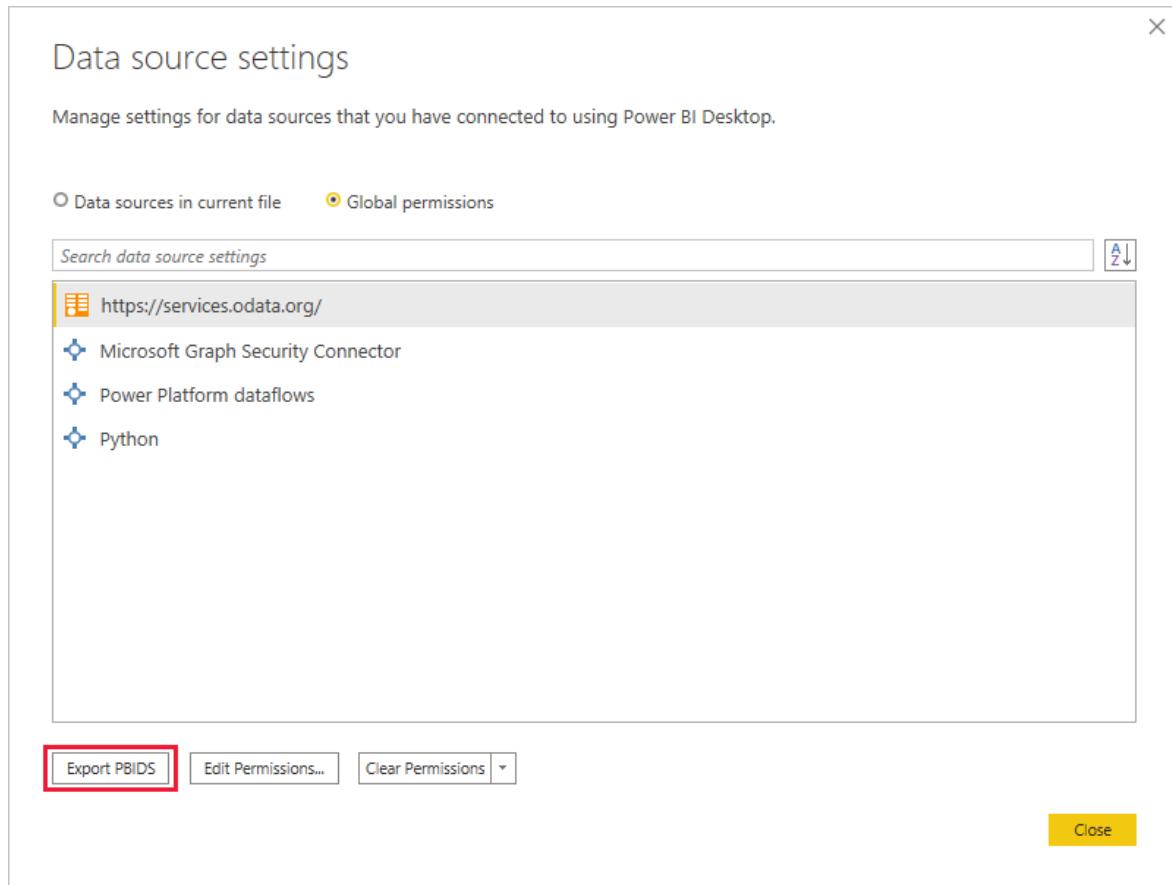
How to create a PBIDS connection file

If you have an existing Power BI Desktop PBIX file that's already connected to the data you're interested in, you can export these connection files from within Power BI Desktop. This method is recommended, since the PBIDS file can be auto-generated from Desktop. You can also still edit or manually create the file in a text editor.

1. To create the PBIDS file, select **File > Options and settings > Data source settings**.



2. In the dialog that appears, select the data source you want to export as a PBIDS file, and then select **Export PBIDS**.



3. In the **Save As** dialog box, give the file a name, and select **Save**. Power BI Desktop generates the PBIDS file, which you can rename and save in your directory, and share with others.

You can also open the file in a text editor, and modify the file further, including specifying the mode of connection in the file itself. The following image shows a PBIDS file open in a text editor.

The screenshot shows a text editor displaying a PBIDS file. The left pane shows a preview icon of a database and the file name 'SQL db connection.pbids'. The right pane shows the JSON content of the PBIDS file:

```
1 "version": "0.1",
2 "connections": [
3     {
4         "details": {
5             "protocol": "tds",
6             "address": {
7                 "server": "sqlclldb4",
8                 "database": "AS_AdventureWorksDW2012"
9             },
10            "authentication": null,
11            "query": null
12        },
13        "options": {},
14        "mode": null
15    }
16]
17}
18}
```

If you prefer to manually create your PBIDS files in a text editor, you must specify the required inputs for a single connection and save the file with the *.PBIDS* extension.

Optionally, you can also specify the connection `mode` as either `DirectQuery` or `Import`. If `mode` is missing or `null` in the file, the user who opens the file in Power BI Desktop is prompted to select `DirectQuery` or `Import`.

ⓘ Important

Some data sources will generate an error if columns are encrypted in the data source. For example, if two or more columns in an Azure SQL Database are encrypted during an Import action, an error will be returned. For more information, see [SQL Database](#).

PBIDS file examples

This section provides some examples from commonly used data sources. The PBIDS file type only supports data connections that are also supported in Power BI Desktop, with the following exceptions: Wiki URLs, Live Connect, and Blank Query.

The PBIDS file doesn't include authentication information and table and schema information.

The following code snippets show several common examples for PBIDS files, but they aren't complete or comprehensive. For other data sources, you can refer to the [git Data Source Reference \(DSR\) format for protocol and address information](#).

If you're editing or manually creating the connection files, these examples are for convenience only, aren't meant to be comprehensive, and don't include all supported connectors in DSR format.

Azure AS

JSON

```
{  
    "version": "0.1",  
    "connections": [  
        {  
            "details": {  
                "protocol": "analysis-services",  
                "address": {  
                    "server": "server-here"  
                },  
            }  
        }  
    ]  
}
```

```
        ]  
    }
```

Folder

JSON

```
{  
  "version": "0.1",  
  "connections": [  
    {  
      "details": {  
        "protocol": "folder",  
        "address": {  
          "path": "folder-path-here"  
        }  
      }  
    }  
  ]  
}
```

OData

JSON

```
{  
  "version": "0.1",  
  "connections": [  
    {  
      "details": {  
        "protocol": "odata",  
        "address": {  
          "url": "URL-here"  
        }  
      }  
    }  
  ]  
}
```

SAP BW

JSON

```
{  
  "version": "0.1",  
  "connections": [  
    {
```

```
        "details": {
            "protocol": "sap-bw-olap",
            "address": {
                "server": "server-name-here",
                "systemNumber": "system-number-here",
                "clientId": "client-id-here"
            },
        },
    }
]
}
```

SAP Hana

JSON

```
{
    "version": "0.1",
    "connections": [
        {
            "details": {
                "protocol": "sap-hana-sql",
                "address": {
                    "server": "server-name-here:port-here"
                },
            }
        }
    ]
}
```

SharePoint list

The URL must point to the SharePoint site itself, not to a list within the site. Users get a navigator that allows them to select one or more lists from that site, each of which becomes a table in the model.

JSON

```
{
    "version": "0.1",
    "connections": [
        {
            "details": {
                "protocol": "sharepoint-list",
                "address": {
                    "url": "URL-here"
                },
            }
        }
    ]
}
```

```
]  
}
```

SQL Server

JSON

```
{  
  "version": "0.1",  
  "connections": [  
    {  
      "details": {  
        "protocol": "tds",  
        "address": {  
          "server": "server-name-here",  
          "database": "db-name-here (optional)"  
        }  
      },  
      "options": {},  
      "mode": "DirectQuery"  
    }  
  ]  
}
```

Text file

JSON

```
{  
  "version": "0.1",  
  "connections": [  
    {  
      "details": {  
        "protocol": "file",  
        "address": {  
          "path": "path-here"  
        }  
      }  
    }  
  ]  
}
```

Web

JSON

```
{  
    "version": "0.1",  
    "connections": [  
        {  
            "details": {  
                "protocol": "http",  
                "address": {  
                    "url": "URL-here"  
                }  
            }  
        }  
    ]  
}
```

Dataflow

JSON

```
{  
    "version": "0.1",  
    "connections": [  
        {  
            "details": {  
                "protocol": "powerbi-dataflows",  
                "address": {  
                    "workspace": "workspace id (Guid)",  
                    "dataflow": "optional dataflow id (Guid)",  
                    "entity": "optional entity name"  
                }  
            }  
        }  
    ]  
}
```

Next steps

You can do all sorts of things with Power BI Desktop. For more information on its capabilities, check out the following resources:

- [What is Power BI Desktop?](#)
- [Query overview with Power BI Desktop](#)
- [Data types in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)

Dynamic M query parameters in Power BI Desktop

Article • 04/14/2023

This article describes how to create and work with dynamic M query parameters in Power BI Desktop. With dynamic M query parameters, model authors can configure the filter or slicer values that report viewers can use for an [M query parameter](#). Dynamic M query parameters give model authors more control over the filter selections to incorporate into DirectQuery source queries.

Model authors understand the intended semantics of their filters, and often know how to write efficient queries against their data source. With dynamic M query parameters, model authors can ensure that filter selections incorporate into source queries at the right point to achieve the intended results with optimum performance. Dynamic M query parameters can be especially useful for query performance optimization.

Watch Sujata explain and use dynamic M query parameters in the following video, and then try them out yourself.

ⓘ Note

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

<https://www.microsoft.com/en-us/videoplayer/embed/RE4QLsb?postJs||Msg=true> ↗

Prerequisites

To work through these procedures, you must have a valid [M query](#) that uses one or more DirectQuery tables.

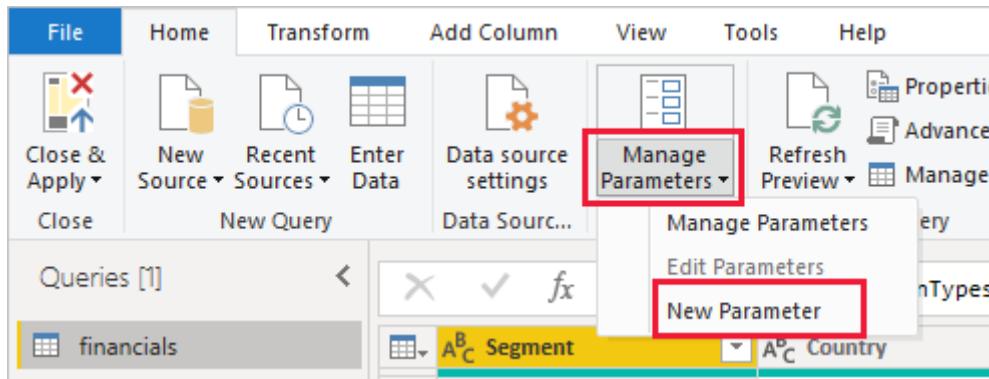
Create and use dynamic parameters

The following example passes a single value through to a parameter dynamically.

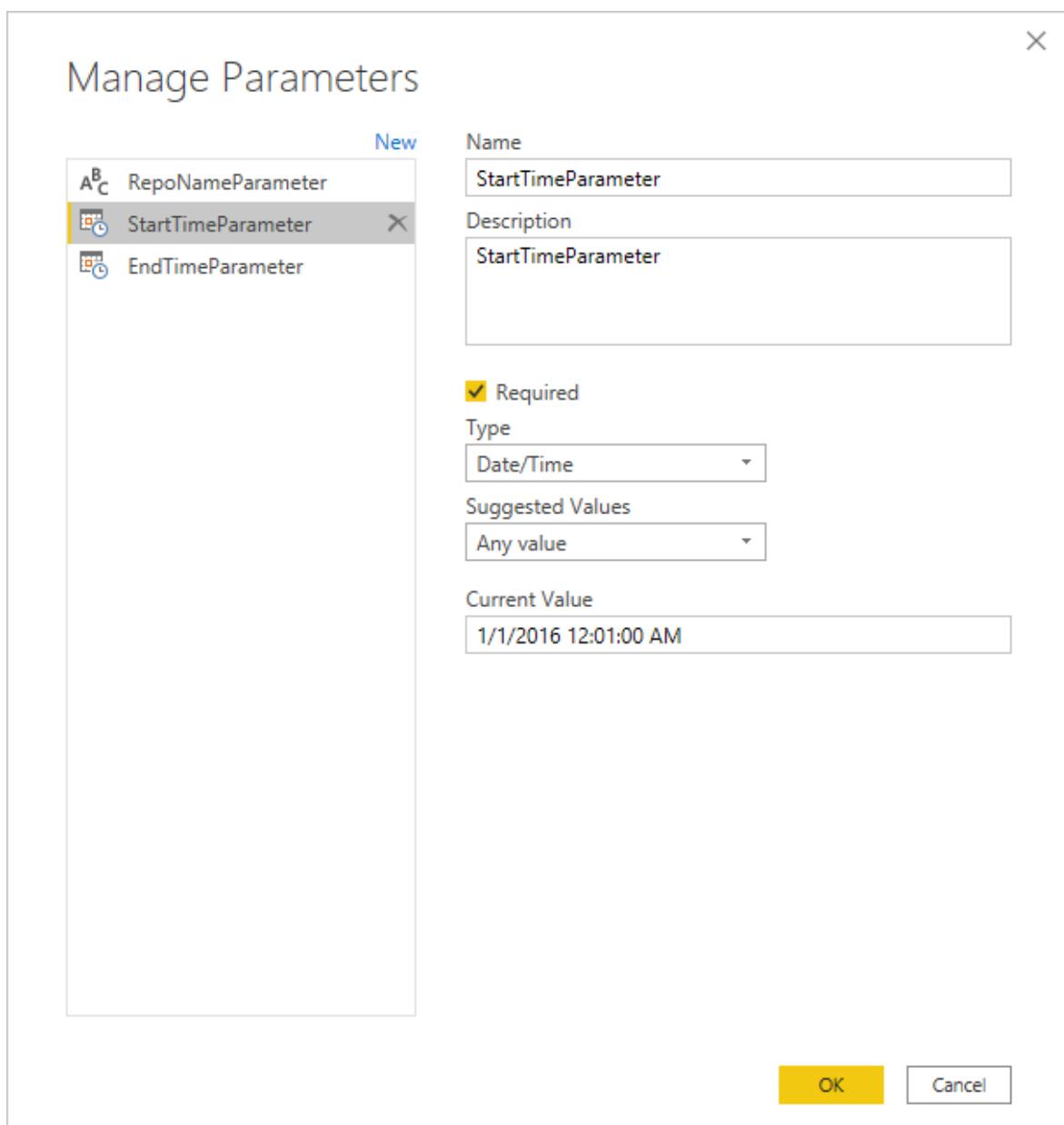
Add parameters

1. In Power BI Desktop, select **Home > Transform data > Transform data** to open the Power Query Editor.

2. In the Power Query Editor, select **New Parameters** under **Manage Parameters** in the ribbon.

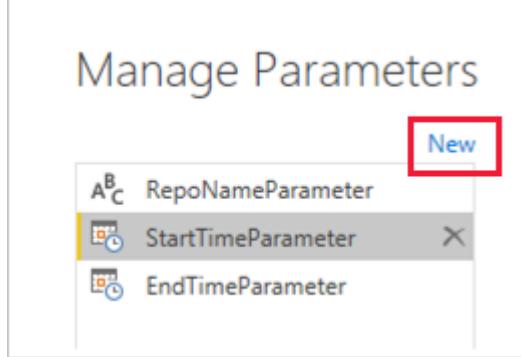


3. In the **Manage Parameters** window, fill out the information about the parameter. For more information, see [Create a parameter](#).



4. Select **New** to add more parameters.

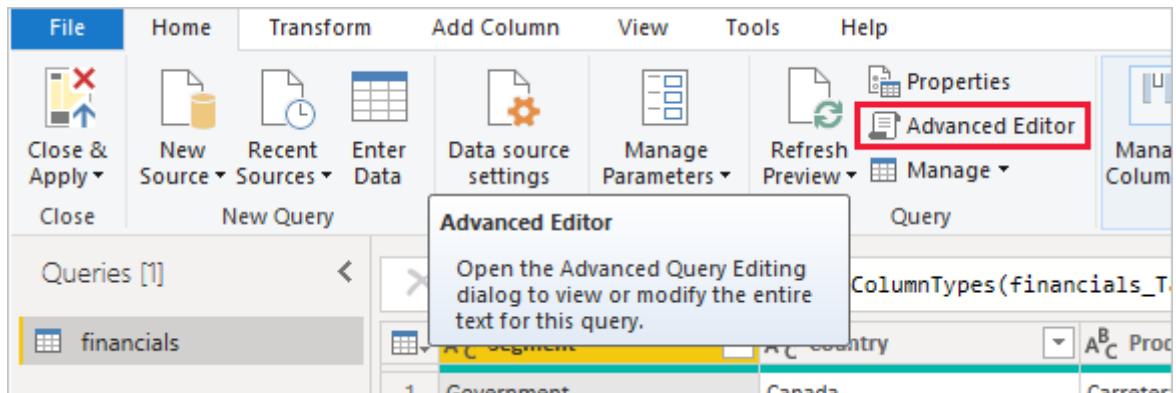
Manage Parameters



- When you're done adding parameters, select OK.

Reference the parameters in the M query

- Once you create the parameters, you can reference them in the M query. To modify the M query, while you have the query selected, open the **Advanced editor**.



- Reference the parameters in the M query, as highlighted in yellow in the following image:

```

let
    selectedRepoNames = if Type.Is(Value.Type(RepoNameParameter), List.Type) then
        Text.Combine("", Text.Combine(RepoNameParameter, ",") , ""))
    else
        Text.Combine("", RepoNameParameter , ""),
    
    KustoParametersDeclareQuery = Text.Combine("declare query_parameters",
        "startTime:datetime = datetime(", DateTime.ToString(StartTimeParameter, "yyyy-MM-dd hh:mm"), "),",
        "endTime:datetime = datetime(", DateTime.ToString(EndTimeParameter, "yyyy-MM-dd hh:mm:ss"), "),",
        "repoNames: dynamic = dynamic([" + selectedRepoNames + "]);"),
    
    ActualQueryWithKustoParameters = "GitHubEvent
        | extend RepoName = tostring(Repo.name)
        | where RepoName in(repoNames) make-series count() on CreatedAt from startTime to endTime step 12h by RepoName
        | evaluate python(typeof(*), 'result = df')
        | mv-expand count_ to typeof(long), CreatedAt to typeof(datetime)",
    
    finalQuery = Text.Combine((KustoParametersDeclareQuery, ActualQueryWithKustoParameters)),
    
    Source = AzureDataExplorer.Contents("demo11.westus", "GitHub", finalQuery, [MaxRows=null, MaxSize=null, NoTruncate=null, AdditionalSetStatements=null])
in
    Source

```

No syntax errors have been detected.

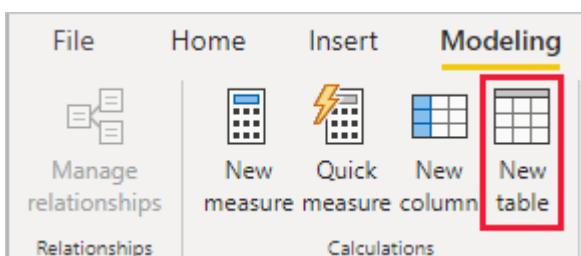
Done **Cancel**

3. When you're done editing the query, select **Done**.

Create tables of values

Create a table for each parameter with a column that provides the possible values available to be dynamically set based on filter selection. In this example, you want the `StartTime` and `EndTime` parameters to be dynamic. Since these parameters require a `Date/Time` parameter, you generate the possible inputs to dynamically set the date for the parameter.

1. In the Power BI Desktop ribbon, under **Modeling**, select **New Table**.



2. Create a table for the values of the `StartTime` parameter, for example:

```
StartDateTable = CALENDAR (DATE(2016,1,1), DATE(2016,12,31))
```

The screenshot shows the Power BI Data View interface. The ribbon at the top has 'Table tools' selected. A query named 'StartDateTable' is currently being edited. The structure pane on the left shows a single column named 'Date'. The preview pane on the right displays five rows of date values from January 1, 2016, to January 5, 2016.

Date
1/1/2016 12:00:00 AM
1/2/2016 12:00:00 AM
1/3/2016 12:00:00 AM
1/4/2016 12:00:00 AM
1/5/2016 12:00:00 AM

3. Create a second table for the values of the `EndTime` parameter, for example:

```
EndDateTable = CALENDAR (DATE(2016,1,1), DATE(2016,12,31))
```

The screenshot shows the Power BI Data View interface. The ribbon at the top has 'Table tools' selected. A query named 'EndDateTable' is currently being edited. The structure pane on the left shows a single column named 'Date'. The preview pane on the right displays five rows of date values from January 1, 2016, to January 5, 2016.

Date
1/1/2016 12:00:00 AM
1/2/2016 12:00:00 AM
1/3/2016 12:00:00 AM
1/4/2016 12:00:00 AM
1/5/2016 12:00:00 AM

ⓘ Note

Use a column name that's not in an actual table. If you use the same name as an actual table column, the selected value applies as a filter in the query.

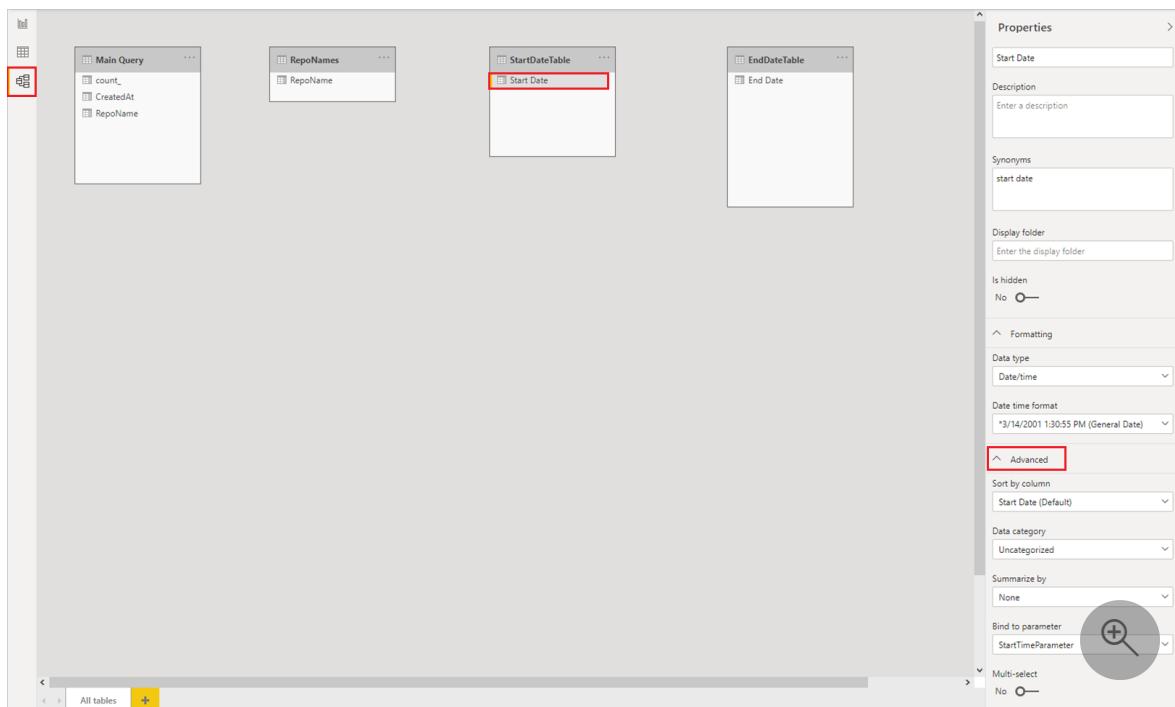
Bind the fields to the parameters

Now that you created the tables with the `Date` fields, you can bind each field to a parameter. Binding a field to a parameter means that as the selected field value changes, the value passes to the parameter and updates the query that references the parameter.

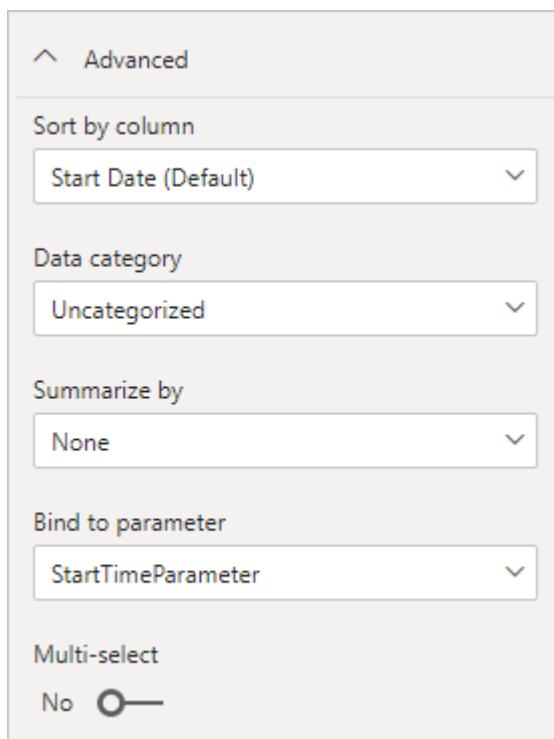
1. To bind a field, in the Power BI Desktop Model view, select the newly created field, and in the **Properties** pane, select **Advanced**.

! Note

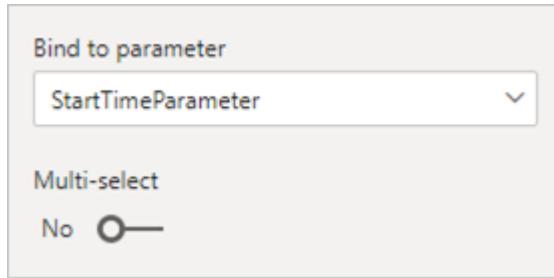
The column data type should match the M parameter data type.



2. Select the dropdown under **Bind to parameter** and select the parameter that you want to bind to the field:



Since this example is for setting the parameter to a single value, keep **Multi-select** set to **No**, which is the default:



If you set the mapped column to **No** for **Multi-select**, you must use a single select mode in the slicer, or require single select in the filter card.

If your use cases require passing multiple values to a single parameter, set the control to **Yes** and make sure your M query is set up to accept multiple values. Here's an example for `RepoNameParameter`, which allows multiple values:

The screenshot shows the 'Advanced Editor' window with the title 'KustoQuery'. The code editor contains the following Kusto query:

```
let
    selectedRepoNames = if Type.Is(Value.Type(RepoNameParameter), List.Type) then
        Text.Combine({""}, Text.Combine(RepoNameParameter, ","))
    else
        Text.Combine({""}, RepoNameParameter, ""));
KustoParametersDeclareQuery = Text.Combine(`declare query_parameters(
    "startTime:datetime = datetime("${", DateTime.ToText(StartTimeParameter, "yyyy-MM-dd hh:mm"), "}", ",
    "endTime:datetime = datetime("${", DateTime.ToText(EndTimeParameter, "yyyy-MM-dd hh:mm:ss"), "}", ",
    "repoNames: dynamic = dynamic(["${", selectedRepoNames, ""]});")`));
ActualQueryWithKustoParameters = "GitHubEvent
    | extend RepoName = tostring(Repo.name)
    | where RepoName in(repoNames) | make-series count() on CreatedAt from startTime to endTime step 12h by RepoName
    | evaluate python(typeof(*), 'result = df')
    | mv-expand count_ to typeof(long), CreatedAt to typeof(datetime),
finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),
Source = AzureDataExplorer.Contents("demo11.westus", "GitHub", finalQuery, [MaxRows=null, MaxSize=null, NoTruncate=null, AdditionalSetStatements=null])
in
Source
```

At the bottom of the code editor, there is a note: 'No syntax errors have been detected.' To the right, there are 'Done' and 'Cancel' buttons, and a circular icon with a plus sign and a magnifying glass.

3. Repeat these steps if you have other fields to bind to other parameters.

Properties

- General
- Formatting
- Advanced

Sort by column

End Date (Default)

Data category

Uncategorized

Summarize by

None

Bind to parameter

EndTimeParameter

Multi-select

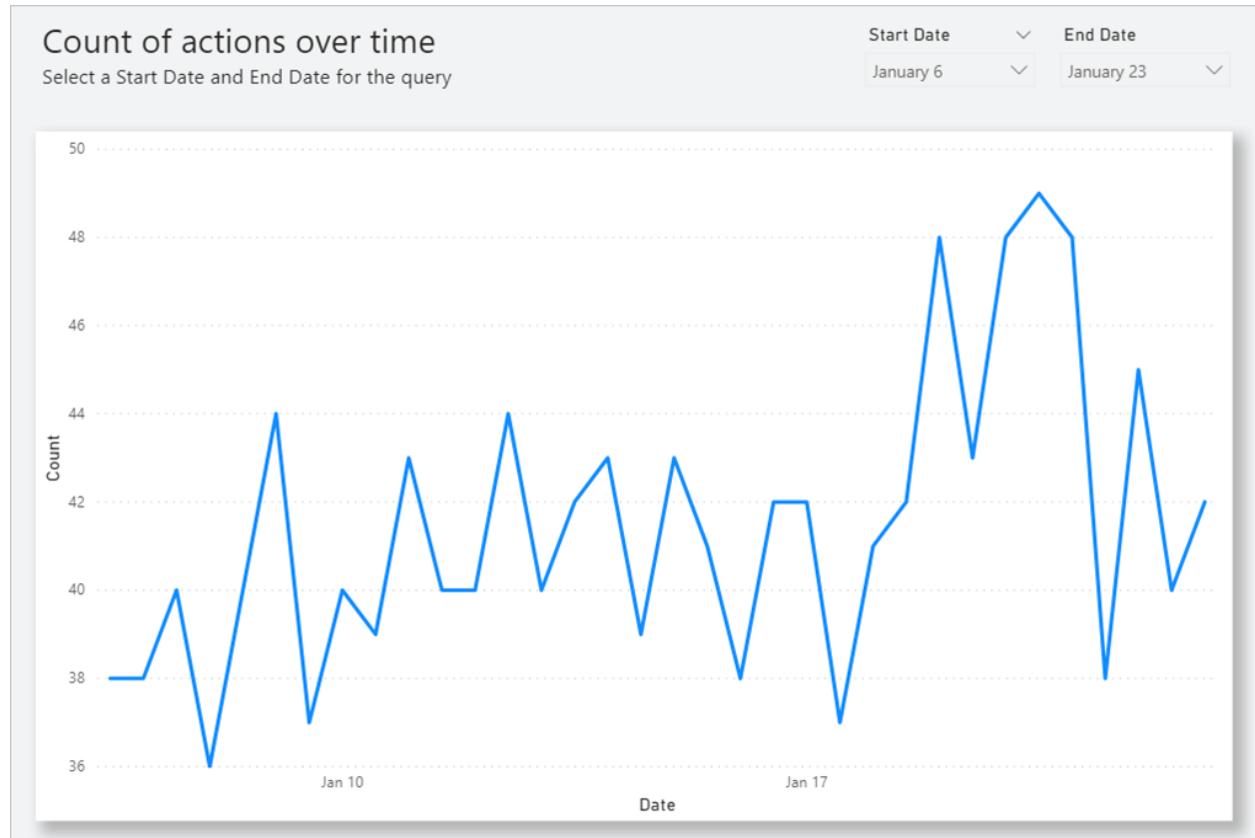
No

Fields

Search

- EndDateTable
 - End Date
- Main Query
- RepoNames
- StartDateTable
 - Start Date

You can now reference this field in a slicer or as a filter:



Enable Select all

In this example, the Power BI Desktop model has a field called **Country**, which is a list of countries/regions bound to an M parameter called **countryNameMParameter**. This parameter is enabled for **Multi-select**, but isn't enabled for **Select all**. To be able to use the **Select all** option in a slicer or filter card, take the following added steps:

The screenshot shows the Power BI Desktop Properties pane on the left and the Fields pane on the right. In the Properties pane, under the 'Multi-select' section, the 'Yes' toggle switch is checked, and the 'Select all' toggle switch is unchecked (indicated by a red border around the 'Select all' section). The Fields pane shows a hierarchy of fields: ConfirmedCases (summarized by Confirmed Cases, Country, Date) and CountryList (Country).

Properties

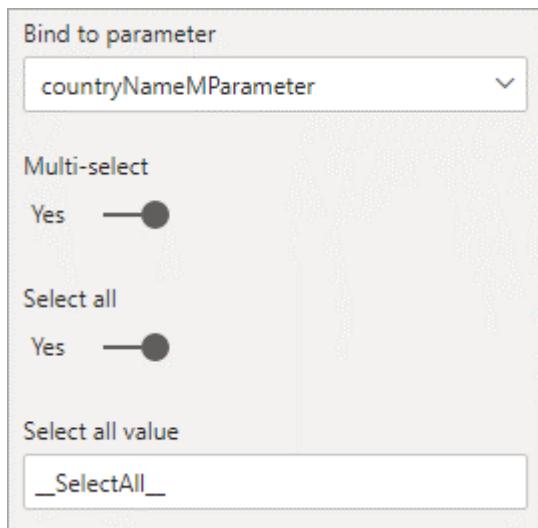
- Display folder: Enter the display folder
- Is hidden: No (toggle switch)
- Formatting
- Data type: Text
- Format: Text
- Advanced
- Sort by column: Country (Default)
- Data category: Uncategorized
- Summarize by: None
- Is nullable: Yes (toggle switch)
- Bind to parameter: countryNameMParameter
- Multi-select**: Yes (checked), Select all (unchecked)
- Select all value: _SelectAll_

Fields

- Search
- ConfirmedCases
 - Σ Confirmed Cases
 - Country
 - Date
- CountryList
 - Country

To enable **Select all** for **Country**:

1. In the **Advanced** properties for **Country**, enable the **Select all** toggle, which enables the **Select all value** input. Edit the **Select all value** or note the default value.



The **Select all value** passes to the parameter as a list that contains the value you defined. Therefore, when you define this value or use the default value, make sure the value is unique and doesn't exist in the field that's bound to the parameter.

2. Launch the Power Query Editor, select the query, and then select **Advanced Editor**. Edit the M query to use the **Select all value** to refer to the **Select all** option.

```
1 let
2     selectedCountryNames = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
3         Text.Combine("", Text.Combine(countryNameMParameter, ","))
4     else
5         Text.Combine("", countryNameMParameter),
6
7     selectAllCountries = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
8         List.Contains(countryNameMParameter, "_SelectAll_")
9     else
10        false,
11
12     KustoParametersDeclareQuery = Text.Combine("declare query_parameters",
13         "startTime:datetime = datetime('2020-01-01T00:00:00Z')",
14         "endTime:datetime = datetime('2020-12-31T23:59:59Z')",
15         "includeAllCountries: bool = " & Logical.ToText(selectAllCountries) & ",",
16         "countryNames: dynamic = dynamic({", selectedCountryNames, "});"),
17
18     ActualQueryWithKustoParameters =
19         "Covid19
20         | where includeAllCountries or Country in(countryNames)
21         | where Timestamp > startTime and Timestamp < endTime
22         | summarize sum(Confirmed) by Country, bin(Timestamp, 30d)",
23
24     finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),
25
26     Source = AzureDataExplorer.Contents("help", "samples", finalQuery, [MaxRows=null, MaxSize=null, NoTruncate=null, AdditionalSetStatements=null]),
27     #"Renamed Columns" = Table.RenameColumns(Source,{{"Timestamp", "Date"}, {"sum_Confirmed", "Confirmed Cases"}}),
28 in
29     #"Renamed Columns"
```

3. In the **Advanced Editor**, add a Boolean expression that evaluates to **true** if the parameter is enabled for **Multi-select** and contains the **Select all value**, and

otherwise returns `false`:

```
KustoParametersDeclareQuery = Text.Combine({"declare query_parameters",
    "starttime:datetime = datetime(", DateTime.ToText(StartTimeMParameter, "yyyy-MM-dd hh:mm"), "),",
    "endtime:datetime = datetime(", DateTime.ToText(EndTimeMParameter, "yyyy-MM-dd hh:mm:ss"), "),",
    "includeAllCountries: bool = ", Logical.ToString(selectAllCountries), ",",
    "countryNames: dynamic = dynamic([" , selectedCountryNames, "]);"},),

ActualQueryWithKustoParameters =
    "Covid19
    | where includeAllCountries or Country in(countryNames)
    | where Timestamp > starttime and Timestamp < endtime
    | summarize sum(Confirmed) by Country, bin(Timestamp, 30d)",

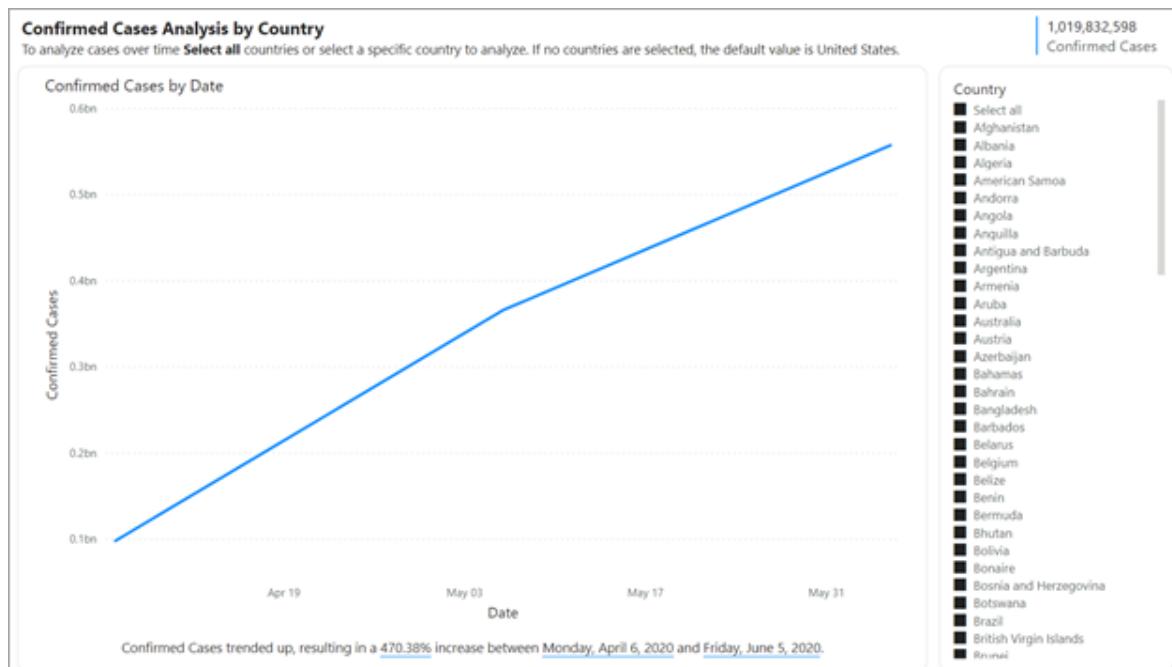
finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),
```

4. Incorporate the result of the **Select all** Boolean expression into the source query.

The example has a Boolean query parameter in the source query called `includeAllCountries` that is set to the result of the Boolean expression from the previous step. You can use this parameter in a filter clause in the query, such that `false` for the Boolean filters to the selected country or region names, and `true` effectively applies no filter.

```
selectAllCountries = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
    List.Contains(countryNameMParameter, "__SelectAll__")
```

5. Once you update your M query to account for the new **Select all** value, you can use the **Select all** function in slicers or filters.



For reference, here's the full query for the preceding example:

```
Kusto

let
    selectedCountryNames = if Type.Is(Value.Type(countryNameMParameter),
List.Type) then
        Text.Combine({'''', Text.Combine(countryNameMParameter, "','"), '''})
```

```

else
    Text.Combine({''' , countryNameMParameter , '''}),

selectAllCountries = if Type.Is(Value.Type(countryNameMParameter),
List.Type) then
    List.Contains(countryNameMParameter, "__SelectAll__")
else
    false,

KustoParametersDeclareQuery = Text.Combine({ "declare query_parameters(",
                                            "startTimep:datetime = datetime(",
DateTime.ToString(StartTimeMParameter, "yyyy-MM-dd hh:mm"), ",",
                                            "endTimep:datetime = datetime(",
DateTime.ToString(EndTimeMParameter, "yyyy-MM-dd hh:mm:ss"), ",",
                                            "includeAllCountries: bool = ",
Logical.ToString(selectAllCountries) ,",
                                            "countryNames: dynamic = dynamic([" ,
selectedCountryNames, "]));" }),

ActualQueryWithKustoParameters =
    "Covid19
     | where includeAllCountries or Country
in(countryNames)
     | where Timestamp > startTimep and Timestamp
< endTimep
     | summarize sum(Confirmed) by Country,
bin(Timestamp, 30d)",

finalQuery = Text.Combine({KustoParametersDeclareQuery,
ActualQueryWithKustoParameters}),

Source = AzureDataExplorer.Contents("help", "samples", finalQuery,
[MaxRows=null, MaxSize=null, NoTruncate=null,
AdditionalSetStatements=null]),
#"Renamed Columns" = Table.RenameColumns(Source,{{"Timestamp", "Date"}, {"sum_Confirmed", "Confirmed Cases"}})
in
#"Renamed Columns"

```

Potential security risk

Report readers who can dynamically set the values for M query parameters may be able to access more data or trigger modifications to the source system by using *injection attacks*. This possibility depends on how you reference the parameters in the M query and what values you pass to the parameters.

For example, you have a parameterized Kusto query constructed as follows:

Kusto

```
Products
| where Category == [Parameter inserted here] & HasReleased == 'True'
| project ReleaseDate, Name, Category, Region
```

There are no issues with a friendly user who passes an appropriate value for the parameter, for example, `Games`:

```
| where Category == 'Games' & HasReleased == 'True'
```

However, an attacker may be able to pass a value that modifies the query to get access to more data, for example, `'Games'//`:

```
Products
| where Category == 'Games'// & HasReleased == 'True'
| project ReleaseDate, Name, Category, Region
```

In this example, the attacker can get access to information about games that haven't released yet by changing part of the query into a comment.

Mitigate the risk

To mitigate the security risk, avoid string concatenation of M parameter values within the query. Instead, consume those parameter values in M operations that fold to the source query, so that the M engine and connector construct the final query.

If a data source supports importing stored procedures, consider storing your query logic there and invoking it in the M query. Alternatively, if available, use a parameter passing mechanism that's built in to the source query language and connectors. For example, [Azure Data Explorer](#) has built-in query parameter capabilities that are designed to protect against injection attacks.

Here are some examples of these mitigations:

- Example that uses the M query's filtering operations:

Kusto

```
Table.SelectRows(Source, (r) => r[Columns] = Parameter)
```

- Example that declares the parameter in the source query, or passes the parameter value as an input to a source query function:

Kusto

```
declare query_parameters (Name of Parameter : Type of Parameter);
```

- Example of directly calling a stored procedure:

Kusto

```
let CustomerByProductFn = AzureDataExplorer.Contents("Help",
"ContosoSales"){[Name="CustomerByProduct"]}[Data] in
CustomerByProductFn({1, 3, 5})
```

Considerations and limitations

There are some considerations and limitations when you use dynamic M query parameters:

- A single parameter can't be bound to multiple fields nor vice-versa.
- Dynamic M query parameters don't support [aggregations](#).
- Dynamic M query parameters don't support row-level security (RLS).
- Parameter names can't be Data Analysis Expressions (DAX) reserved words nor contain spaces. You can append `Parameter` to the end of the parameter name to help avoid this limitation.
- Table names can't contain spaces or special characters.
- If your parameter is the `Date/Time` data type, you need to cast it within the M query as `DateTime.Date(<YourDateParameter>)`.
- If you use SQL sources, you might get a confirmation dialog every time the parameter value changes. This dialog is due to a security setting: **Require user approval for new native database queries**. You can find and turn off this setting in the **Security** section of the Power BI Desktop Options.
- Dynamic M query parameters may not work when accessing a dataset in Excel.
- Dynamic M query parameters are not supported on Power BI Report Server.

Unsupported out-of-box parameter types

- Any
- Duration
- True/False

- Binary

Unsupported filters

- Relative time slicer or filter
- Relative date
- Hierarchy slicer
- Multifield include filter
- Exclude filters / Not filters
- Cross-highlighting
- Drilldown filter
- Cross drill filter
- Top N filter

Unsupported operations

- And
- Contains
- Less than
- Greater than
- Starts with
- Does not start with
- Is not
- Does not contain
- Is blank
- Is not blank

Next steps

For more information about Power BI Desktop capabilities, check out the following resources:

- [DirectQuery in Power BI Desktop](#)
- [What is Power BI Desktop?](#)
- [Query overview with Power BI Desktop](#)
- [Data types in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)

Create a Power BI dataset directly from Log Analytics

Article • 03/08/2023

You can quickly create a Power BI dataset directly from a Log Analytics query. The dataset will be full-fledged Power BI dataset that you can use to create reports, analyze in Excel, and more.

Creating a dataset directly from a Log Analytics query is an easy and quick way to share a dataset, because if you save it to a shared workspace, everyone with the sufficient permissions in the workspace can use it. You can also use [dataset sharing](#) to share it with other users who don't have a role in the workspace.

This feature creates a dataset in the Power BI service directly from a Log Analytics query. If you need to model or transform the data in ways that aren't available in the service, you can also export the query from Log Analytics, paste it into Power BI Desktop, and do your advanced modeling there. For more information, see [Create Power BI datasets and reports from Log Analytics queries](#).

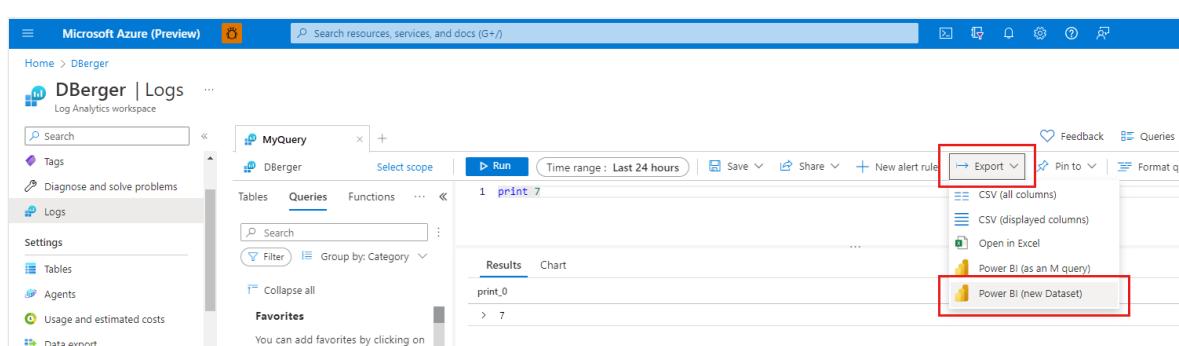
Prerequisites

You must have a Power BI account to be able to use this functionality.

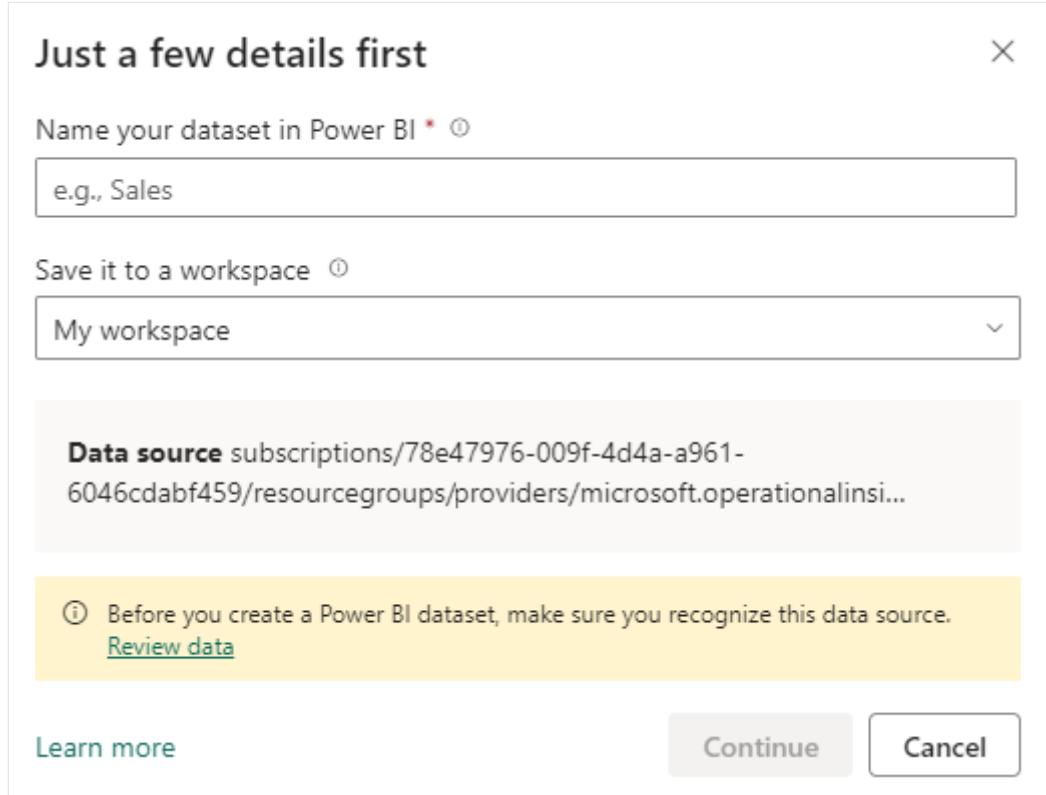
Create a dataset from Log Analytics

To create a Power BI dataset from a Log Analytics query:

1. Open and run the Log Analytics query you want to use to create the Power BI database.
2. In the actions bar, select **Export > Export to Power BI**.



3. Power BI will open and a dialog will ask you to name the dataset and choose a workspace to save it in. By default the dataset will be given the same name as the query and saved to My workspace. You can choose your own name and destination workspace. If you're a [free user](#) in Power BI, you'll only be able to save to My workspace.

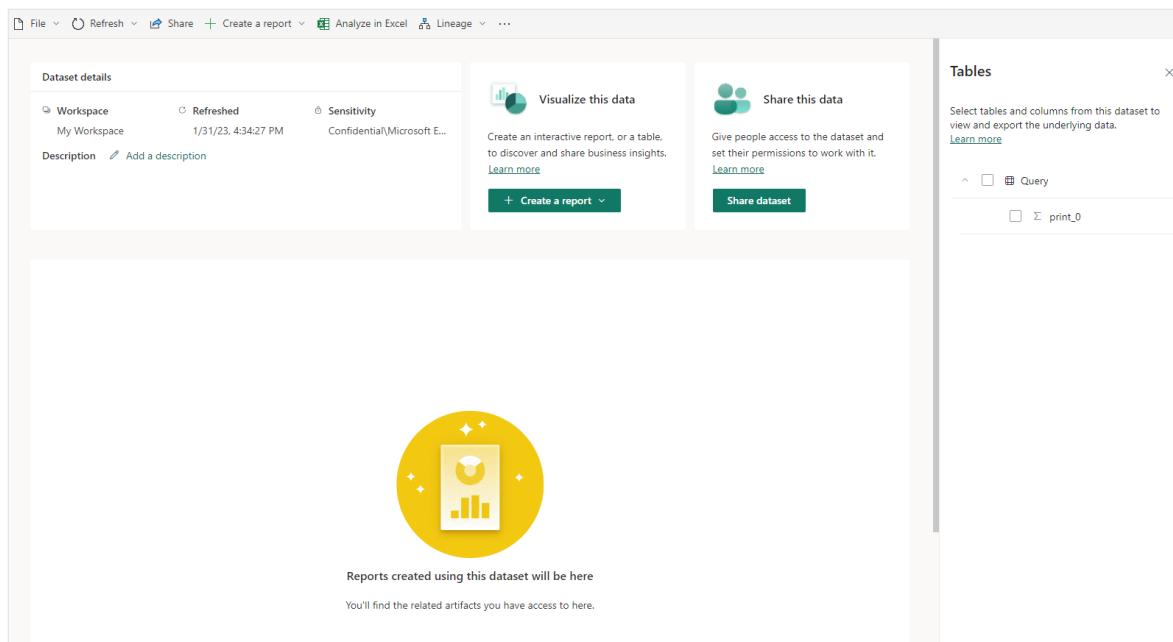


The dialog also shows the URL of the Log Analytics data source. To prevent inadvertently exposing sensitive data, make sure that you recognize the data source and are familiar with the data. Select **Review data** if you want to check the Log Analytic query results before allowing export to continue. For more information about when reviewing the data might be a good idea, see [Reviewing the Log Analytics data](#).

4. Select **Continue**. Your dataset will be created, and you'll be taken to the details page of the new dataset. From there you can do all the things you can do with a regular Power BI dataset - refresh the data, share the dataset, create new reports, and more. See [dataset details](#) for more information.

ⓘ Note

If you've connected to Log Analytics from Power BI before, you'll be asked to choose which credentials to use to for the connection between Power BI and Log Analytics before being taken to the dataset details page. For help deciding which credentials to choose, see [Choosing which credentials to authenticate with](#).



To keep the data fresh after you've created the dataset, either refresh the data manually or set up scheduled refresh.

Reviewing the Log Analytics data

When you export data from a Log Analytics query to Power BI, a redirect URL is created that includes all the parameters needed to launch the create dataset process in Power BI. If you're the person who selected **Export to BI** in Log Analytics, you probably don't need to worry about reviewing the data because you most likely are familiar with the data you're exporting.

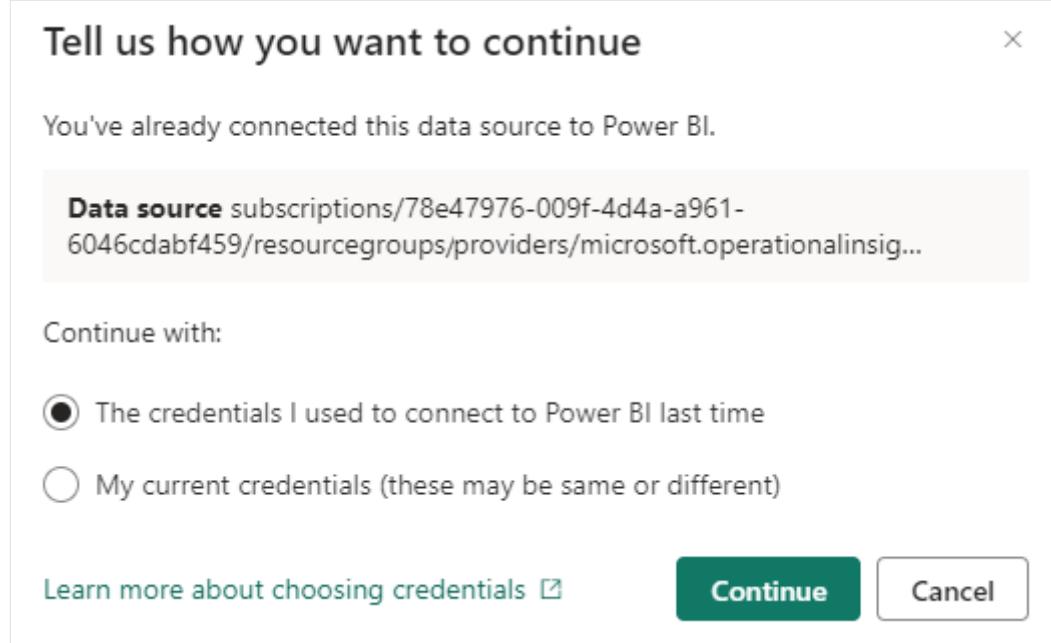
Reviewing the data is important if you weren't the one who exported the Log Analytics data, but rather received a link from someone for creating a dataset from Log Analytics. In such a case, you might not be familiar with the data that is being exported, and hence it's important to review it to make sure that no sensitive data is inadvertently being exposed.

Choosing which credentials to authenticate with

When you export data from Log Analytics to Power BI, Power BI connects to Log Analytics to get the data. In order to connect, it needs to authenticate with Log Analytics.

If you get the following dialog, it means that you've already established a connection to Log Analytics in the past. The credentials you used at that time may or may not be

different than the credentials of your current sign in. You need to choose whether to continue using the sign-in details you used the last time you connected (*The credentials I used to connect to Power BI last time*), or whether the connection should use your current sign-in credentials from now on (*My current credentials (these may be the same or different)*).



Why is this important?

The Power BI view of the Log Analytics data is determined by the permissions of the account used to establish the Power BI connection to the Log Analytics data source.

If you let Power BI use the sign-in details you used last time for the connection, the data you'll see in the dataset you're creating may differ from what you see in Log Analytics. This is because the data that is shown in the dataset is what the account with the credentials you used last time can see in Log Analytics.

If you replace the credentials you used last time with your current sign-in credentials, the data you see in the dataset you're creating will be exactly the same as what you see in Log Analytics. **However, since the connection now uses your current login credentials, views of the data in datasets you might have created previously from that Log Analytics query might also change, and this could affect reports and other downstream items that users might have created based on those datasets.**

Take the above considerations into account when you make your choice.

If you've never previously connected to Log Analytics from Power BI, Power BI will automatically use your current credentials to establish the connection, and you won't see this dialog.

Considerations and limitations

- This flow does not support business-to-business (B2B) scenarios or scenarios where authentication takes place against a service principal.
- If the Windows Azure Service Management API, the Log Analytics API service, or both, are configured to use multi-factor authentication, then in order for this flow to work, Power BI must also be configured to use multi-factor authentication. Consult your organization's IT support if you encounter a problem related to this consideration.

Next steps

- [Log Analytics integration with Power BI](#)
- [Dataset details](#)
- [Share access to a dataset](#)

Create a Power BI dataset directly from a SharePoint list

Article • 03/08/2023

You can quickly create a Power BI dataset directly from a SharePoint list. The dataset will be full-fledged Power BI dataset that you can use to create reports, analyze in Excel, and more.

Creating a dataset directly from a SharePoint list is an easy and quick way to share a dataset, because if you save it to a shared workspace, everyone with the sufficient permissions in the workspace can use it. You can also use [dataset sharing](#) to share it with other users who don't have a role in the workspace.

To keep the data fresh after you've created the dataset, either refresh the data manually or set up scheduled refresh.

This feature creates a dataset in the Power BI service directly from a SharePoint list. If you need to model or transform the data in ways that aren't available in the service, you can also connect to the SharePoint list from Power BI Desktop. For more information, see [Create a report on a SharePoint List in Power BI Desktop](#).

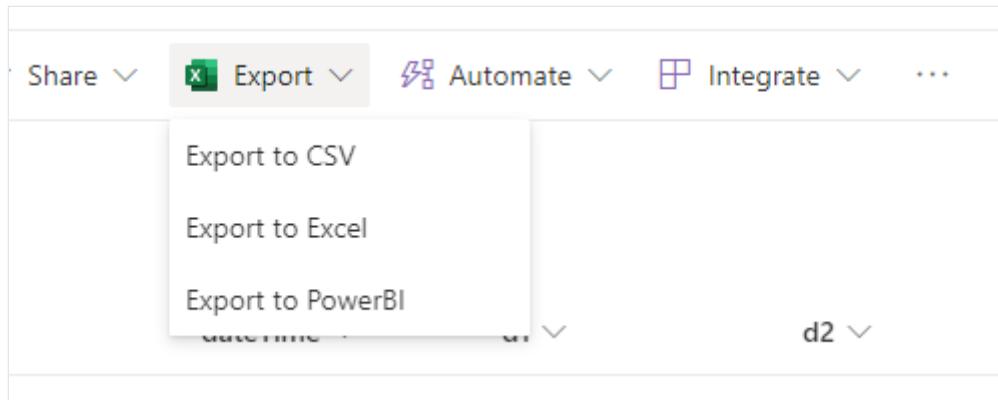
Prerequisites

You must have a Power BI account to be able to use this functionality.

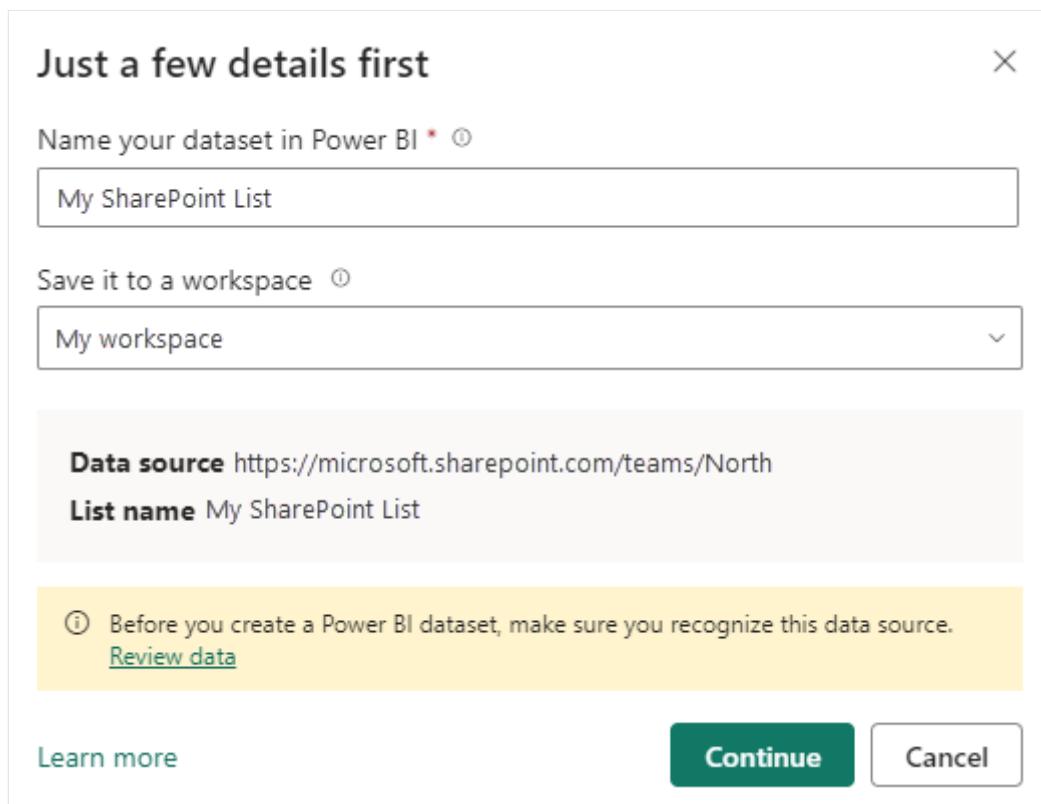
Create a dataset from a SharePoint list

To create a Power BI dataset from a SharePoint list:

1. Open your SharePoint list.
2. In the actions bar, select **Export > Export to Power BI**.



3. Power BI will open and a dialog will ask you to name the dataset and choose a workspace to save it in. By default the dataset will be given the same name as the SharePoint list and saved to My workspace. You can choose your own name and destination workspace. If you're a [free user](#) in Power BI, you'll only be able to save to My workspace.



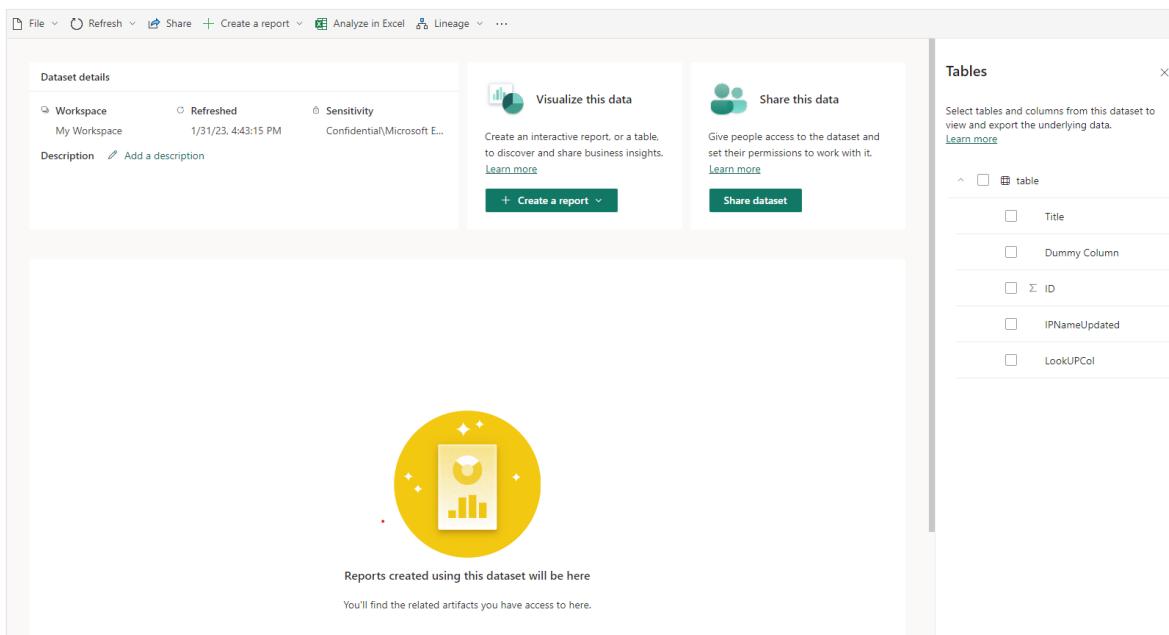
The dialog also shows the URL of the data source (SharePoint site) and name of the SharePoint list. To prevent inadvertently exposing sensitive data, make sure that you recognize the data source and are familiar with the data. Select **Review data** if you want to check the SharePoint list before allowing export to continue. For more information about when reviewing the data might be a good idea, see [Reviewing the SharePoint list data](#).

4. Select **Continue**. Your dataset will be created, and you'll be taken to the details page of the new dataset. From there you can do all the things you can do with a

regular Power BI dataset - refresh the data, share the dataset, create new reports, and more. See [dataset details](#) for more information.

ⓘ Note

If you've connected to the SharePoint site from Power BI before, you'll be asked to choose which credentials to use to for the connection between Power BI and the Sharepoint site before being taken to the dataset details page. For help deciding which credentials to choose, see [Choosing which credentials to authenticate with](#).



To keep the data fresh after you've created the dataset, either refresh the data manually or set up scheduled refresh.

Reviewing the SharePoint list data

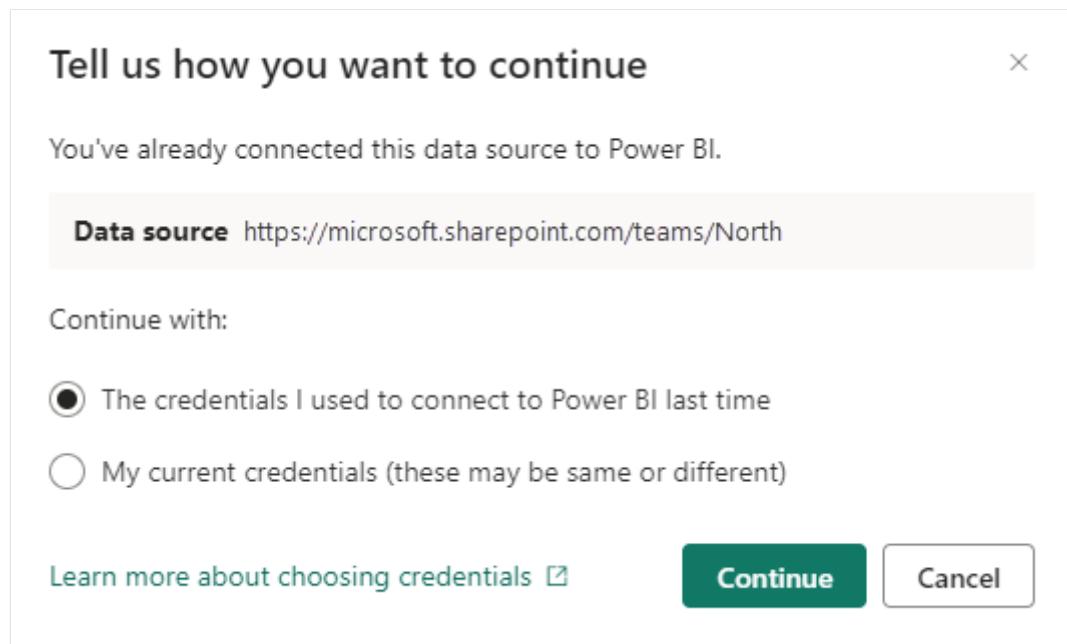
When you export a SharePoint list to Power BI, a redirect URL is created that includes all the parameters needed to launch the create dataset process in Power BI. If you're the person who selected **Export to BI** in your SharePoint list, you probably don't need to worry about reviewing the data because you most likely are familiar with the data you're exporting.

Reviewing the data is important if you weren't the one who exported the SharePoint list, but rather received a link from someone for creating a dataset from a SharePoint list. In such a case, you might not be familiar with the data that is being exported, and hence it's important to review it to make sure that no sensitive data is inadvertently being exposed.

Choosing which credentials to authenticate with

When you export a SharePoint list to Power BI, Power BI connects to the SharePoint site to get the data from the list. In order to connect, it needs to authenticate with SharePoint.

If you get the following dialog, it means that you've already established a connection to the SharePoint site in the past. The credentials you used at that time may or may not be different than the credentials of your current sign in. You need to choose whether to continue using the sign-in details you used the last time you connected (*The credentials I used to connect to Power BI last time*), or whether the connection should use your current sign-in credentials from now on (*My current credentials (these may be the same or different)*).



Why is this important?

The Power BI view of the SharePoint list data is determined by the permissions of the account used to establish the Power BI connection to the SharePoint data source (that is, the SharePoint site).

If you let Power BI use the sign-in details you used last time for the connection, the data you'll see in the dataset you're creating may differ from what you see in the SharePoint list. This is because the data that is shown in the dataset is what the account with the credentials you used last time can see in the SharePoint list.

If you replace the credentials you used last time with your current sign-in credentials, the data you see in the dataset you're creating will be exactly the same as what you see

in the SharePoint list. However, since the connection now uses your current login credentials, views of the data in datasets you might have created previously from that SharePoint site might also change, and this could affect reports and other downstream items that users might have created based on those datasets.

Take the above considerations into account when you make your choice.

If you've never previously connected to the SharePoint site from Power BI, Power BI will automatically use your current credentials to establish the connection, and you won't see this dialog.

Considerations and limitations

- The dataset won't be created if the SharePoint list contains values with more than four digits after a decimal place (".")
- The sensitivity label (if any) of the SharePoint list isn't inherited by the dataset that is created.
- This flow does not support business-to-business (B2B) scenarios or scenarios where authentication takes place against a service principal.
- If the SharePoint service is configured to use multi-factor authentication, then in order for this flow to work, Power BI must also be configured to use multi-factor authentication. Consult your organization's IT support if you encounter a problem related to this consideration.

Next steps

- [Dataset details](#)
- [Share access to a dataset](#)

Create a report on a SharePoint List in Power BI Desktop

Article • 02/22/2023

Many teams and organizations use lists in SharePoint Online to store data because it's easy to set up and easy for users to update. Sometimes a chart is a much easier way for users to quickly understand the data rather than looking at the list itself. In this tutorial, you learn how to transform your SharePoint list data into a Power BI report.

Watch this five-minute tutorial video, or scroll down for step-by-step instructions.

Note

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

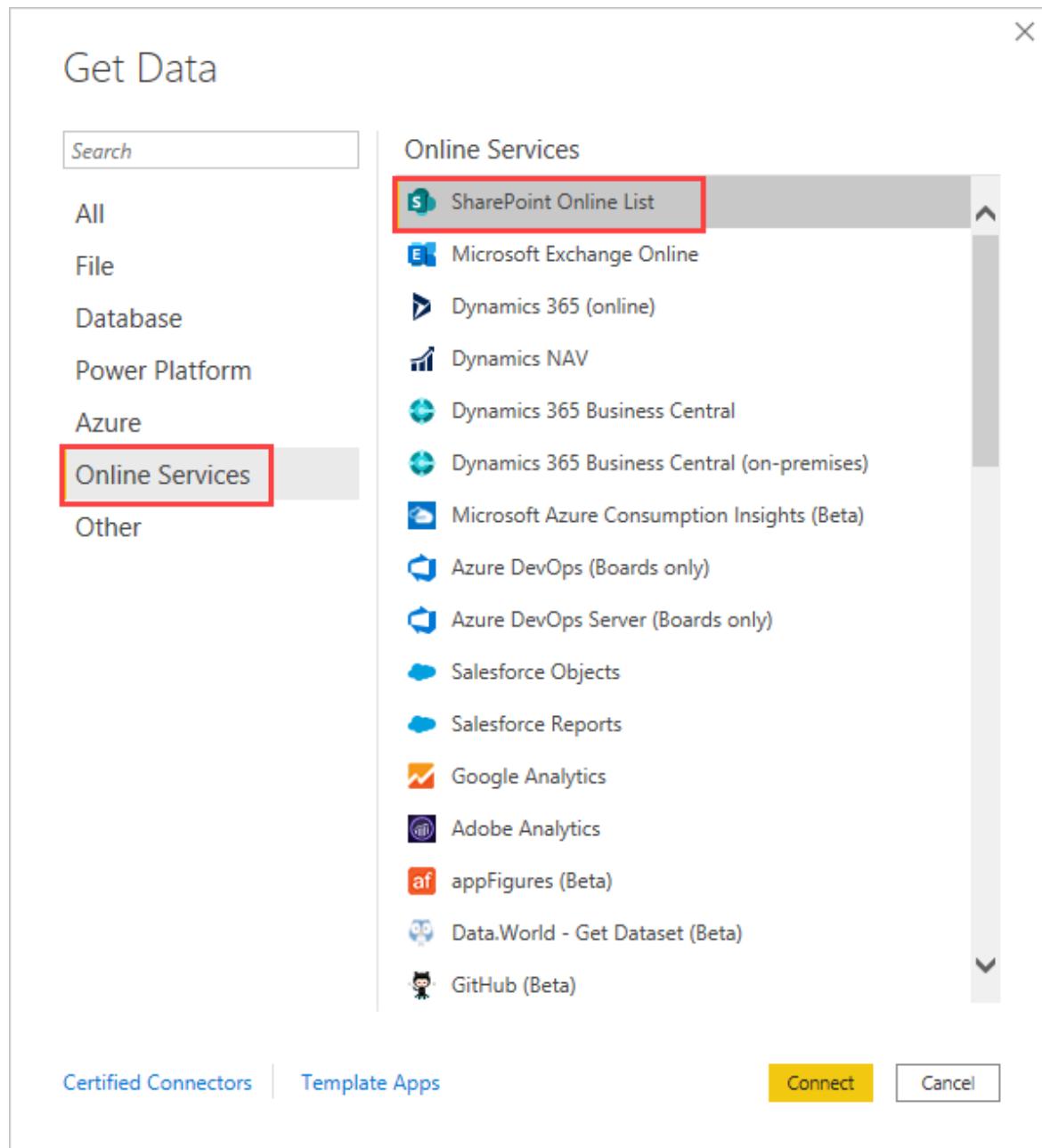


In the Power BI service, you can also [create a report quickly from data in a SharePoint list](#).

If your purpose is to quickly create a dataset in the Power BI service, you can do so directly from the SharePoint list. For more information, see [Create a dataset from a SharePoint list](#).

Part 1: Connect to your SharePoint List

1. If you don't have it already, download and install [Power BI Desktop](#).
2. Open Power BI Desktop and in the **Home** tab of the ribbon, select **Get data > More**.
3. Select **Online Services**, and then select **SharePoint Online List**.

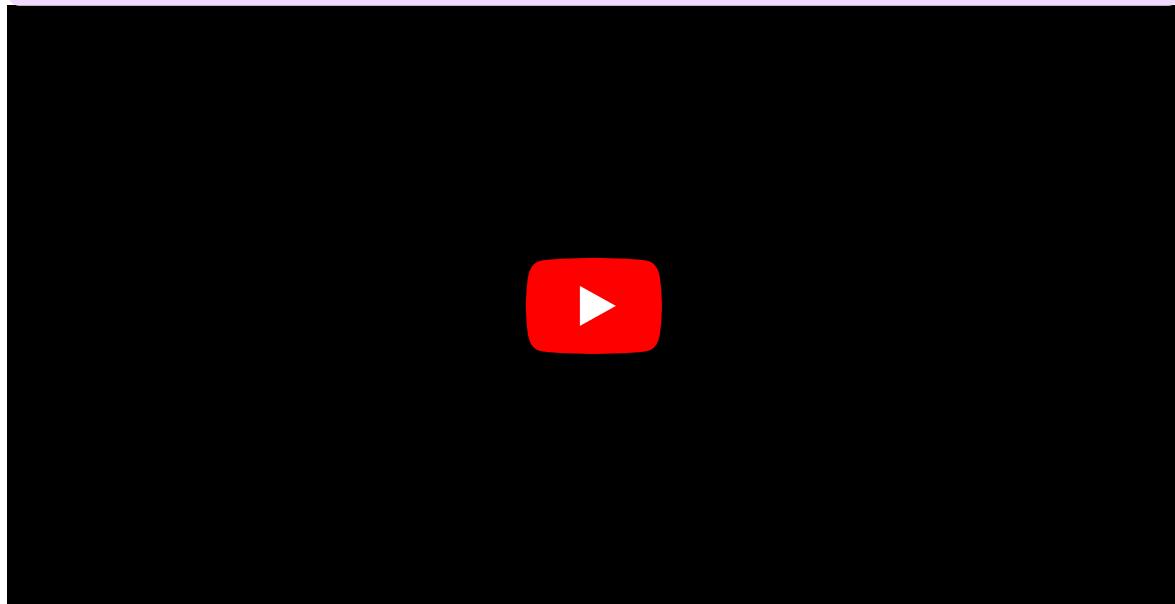


4. Select **Connect**.
5. Find the address (also known as a URL) of your SharePoint Online site that contains your list. From a page in SharePoint Online, you can usually get the site address by selecting **Home** in the navigation pane, or the icon for the site at the top, then copying the address from your web browser's address bar.

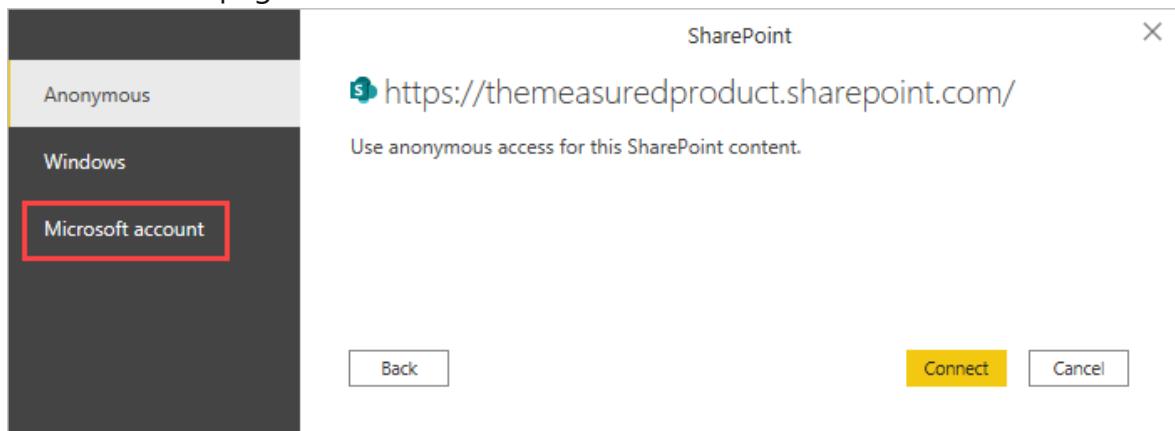
Watch a video of this step:

 **Note**

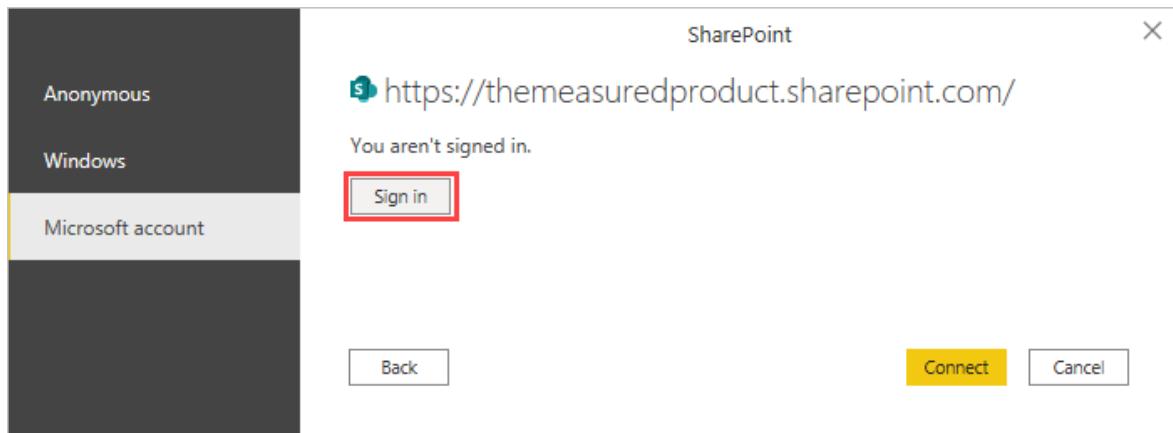
This video might use earlier versions of Power BI Desktop or the Power BI service.



6. In Power BI Desktop, paste the address into the **Site URL** field of the **SharePoint Online Lists** dialog box, and then select **OK**.
7. You might or might not see a SharePoint access screen like the following image. If you don't see it, skip to step 10. If you do see it, select **Microsoft Account** on the left side of the page.

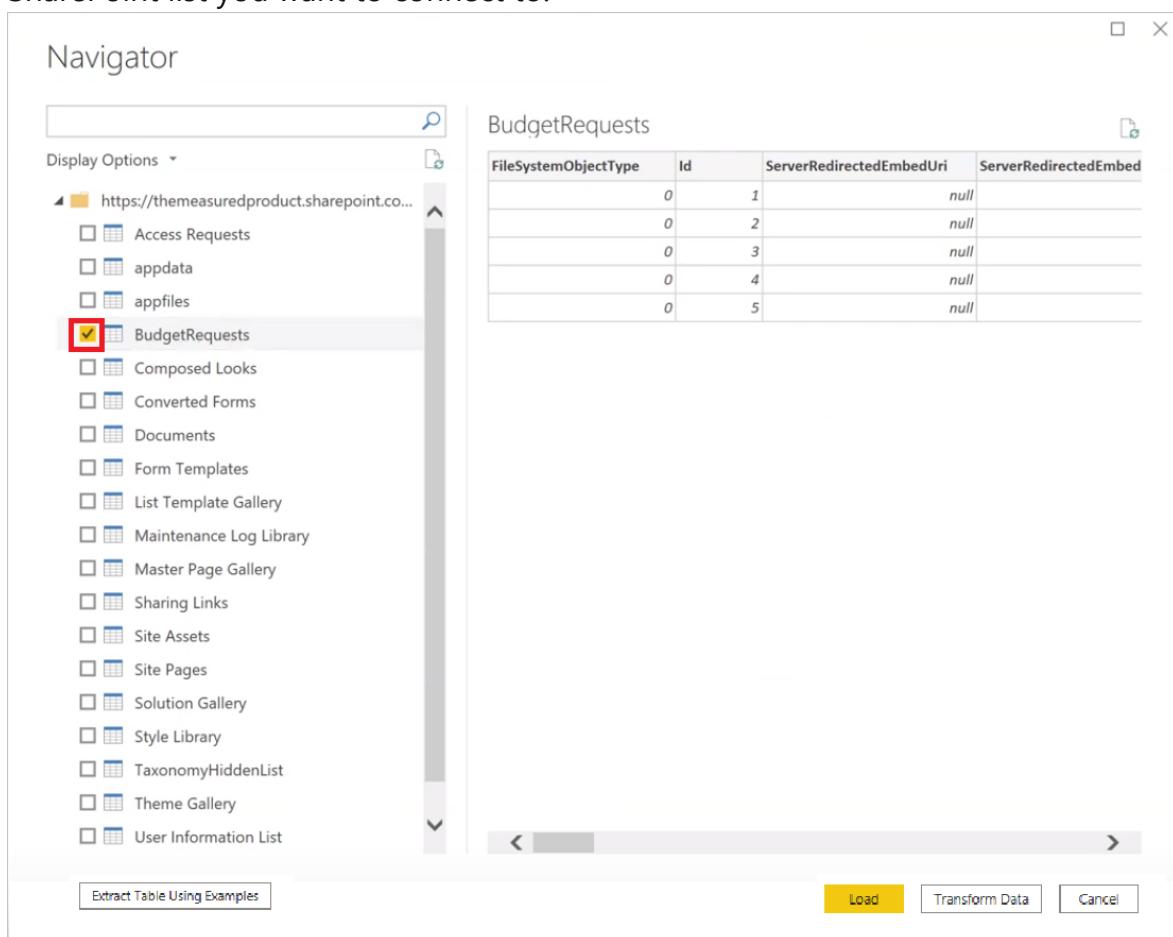


8. Select **Sign In**, and enter the user name and password you use to sign in to Microsoft 365.



9. When you finish signing in, select **Connect**.

10. On the left side of the **Navigator** dialog box, select the checkbox beside the SharePoint list you want to connect to.



A screenshot of the Power BI Navigator dialog box. The title bar says 'Navigator'. On the left, there is a tree view of SharePoint sites and lists. The 'BudgetRequests' list under the root site is selected, indicated by a checked checkbox which is highlighted with a red box. On the right, there is a table titled 'BudgetRequests' with four columns: 'FileSystemObjectType', 'Id', 'ServerRedirectedEmbedUri', and 'ServerRedirectedEmbed'. The table has five rows, all of which have 'null' in the 'ServerRedirectedEmbedUri' and 'ServerRedirectedEmbed' columns. At the bottom of the dialog box are buttons for 'Extract Table Using Examples', 'Load' (which is yellow), 'Transform Data', and 'Cancel'.

FileSystemObjectType	Id	ServerRedirectedEmbedUri	ServerRedirectedEmbed
0	1	null	null
0	2	null	null
0	3	null	null
0	4	null	null
0	5	null	null

11. Select **Load**. Power BI loads your list data into a new report.

Part 2: Create a report

1. On the left side of the Power BI Desktop screen, select the **Data** icon to see that your SharePoint list data was loaded.
2. Make sure your list columns with numbers show the Sum, or Sigma, icon in the **Data** pane on the right. For any that don't, select the column header in the table

view, select the **Structure** group in the **Column tools** tab, then change the **Data type** to **Decimal Number** or **Whole Number**, depending on the data. If prompted to confirm your change, select **Yes**. If your number is a special format, like currency, you can also choose that by setting the **Format** in the **Formatting** group.

Watch a video of this step:

 **Note**

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



3. On the left side of the Power BI Desktop screen, select the **Report** icon.
4. Select columns you want to visualize by selecting the checkboxes beside them in the **Fields** pane on the right.

Watch a video of this step:

 **Note**

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



5. Change the visual type if you need to.
6. You can create multiple visualizations in the same report by deselecting the existing visual, then selecting checkboxes for other columns in the **Fields** pane.
7. Select **Save** to save your report.

Next steps

- [Create a report quickly from a SharePoint list](#)

Connect to LinkedIn Sales Navigator in Power BI Desktop

Article • 03/20/2023

In Power BI Desktop, you can connect to *LinkedIn Sales Navigator* to help find and build relationships just like any other data source in Power BI Desktop, and create ready-made reports about your progress.



To connect to LinkedIn data using the **LinkedIn Sales Navigator**, you need to have a LinkedIn Sales Navigator Enterprise plan, and either be an Admin or Reporting User on the Sales Navigator Contract.

The following video provides a quick tour and tutorial for using the **LinkedIn Sales Navigator** template app, which is described in detail [later in this article](#).

ⓘ Note

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

<https://www.youtube-nocookie.com/embed/ZqhmaiORLw0>

Connect to LinkedIn Sales Navigator

To connect to LinkedIn Sales Navigator data, follow the instructions in the [Power Query LinkedIn Sales Navigator article](#).

Using the LinkedIn Sales Navigator template app

To make using the LinkedIn Sales Navigator as easy as possible, you can use the [template app](#) that automatically creates a ready-made report from your LinkedIn Sales Navigator data.

The screenshot shows the Microsoft AppSource page for the "LinkedIn Sales Navigator Analytics Integration" app. At the top, there's a navigation bar with "Microsoft", "AppSource", "Apps", "Consulting Services", a search bar, and user profile icons. Below the header, the app's title is displayed with a blue "GET IT NOW" button. To the left, there's a sidebar with links for "Pricing Free", "Products Power BI apps", "Publisher Microsoft", "Acquire Using Work or school account", "Categories Analytics Marketing Sales", "Support Support Help", and "Legal License Agreement Privacy Policy". The main content area features a large LinkedIn logo, a brief overview, and a detailed description of how the app integrates LinkedIn Sales Navigator data with Power BI. It also includes a screenshot of the Power BI dashboard showing various sales metrics like "Overall Performance & Active Users by Date" and "Overall Leads by Date". A "Reviews" tab is visible at the top of this section.

When you download the app, you can select whether to connect to your data, or explore the app with sample data. You can always go back and connect to your own LinkedIn Sales Navigator data after you explore the sample data.

This screenshot shows the "Get started with your new app" screen. It has three main sections: "Connect your data" (with an icon of a database and cloud, and a "Connect" button), "Explore with sample data" (with an icon of gears and a magnifying glass, and a "Explore app" button), and "Customize and share" (with an icon of a workspace with multiple windows, and an "Edit workspace" button). A "Don't show this again" link is located at the bottom right. A large "Get started with your new app" heading is centered at the top, and a descriptive text block is located between the first two sections.

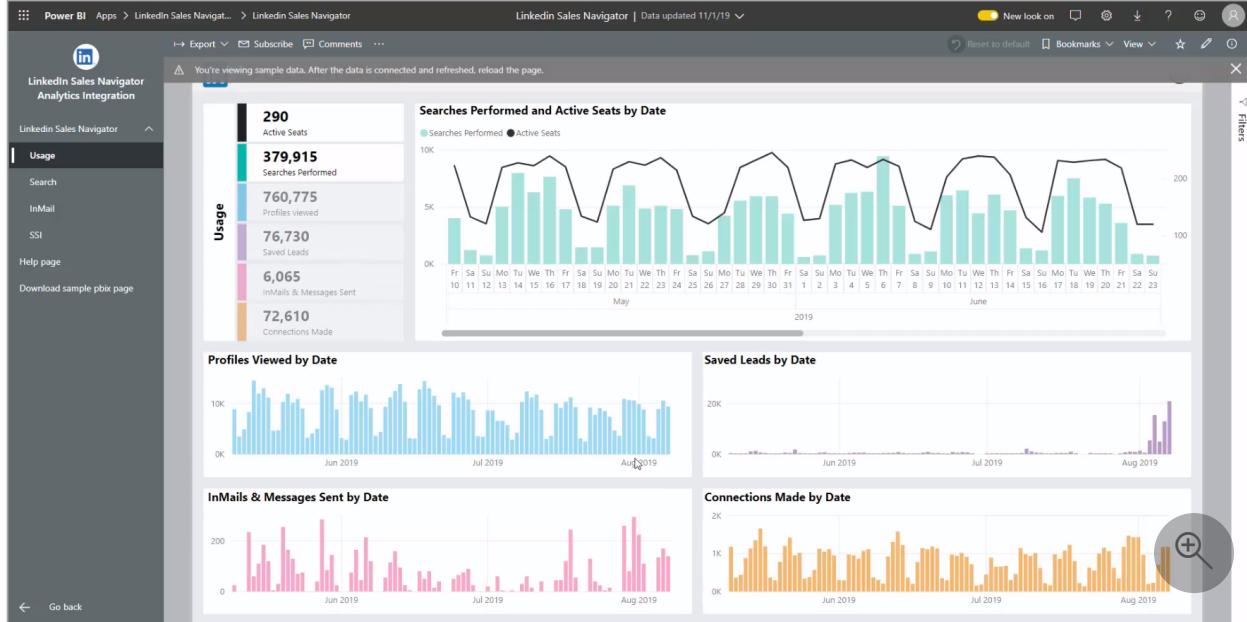
You can get the [LinkedIn Sales Navigator template app](#) from the following link:

- [LinkedIn Sales Navigator template app ↗](#)

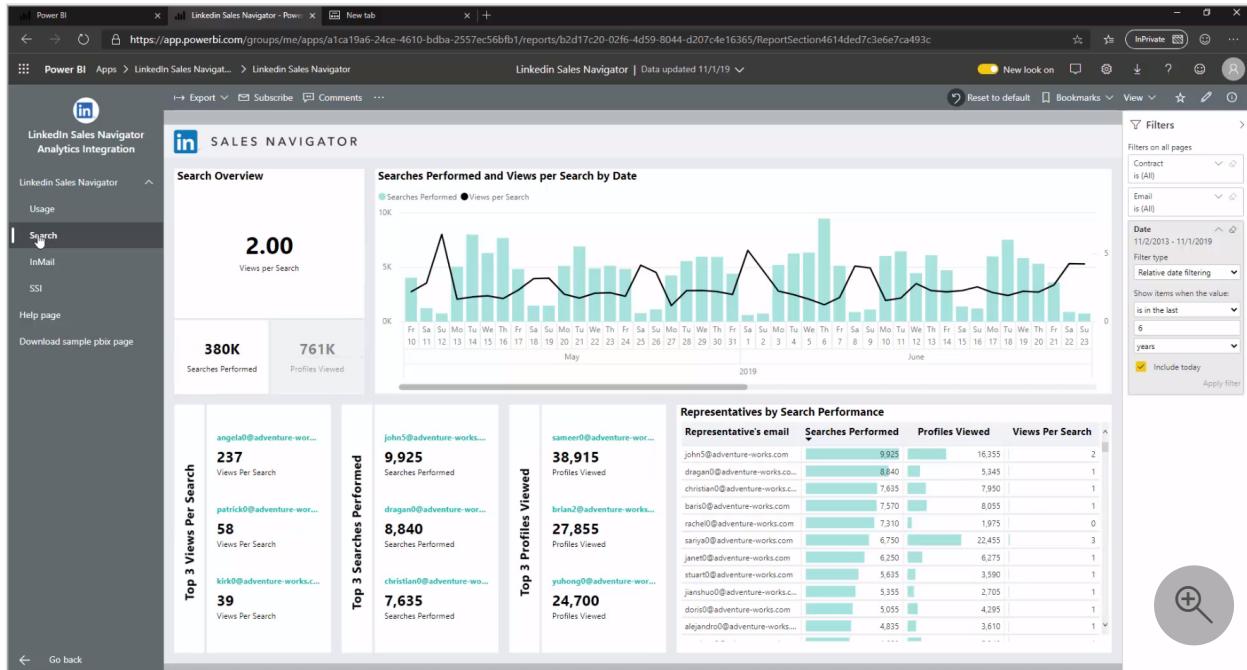
The template app provides four tabs to help analyze and share your information:

- Usage
- Search
- InMail
- SSI

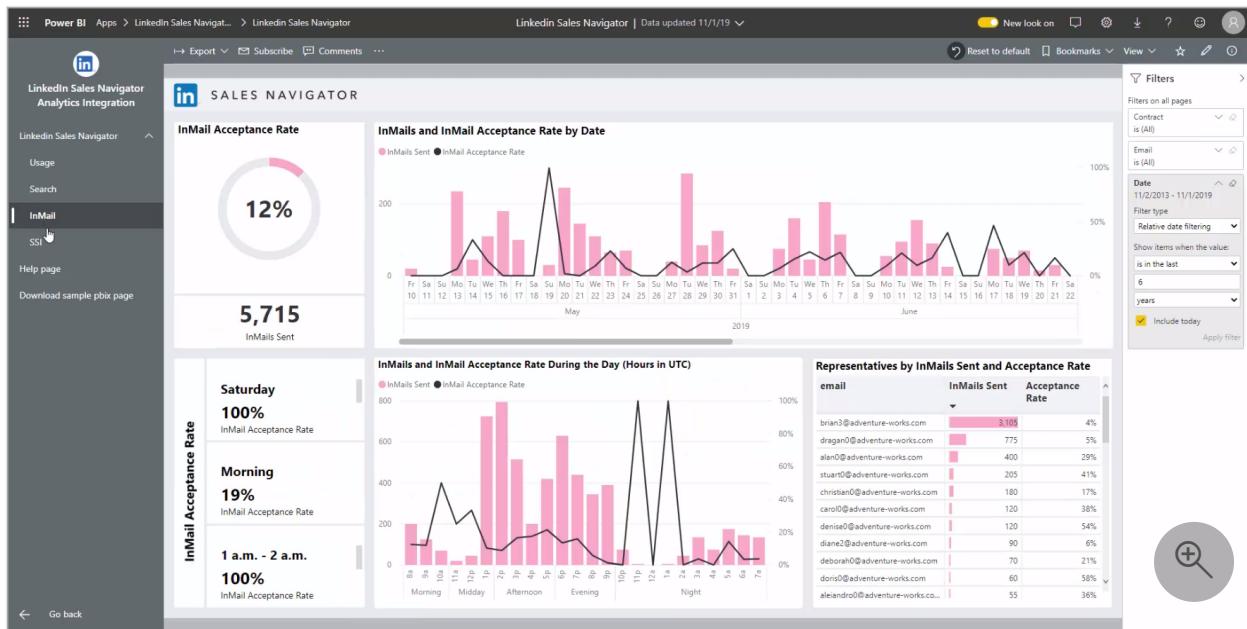
The **Usage** tab shows your overall LinkedIn Sales Navigator data.



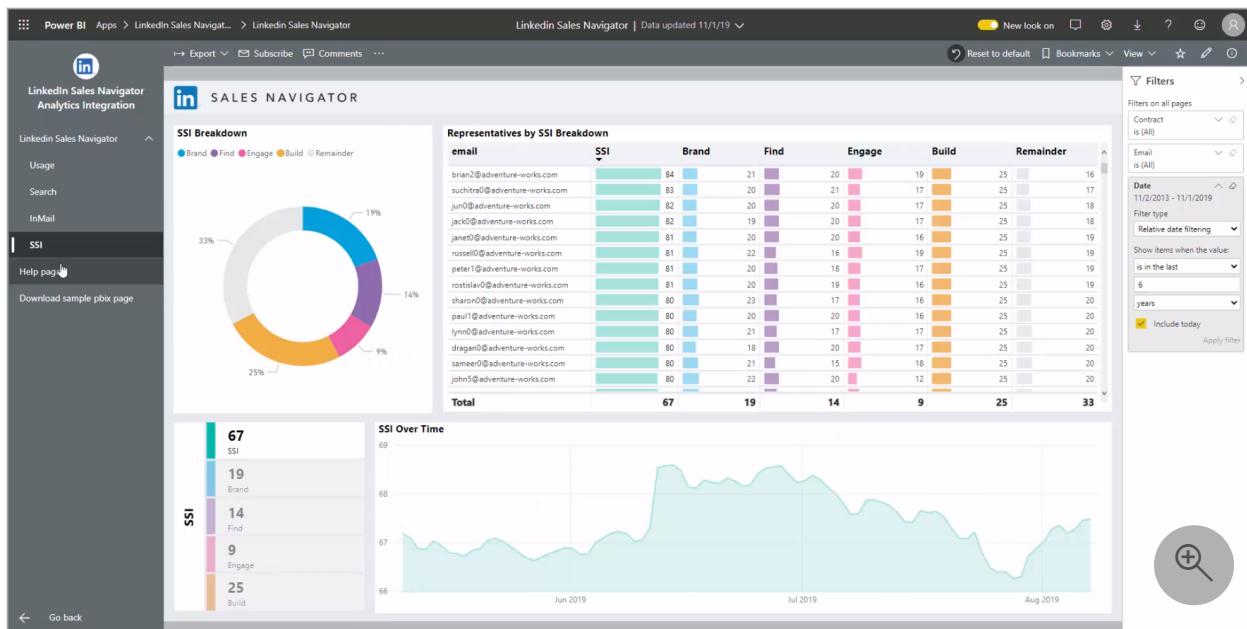
The **Search** tab lets you drill deeper into your search results.



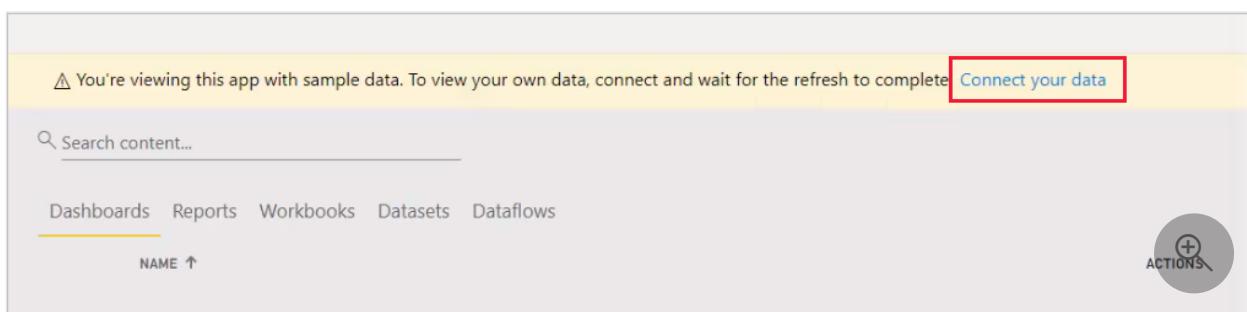
The **InMail** tab provides insights into your InMail usage, including number of InMails sent, acceptance rates, and other useful information.



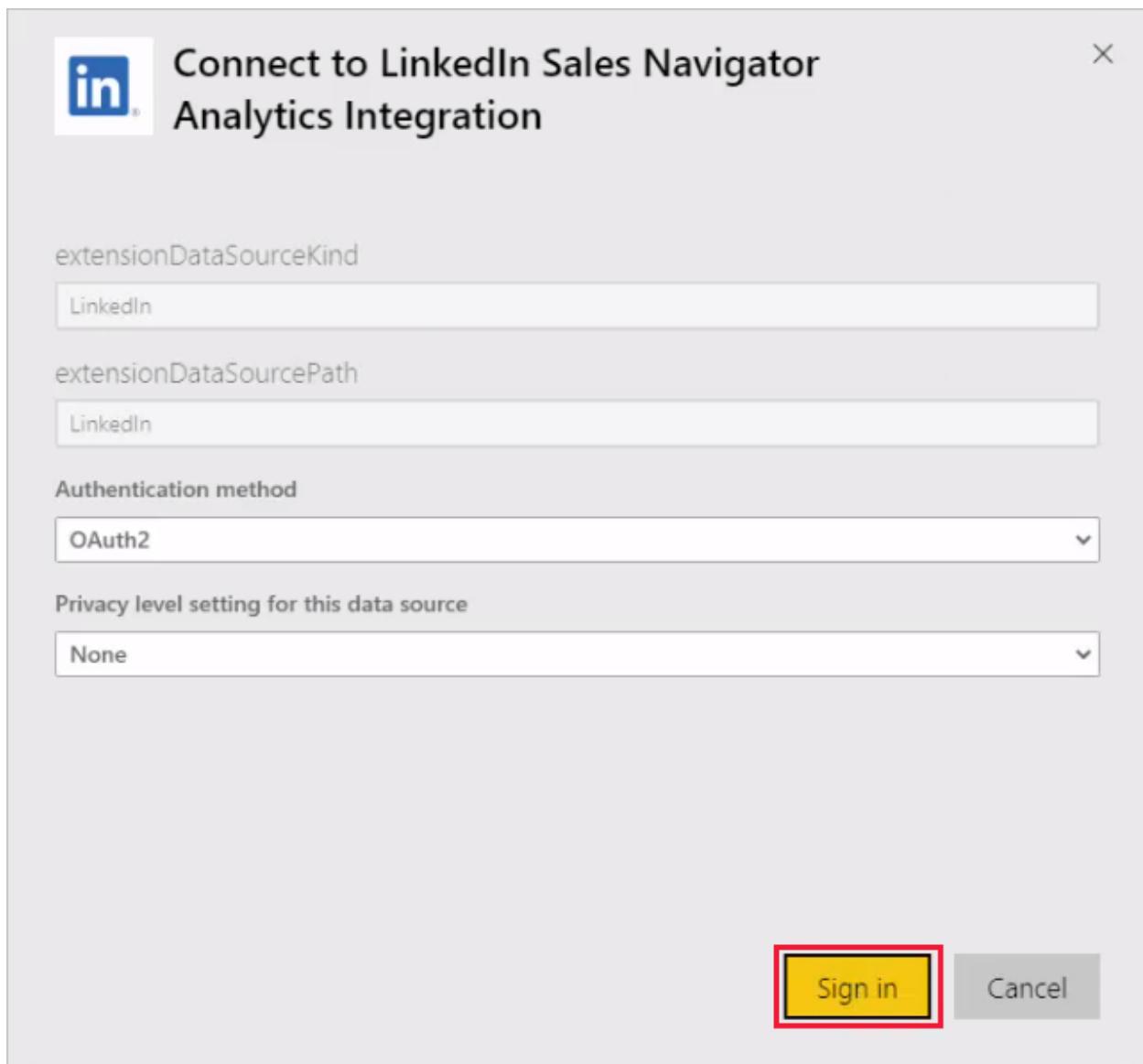
The **SSI** tab provides more details into your social selling index (SSI).



To go from the sample data to your own data, select **edit app** in the top-right corner (the pencil icon) and then select **Connect your data** from the screen that appears.



From there you can connect your own data, selecting how many days of data to load. You can load up to 365 days of data. You need to sign in, again using the same email address you use to sign in to LinkedIn Sales Navigator through the website.



The template app then refreshes the data in the app with your data. You can also set up a scheduled refresh, so the data in your app is as current as your refresh frequency specifies.

Once the data updates, you can see the app populated with your own data.

Getting help

If you run into problems when connecting to your data, you can contact [LinkedIn Sales Navigator support](#).

Next steps

There are all sorts of data you can connect to using Power BI Desktop. For more information on data sources, check out the following resources:

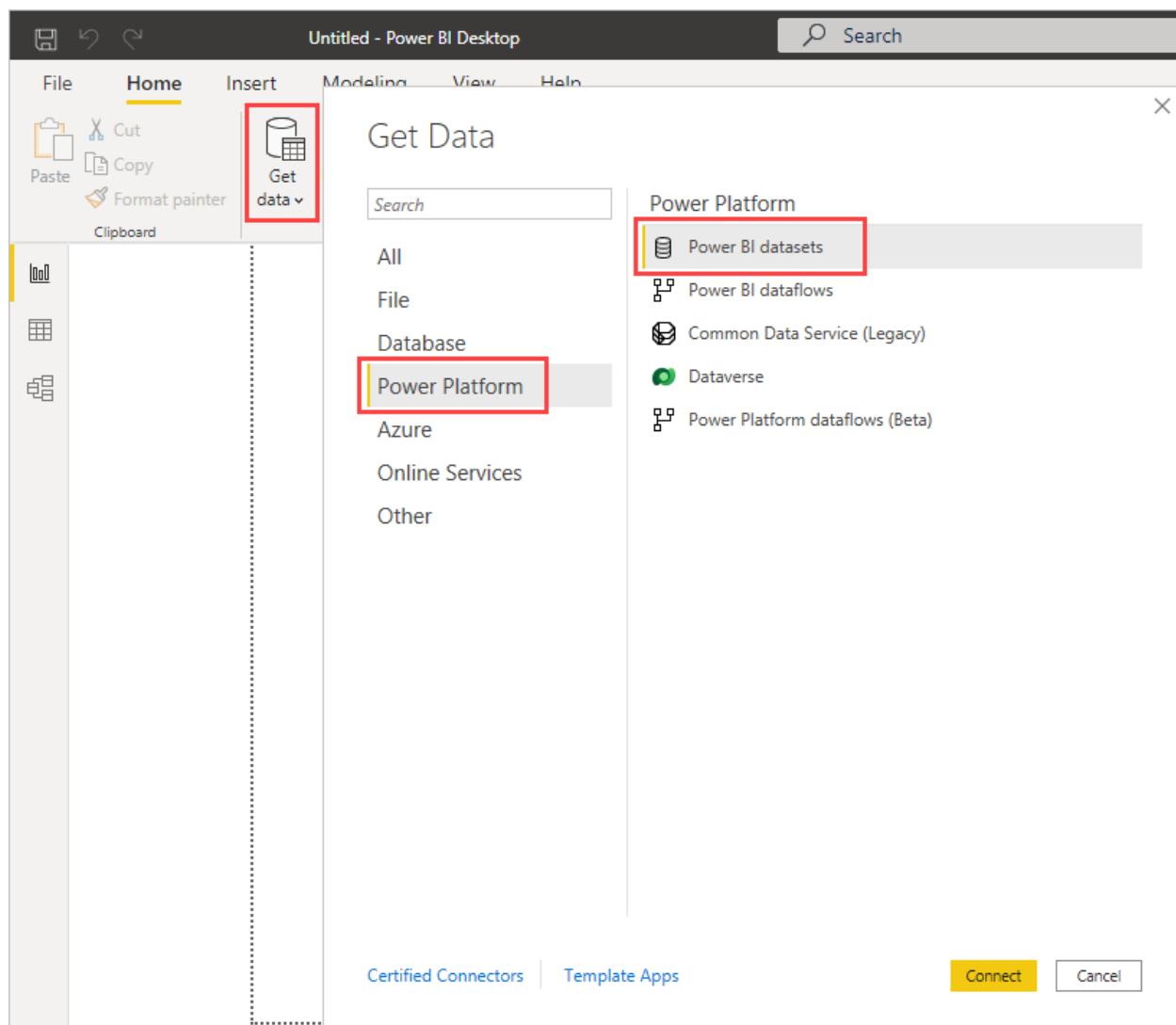
- [What is Power BI Desktop?](#)

- Data Sources in Power BI Desktop
- Shape and Combine Data with Power BI Desktop
- Connect to Excel workbooks in Power BI Desktop
- Enter data directly into Power BI Desktop

Connect to datasets in the Power BI service from Power BI Desktop

Article • 07/17/2023

In Power BI Desktop, you can create a data model and publish it to the Power BI service. Then you and others can establish a live connection to the shared dataset that's in the Power BI service, and create many different reports from that common data model. You can use the *Power BI service live connection* feature to create multiple reports in *.pbix* files from the same dataset, and save them to different workspaces.



This article discusses the benefits, best practices, considerations, and limitations of the Power BI service live connection feature.

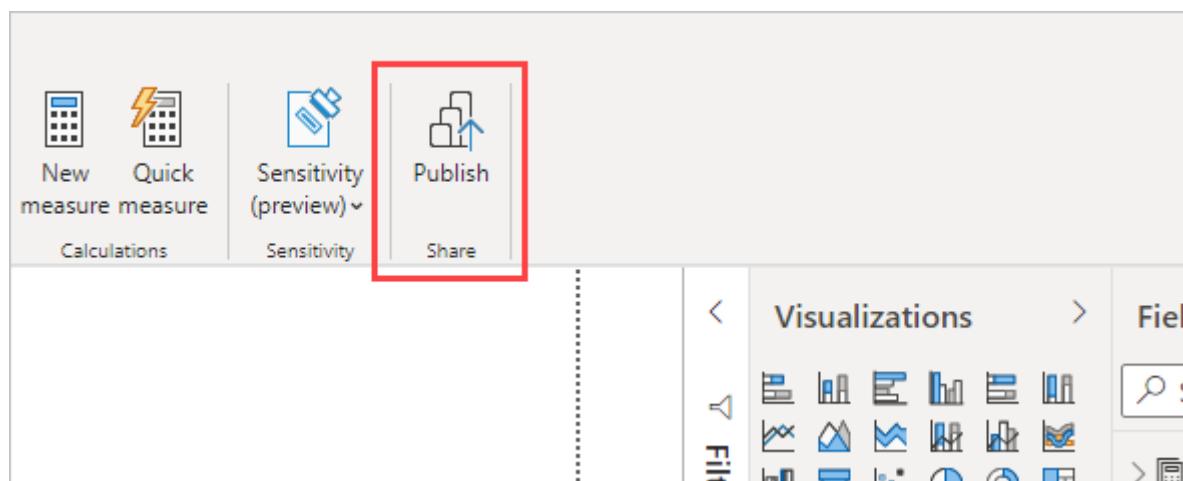
Power BI live connection and report lifecycle management

One challenge with the popularity of Power BI is the resulting proliferation of reports, dashboards, and underlying data models. It's easy to create compelling reports in Power BI Desktop, [publish](#) those reports in the Power BI service, and create great dashboards from those datasets.

Because report creators often use the same or nearly the same datasets, knowing which dataset a report is based on and the freshness of that dataset becomes a challenge. The Power BI service live connection addresses that challenge by using common datasets to make it easier and more consistent to create, share, and expand on reports and dashboards.

Create and share a dataset everyone can use

A business analyst on your team who is skilled at creating good data models, also called datasets, can create a dataset and report and then share that report in the Power BI service.



If everyone on the team created their own versions of the dataset and shared their reports with the team, there would be many reports from different datasets in your team's Power BI workspace. It would be hard to tell which report was the most recent, whether the datasets were the same, or what the differences were.

With the Power BI service live connection feature, other team members can use the analyst's published dataset for their own reports in their own workspaces. Everyone can use the same solid, vetted, published dataset to build their own unique reports.

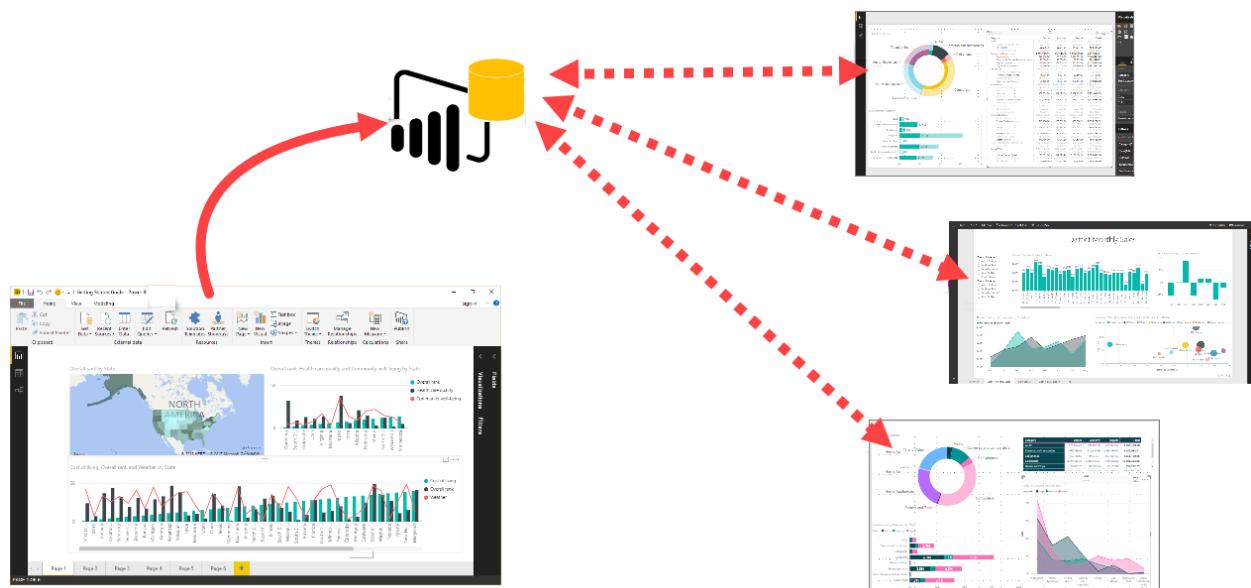
Connect to the dataset by using a Power BI service live connection

In Power BI Desktop, the team business analyst creates a report and the dataset the report is based on. The analyst then publishes the report to the Power BI service, and the

report shows up in the team's workspace. For more information about workspaces, see [Workspaces in Power BI](#).

The business analyst can use the Build permission setting to make the report available for anyone in or out of the workspace to see and use. Team members in and out of the team workspace can now establish a live connection to the shared data model by using the Power BI service live connection feature. Team members can create their own unique reports, from the original dataset, in their own workspaces.

The following image shows how one Power BI Desktop report and its data model publish to the Power BI service. Other users connect to the data model by using the Power BI service live connection, and base their own unique reports in their own workspaces on the shared dataset.



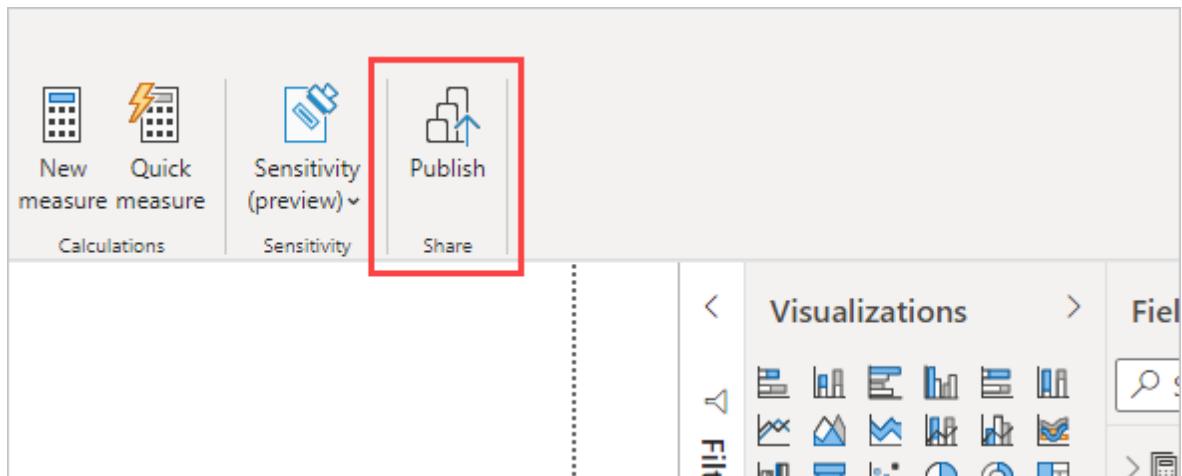
Set up and use a Power BI service live connection

You can see the usefulness of the Power BI service live connection for report lifecycle management. Now find out how to get from a great report and dataset to a shared dataset that teammates can use in Power BI.

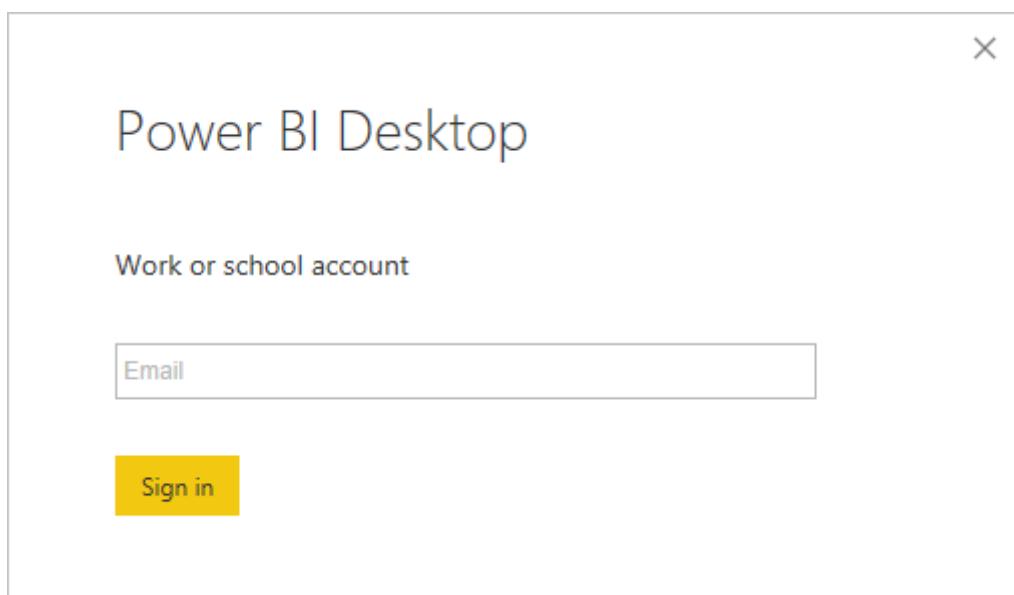
Publish a Power BI report and dataset

The first step in using a Power BI service live connection to manage report lifecycle is to publish a report and dataset for teammates to use.

1. To publish the report, from Power BI Desktop, select **Publish** from the **Home** tab.



If you're not signed in to the Power BI service account, Power BI prompts you to sign in.



2. Select the workspace destination to publish the report and dataset to, and choose **Select**. Anyone who has Build permission can then access that dataset. You can set Build permission in the Power BI service after publishing.

Publish to Power BI

Select a destination

My workspace

Select

Cancel

The publishing process begins, and Power BI Desktop shows the progress.

Publishing to Power BI

 Publishing 'Report Lifecycle Best Practices.pbix' to Power BI



Go to the "view" tab, select "phone reports" to tailor a portrait view of your existing report for mobile viewers. [Learn more](#)

Cancel

Once complete, Power BI Desktop shows success, and provides links to the report in the Power BI service and to quick insights about the report.

Publishing to Power BI

 Success!

[Open 'Report Lifecycle Best Practices.pbix' in Power BI](#)

[Get Quick Insights](#)



Go to the "view" tab, select "phone reports" to tailor a portrait view of your existing report for mobile viewers. [Learn more](#)

Got it

3. Now that your report with its dataset is in the Power BI service, you can *promote* it, or attest to its quality and reliability. You can also request for the report to be *certified* by a central authority in your Power BI tenant. For more information, see [Endorse your content](#).

4. The last step is to set Build permission in the Power BI service for the dataset the report is based on. Build permission determines who can see and use your dataset. You can set Build permission in the workspace itself, or when you share an app from the workspace. For more information, see [Build permission for shared datasets](#).

Establish a Power BI service live connection to the published dataset

Teammates who have access to the workspace where the report and dataset were published can connect to the dataset and build their own reports. To establish a connection to a published report and create your own report based on the published dataset:

1. In Power BI Desktop, on the **Home** tab, select **Get data > Power BI datasets**.

Or, select **Get data**, and on the **Get Data** screen, select **Power Platform** in the left pane, select **Power BI datasets**, and then select **Connect**.

If you're not signed in, Power BI prompts you to sign in.

2. The **Data hub** shows the workspaces you're a member of, and all the shared datasets you have Build permission for in any workspace.

To find the dataset you want, you can:

- Filter the list to **My data** or datasets that are **Endorsed in your org**.
- Search for a specific dataset or filter by keyword.
- See dataset name, owner, workspace, last and next refresh time, and sensitivity.

Data hub

Discover data from across your org and use it to create reports. [Learn more](#)

All My data Endorsed in your org

lifecycle Filter(1)

Name	Endorsement	Owner	Workspace	Refreshed	Sensitivity
Power BI Lifecycle Dataset	Certified	Jacob Knight...	Power BI Lifecycle D...	1/5/23, 11:59:20 PM	Confidential\Mi...

Connect Cancel

3. Select a dataset, and then select **Connect** to establish a live connection to the selected dataset. Power BI Desktop loads the dataset fields and their values in real time.

Visualizations > Fields >

Search

- > Measures for Custom ...
- > Shared Datasets
- > Account
- > Azure Connected Ten...
- > Azure CR for DB, AI a...
- > Azure Direct Query
- > Azure Must Win
- > Billed Revenue
- > Calendar
- > Calendar (Expirations)
- > Dataflows Direct
- > Dataflows Downstream
- > ERC_RI Sessions

Now you and others can create and share custom reports, all from the same dataset. This approach is a great way to have one knowledgeable person create a well-formed dataset. Many teammates can use that shared dataset to create their own reports.

Considerations and limitations

When you use the Power BI service live connection, keep a few considerations and limitations in mind.

- Only users with Build permission for a dataset can connect to a published dataset by using the Power BI service live connection.
- Hidden columns will become visible to users with Build permissions when they create live connections to the dataset in Power BI Desktop.
- Free users only see datasets that are in their **My Workspace** and in Premium-based workspaces.
- Because this connection is live, left navigation and modeling are disabled. The behavior is similar to a SQL Server Analysis Services (SSAS) connection. However, composite models in Power BI make it possible to combine data from different sources. For more information, see [Use composite models in Power BI Desktop](#).
- Because this connection is live, row-level security (RLS) and similar connection behaviors are enforced. This behavior is the same as when connected to SSAS.
- If the owner modifies the original shared .pbix file, the shared dataset and report in the Power BI service are overwritten. Reports based on the dataset aren't overwritten, but any changes to the dataset reflect in the report.
- Members of a workspace can't replace the original shared report. If they try to do so, they get a prompt to rename the file and publish.
- If you delete the shared dataset in the Power BI service, reports based on that dataset will no longer work properly or display visuals. You can no longer access that dataset from Power BI Desktop.
- Reports that share a dataset on the Power BI service don't support automated deployments that use the Power BI REST API.

Next steps

For more information on DirectQuery and other Power BI data connection features, check out the following resources:

- [Use DirectQuery in Power BI](#)
- [Data sources supported by DirectQuery](#)
- [Using DirectQuery for Power BI datasets and Azure Analysis Services \(preview\)](#)

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

- [What is Power BI Desktop?](#)
- [Query overview with Power BI Desktop](#)
- [Data types in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)

- Publish datasets and reports from Power BI Desktop

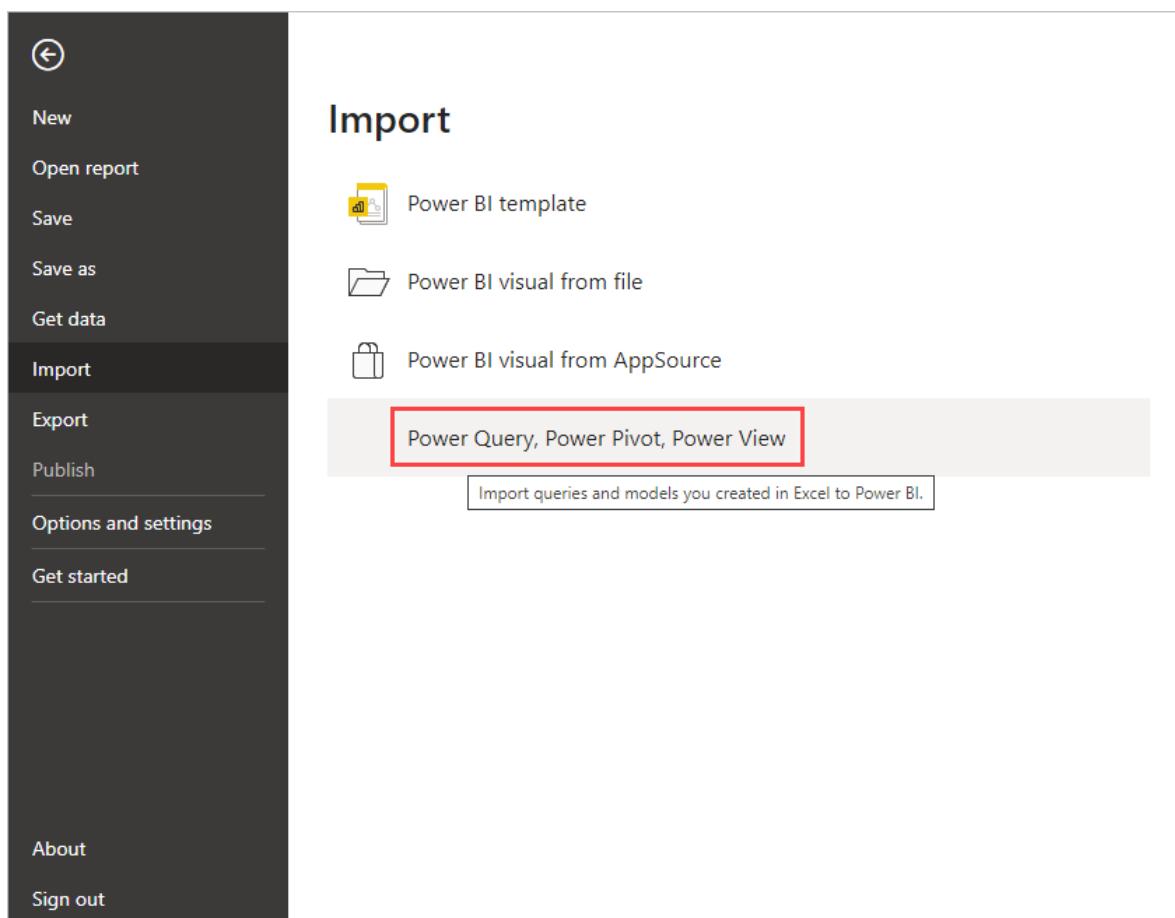
Import Excel workbooks into Power BI Desktop

Article • 01/09/2023

With Power BI Desktop, you can easily import Excel workbooks that contain Power Query queries and Power Pivot models into Power BI Desktop. Power BI Desktop automatically creates reports and visualizations based on the Excel workbook. Once imported, you can continue to improve and refine those reports with Power BI Desktop, using the existing features and new features released with each Power BI Desktop monthly update.

Import an Excel workbook

1. To import an Excel workbook into Power BI Desktop, select **File > Import > Power Query, Power Pivot, Power View**.



2. From the **Open** window, select an Excel workbook to import.

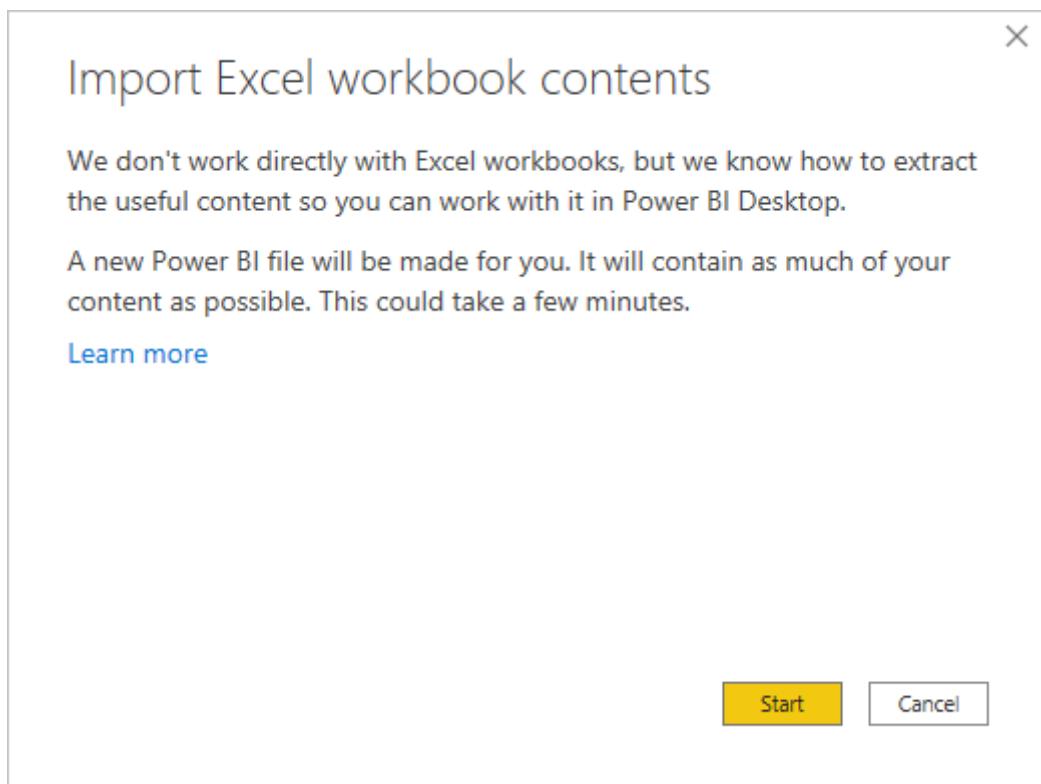
Although there's currently no limitation on the size or number of objects in the workbook, larger workbooks take longer for Power BI Desktop to analyze and

import.

 **Note**

To load or import Excel files from shared OneDrive for work or school folders or from Microsoft 365 group folders, use the URL of the Excel file, and input it into the Web data source in Power BI Desktop. There are a few steps you need to follow to properly format the OneDrive for work or school URL; for information and the correct series of steps, see [Use OneDrive for work or school links in Power BI Desktop](#).

3. From the import dialog box that appears, select **Start**.



Power BI Desktop analyzes the workbook and converts it into a Power BI Desktop file (.pbix). This action is a one-time event. Once created with these steps, the Power BI Desktop file has no dependence on the original Excel workbook. You can modify, save, and share it without affecting the original workbook.

After the import finishes, a summary page appears that describes the items that were converted. The summary page also lists any items that couldn't be imported.

Import Excel workbook contents

✓ Migration completed

Queries (5 items)

- ✓ Categories
- ✓ Customers
- ✓ Order_Details
- ✓ Orders
- ✓ Products

Data model tables (5 items)

- ✓ Categories

[Close](#)

4. Select **Close**.

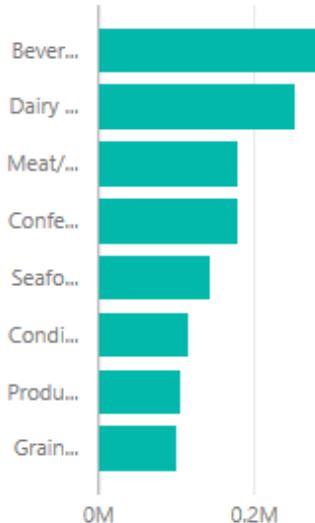
Power BI Desktop imports the Excel workbook and loads a report based on the workbook contents.

Sales Report

1.35M

Total_Sales

Total_Sales by CategoryName



Total_Sales by Country



Power View1



After the workbook is imported, you can continue working on the report. You can create new visualizations, add data, or create new report pages by using any of the features and capabilities included in Power BI Desktop.

Which workbook elements import?

Power BI Desktop can import the following elements, commonly referred to as *objects*, in Excel.

Object in Excel workbook	Final result in Power BI Desktop file
Power Query queries	All Power Query queries from Excel are converted to queries in Power BI Desktop. If there are query groups defined in the Excel Workbook, the same organization replicates in Power BI Desktop. All queries are loaded unless they're set to Only Create Connection in the Import Data Excel dialog box. Customize the load behavior by selecting Properties from the Home tab of Power Query Editor in Power BI Desktop.
Power Pivot external data connections	All Power Pivot external data connections convert to queries in Power BI Desktop.
Linked tables or current workbook tables	If there's a worksheet table in Excel linked to the data model, or linked to a query (by using <i>From Table</i> or the <i>Excel.CurrentWorkbook()</i> function in M), you'll see the following options: Import the table to the Power BI Desktop file. This table is a one-time snapshot of the data, after which the data is read-only in the table in Power BI Desktop. There's a size limitation of 1 million characters (total, combining all column headers and cells) for tables created using this option. Keep a connection to the original workbook. Alternatively, you can keep a connection to the original Excel Workbook. Power BI Desktop retrieves the latest content in this table with each refresh, just like any other query you create against an Excel workbook in Power BI Desktop.
Data model calculated columns, measures, KPIs, data categories, and relationships	These data model objects convert to the equivalent objects in Power BI Desktop. Note there are certain data categories that aren't available in Power BI Desktop, such as Image . In these cases, the data category information resets for the columns in question.

Are there any limitations to importing a workbook?

There are a few limitations to importing a workbook into Power BI Desktop:

- **External connections to SQL Server Analysis Services tabular models:** In Excel 2013, it's possible to create a connection to SQL Server Analysis Services tabular models without the need to import the data. This type of connection isn't currently supported as part of importing Excel workbooks into Power BI Desktop. As a workaround, you must recreate these external connections in Power BI Desktop.
- **Hierarchies:** This type of data model object isn't currently supported in Power BI Desktop. As such, hierarchies are skipped as part of importing an Excel workbook into Power BI Desktop.
- **Binary data columns:** This type of data model column isn't currently supported in Power BI Desktop. Binary data columns are removed from the resulting table in Power BI Desktop.
- **Named Ranges using *From Table* in Power Query, or using *Excel.CurrentWorkbook* in M:** Importing this named range data into Power BI Desktop isn't currently supported, but it's a planned update. Currently, these named ranges are loaded into Power BI Desktop as a connection to the external Excel workbook.
- **PowerPivot to SSRS:** PowerPivot external connections to SQL Server Reporting Services (SSRS) aren't currently supported, because that data source isn't currently available in Power BI Desktop.

Connect to an Oracle database with Power BI Desktop

Article • 07/26/2023

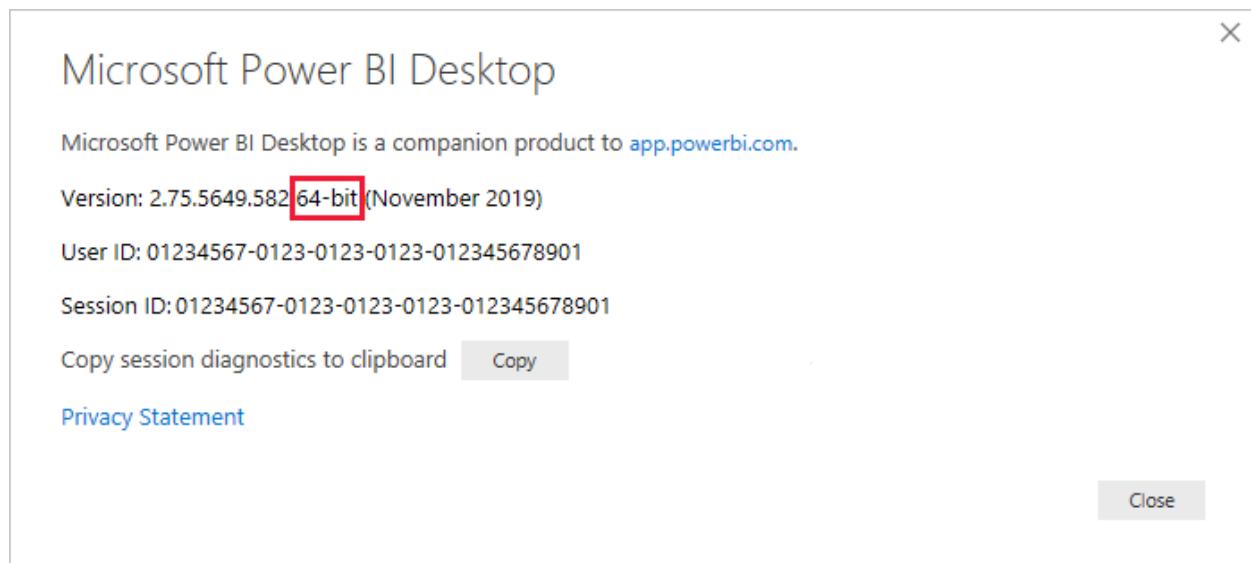
To connect to an Oracle database or Oracle Autonomous Database with Power BI Desktop, install Oracle Client for Microsoft Tools (OCMT) on the computer running Power BI Desktop. The OCMT software you use depends on which version of Power BI Desktop you've installed: 32-bit or 64-bit. It also depends on your version of Oracle server.

Supported Oracle Database versions:

- Oracle Database 12c (12.1.0.2) and later
- Oracle Autonomous Database - all versions

Determining which version of Power BI Desktop is installed

To determine which version of Power BI Desktop is installed, on the **Help** ribbon, select **About**, then check the **Version** line. In the following image, a 64-bit version of Power BI Desktop is installed:



Install the Oracle Client for Microsoft Tools

Oracle Client for Microsoft Tools installs and configures Oracle Data Provider for .NET (ODP.NET) to support 32-bit and 64-bit Microsoft tool connections with Oracle on-premises and cloud databases, including Oracle Autonomous Database. It is a graphical

installer that automates the Oracle Database Client setup process. It supports connecting with Power BI Desktop, Power BI service, Excel, SQL Server Analysis Services, SQL Server Data Tools, SQL Server Integration Services, SQL Server Reporting Services, and BizTalk Server.

OCMT is free software. It can be downloaded from the [Oracle Client for Microsoft Tools page](#) and is available for 32-bit or 64-bit Power BI Desktop.

Power BI Desktop uses unmanaged ODP.NET to connect to Oracle database or Oracle Autonomous Database.

Here are [step by step instructions to use OCMT and setup Oracle database connectivity to Power BI Desktop](#).

Connect to an Oracle database with on-premises data gateway

Some Power BI Desktop app deployments use on-premises data gateway to connect to Oracle database. To connect to an Oracle database with the [on-premises data gateway](#), use 64-bit OCMT on the computer running the gateway since the gateway is a 64-bit app. For more information, go to [Manage your data source - Oracle](#).

Connect to an Oracle Database

For information about connecting to an Oracle database or an Oracle Autonomous database from either Power BI Desktop or Power BI service, go to the Power Query article on [Oracle databases](#).

Next steps

- [DirectQuery in Power BI](#)
- [What is Power BI?](#)
- [Data sources for the Power BI service](#)
- [Oracle Client for Microsoft Tools](#)

More questions? [Ask the Power BI Community](#)

Run Python scripts in Power BI Desktop

Article • 12/14/2022

You can run Python scripts directly in Power BI Desktop and import the resulting datasets into a Power BI Desktop data model. From this model, you can create reports and share them on the Power BI service.

Prerequisites

- To run Python scripts in Power BI Desktop, you need to install Python on your local machine. You can download Python from the [Python website](#). The current Python scripting release supports Unicode characters and spaces in the installation path.
- The Power BI Python integration requires installation of the following two Python packages. In a console or shell, use the [pip](#) command-line tool to install the packages. The pip tool is packaged with recent Python versions.
 - [Pandas](#) is a software library for data manipulation and analysis. Pandas offers data structures and operations for manipulating numerical tables and time series. To import into Power BI, Python data must be in a [pandas data frame](#). A data frame is a two-dimensional data structure, such as a table with rows and columns.
 - [Matplotlib](#) is a plotting library for Python and its numerical mathematics extension [NumPy](#). Matplotlib provides an object-oriented API for embedding plots into general-purpose graphical user interface (GUI) applications for Python, such as Tkinter, wxPython, Qt, or GTK+.

Console

```
pip install pandas  
pip install matplotlib
```

Enable Python scripting

To enable Python scripting in Power BI:

1. In Power BI Desktop, select **File > Options and settings > Options > Python scripting**. The **Python script options** page appears.

Options

GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Python scripting**
- Security
- Privacy
- Updates
- Usage Data
- Diagnostics
- Preview features
- Auto recovery
- Report settings

CURRENT FILE

- Data Load
- Regional Settings
- Privacy
- Auto recovery
- DirectQuery
- Query reduction
- Report settings

Python script options

To choose a home directory for Python, select a detected Python installation from the drop-down list, or select Other and browse to the location you want.

Detected Python home directories:

C:\Users\Python

[How to install Python](#)

To choose which Python integrated development environment (IDE) you want Power BI Desktop to launch, select a detected IDE from the drop-down list, or select Other to browse to another IDE on your machine.

Detected Python IDEs:

Visual Studio Code

[Learn more about Python IDEs](#)

[Change temporary storage location](#)

Note: Sometimes, Python custom visuals automatically install additional packages. For those to work, the temporary storage folder name must be written in Latin characters (letters in the English alphabet).

OK

Cancel

2. If necessary, supply or edit your local Python installation path under **Detected Python home directories**. In the preceding image, the Python's installation local path is *C:\Python*. If you have more than one local Python installation, make sure to select the one that you want to use.

3. Select OK.

Important

Power BI runs scripts directly by using the *python.exe* executable from the directory you provide in **Settings**. Python distributions that require an extra step to prepare the environment, such as Conda, might fail to run. To avoid these issues, use the official Python distribution from <https://www.python.org>. Another possible solution is to start Power BI Desktop from your custom Python environment prompt.

Create a Python script

Create a script in your local Python development environment and make sure it runs successfully. To prepare and run a Python script in Power BI Desktop, there are a few limitations:

- Only pandas data frames import, so make sure the data you want to import to Power BI is represented in a data frame.
- Any Python script that runs longer than 30 minutes times out.
- Interactive calls in the Python script, such as waiting for user input, halt the script's execution.
- If you set a working directory within the Python script, you must define a full path to the working directory rather than a relative path.
- Nested tables aren't supported.

Here's a simple example Python script that imports pandas and uses a data frame:

Python

```
import pandas as pd
data = [['Alex',10],['Bob',12],['Clarke',13]]
df = pd.DataFrame(data,columns=[ 'Name' , 'Age' ],dtype=float)
print (df)
```

When run, this script returns:

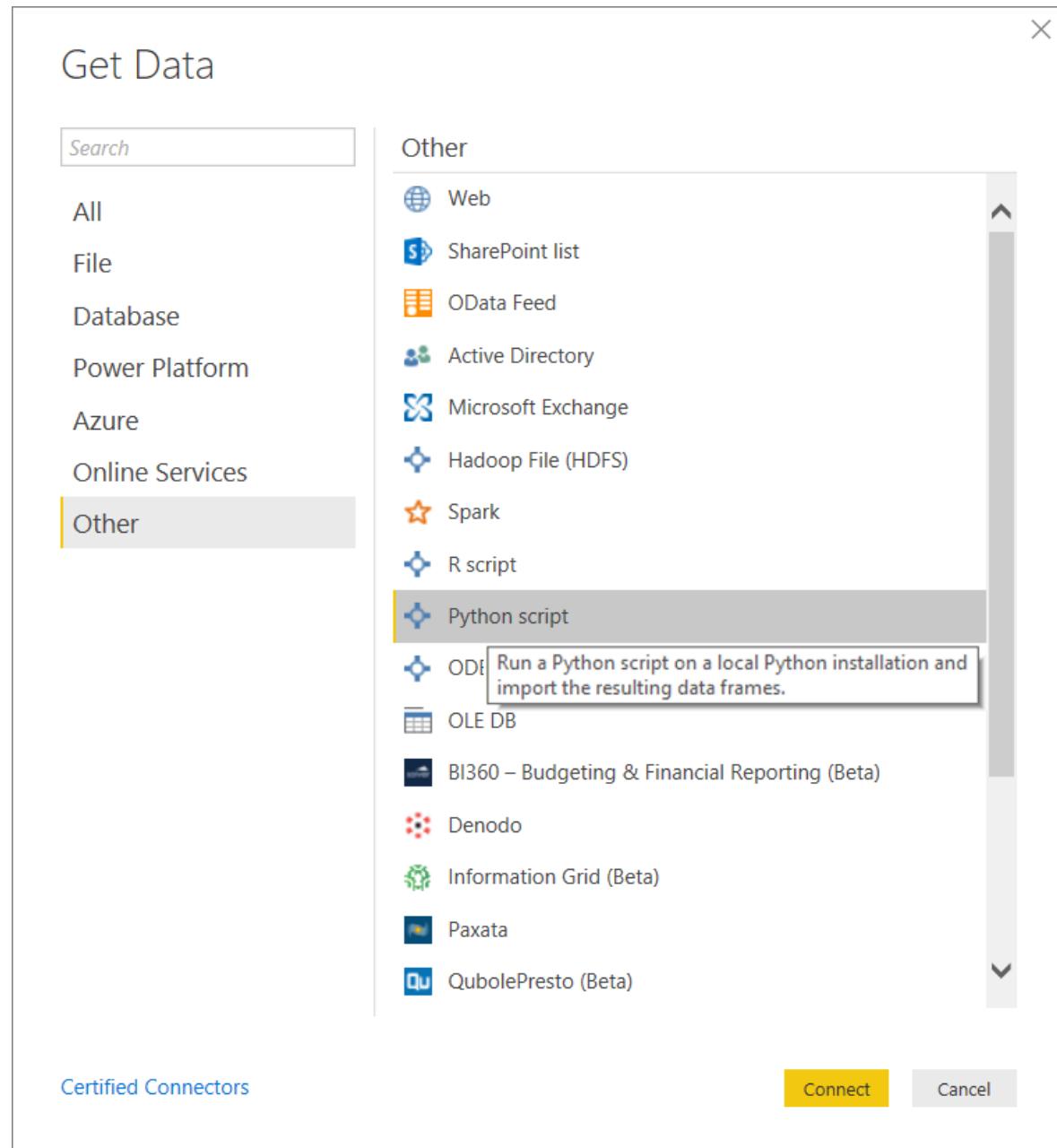
Output

	Name	Age
0	Alex	10.0
1	Bob	12.0
2	Clarke	13.0

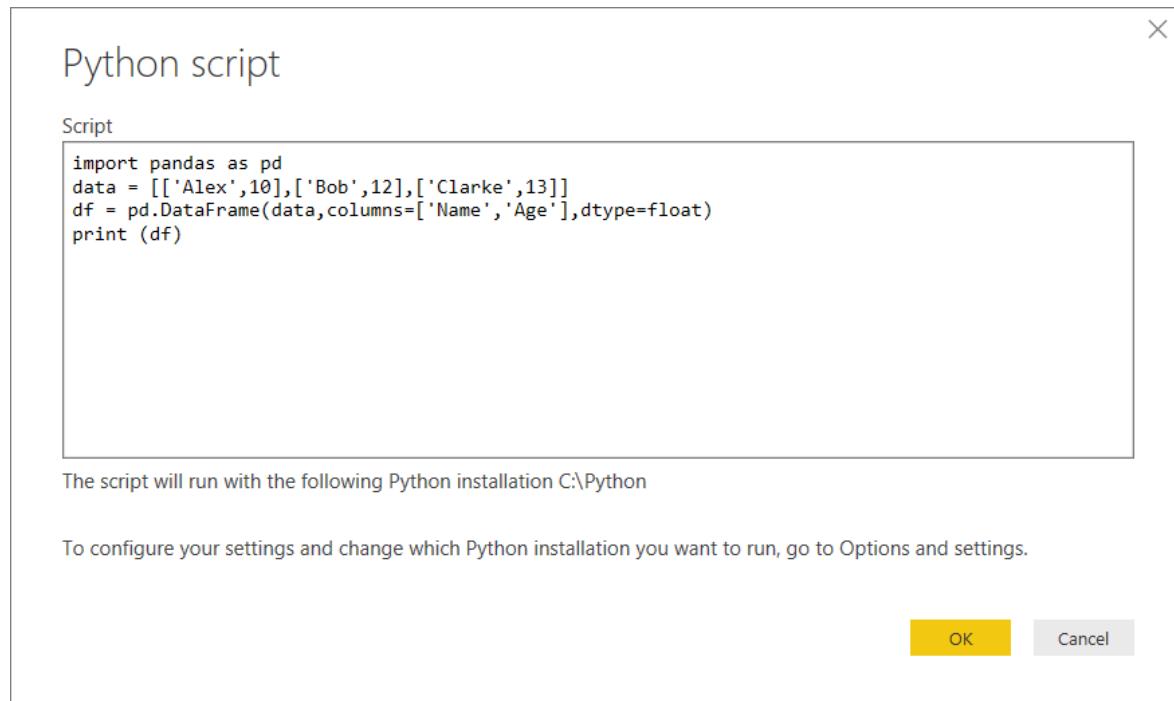
Run the script and import data

To run your Python script:

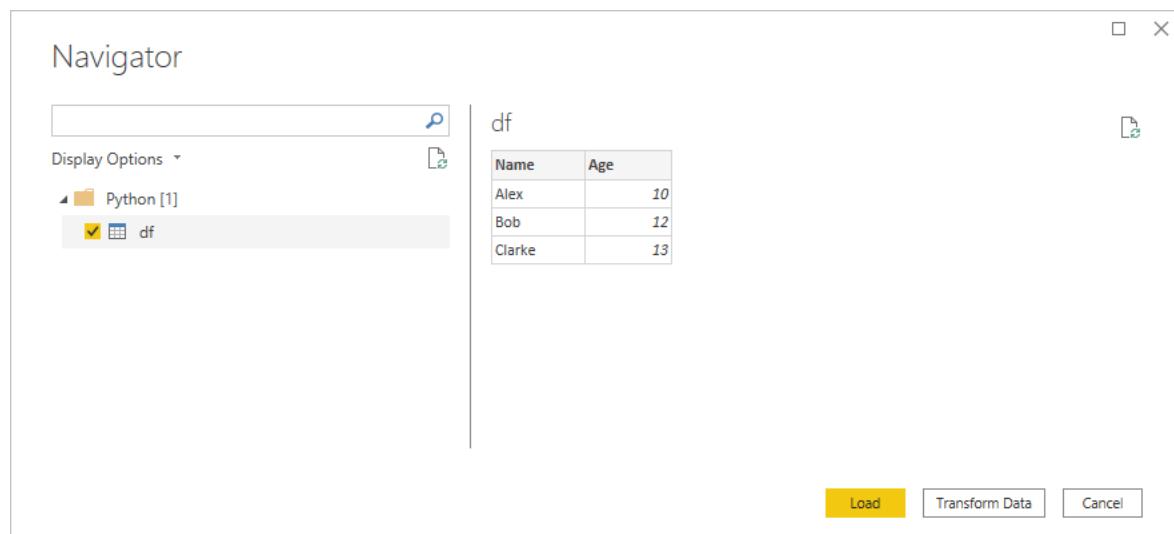
1. In the **Home** group of the Power BI Desktop ribbon, select **Get data**.
2. In the **Get Data** dialog box, select **Other > Python script**, and then select **Connect**.
Power BI uses your latest installed Python version as the Python engine.



3. On the **Python script** screen, paste your Python script into the **Script** field, and select **OK**.



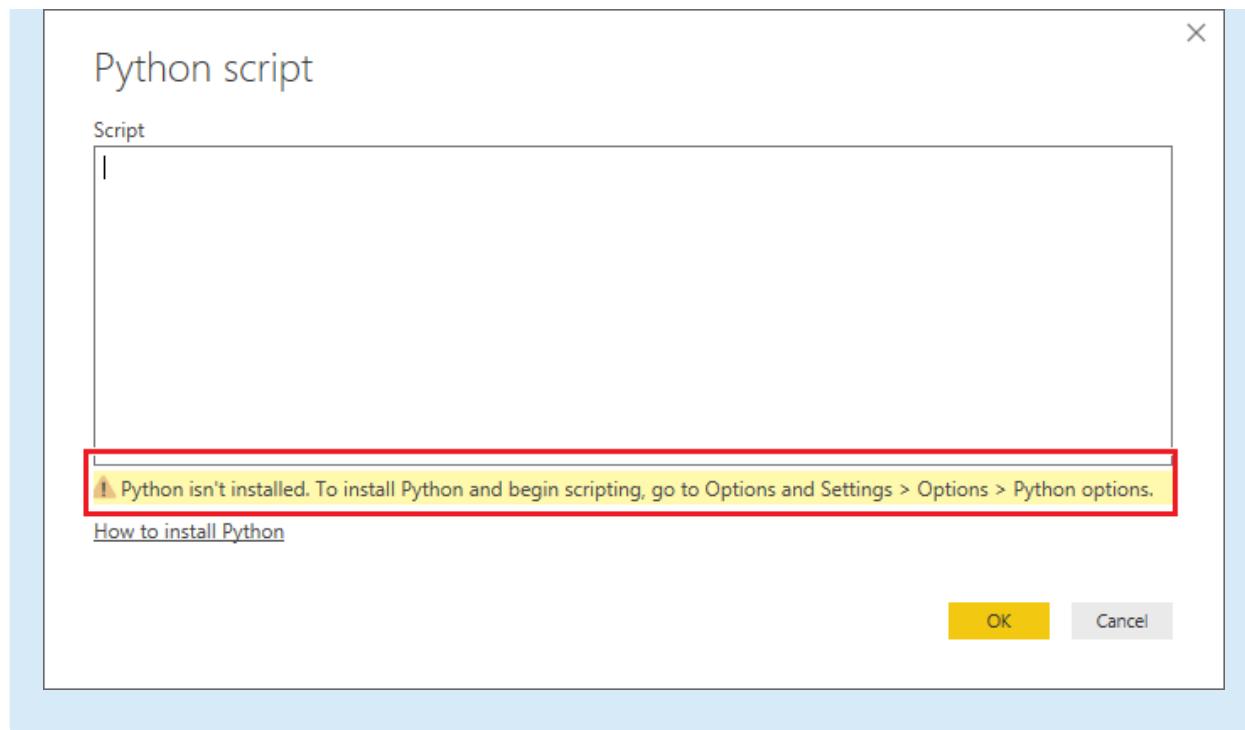
4. If the script runs successfully, the **Navigator** window appears, and you can load the data. Select the **df** table, and then select **Load**.



Power BI imports the data, and you can use it to create visualizations and reports. To refresh the data, select **Refresh** in the **Home** group of the Power BI Desktop ribbon. When you refresh, Power BI runs the Python script again.

ⓘ Important

If Python isn't installed or identified, a warning appears. You might also get a warning if you have multiple local machine installations.



Next steps

For more information about Python in Power BI, see:

- [Create Python visuals in Power BI Desktop](#)
- [Use an external Python IDE with Power BI](#)
- [Use Python in Power Query Editor](#)

Use Python in Power Query Editor

Article • 02/13/2023

You can use *Python*, a programming language widely used by statisticians, data scientists, and data analysts, in the Power BI Desktop **Power Query Editor**. This integration of Python into **Power Query Editor** lets you perform data cleansing using Python, and perform advanced data shaping and analytics in datasets, including completion of missing data, predictions, and clustering, just to name a few. Python is a powerful language, and can be used in **Power Query Editor** to prepare your data model and create reports.

Prerequisites

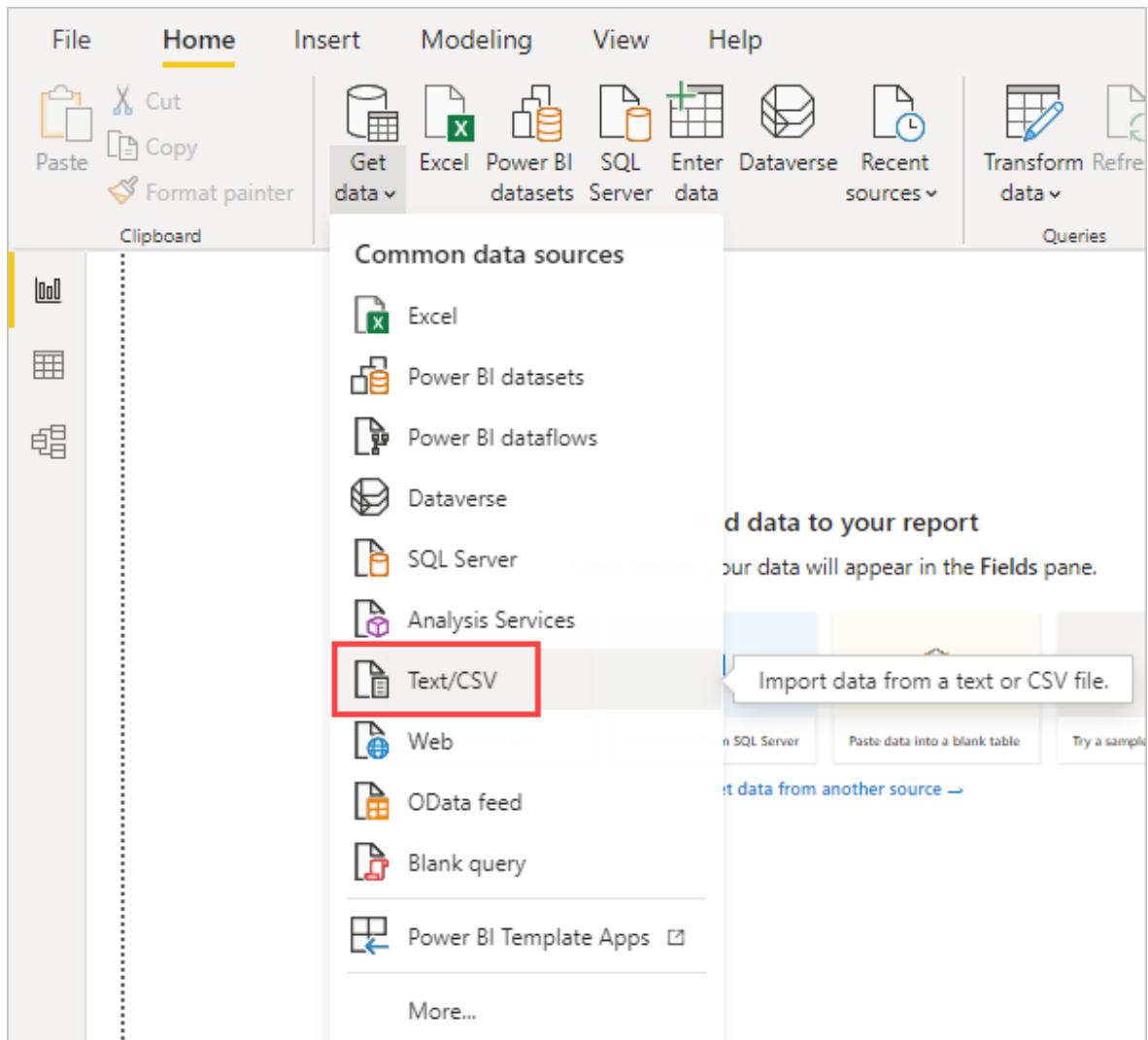
You'll need to install Python and pandas before you begin.

- **Install Python** - To use Python in Power BI Desktop's *Power Query Editor*, you need to install Python on your local machine. You can download and install Python for free from many locations, including the [Official Python download page](#) and the [Anaconda](#).
- **Install pandas** - To use Python with the Power Query Editor, you'll also need to install [pandas](#). Pandas is used to move data between Power BI and the Python environment.

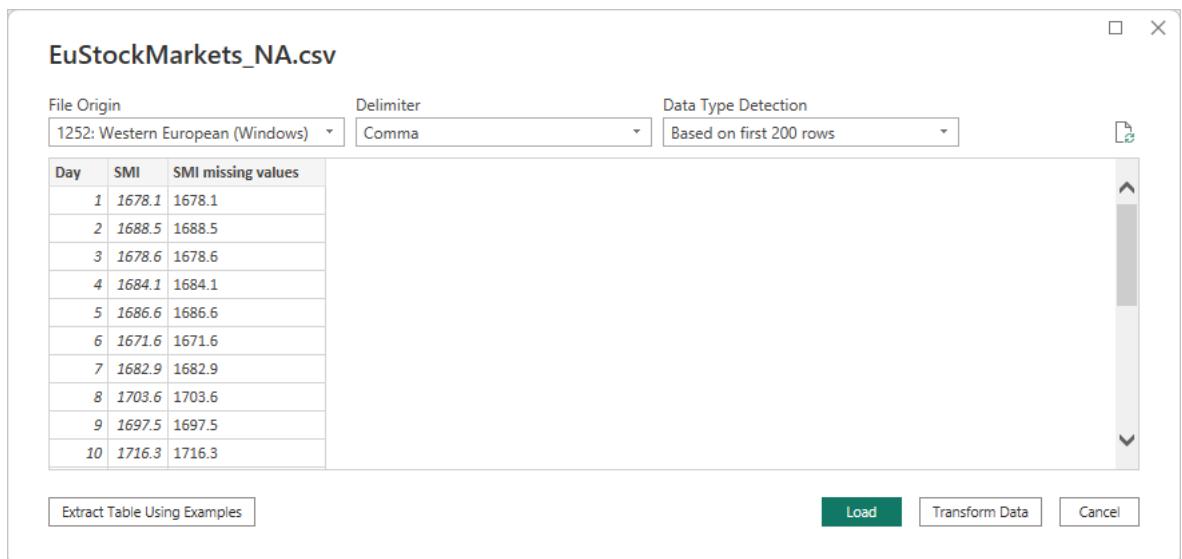
Use Python with Power Query Editor

To show how to use Python in **Power Query Editor**, take this example from a stock market dataset, based on a CSV file that you can [download from here](#) and follow along. The steps for this example are the following procedure:

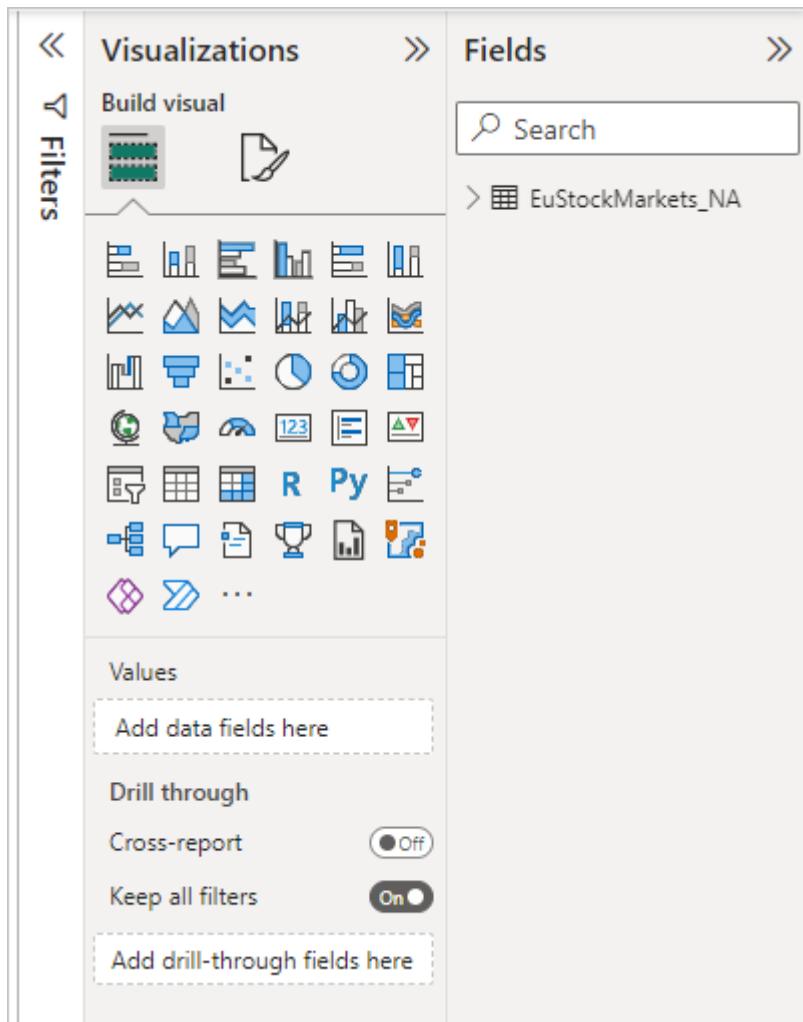
1. First, load your data into **Power BI Desktop**. In this example, load the *EuStockMarkets_NA.csv* file and select **Get data > Text/CSV** from the **Home** ribbon in **Power BI Desktop**.



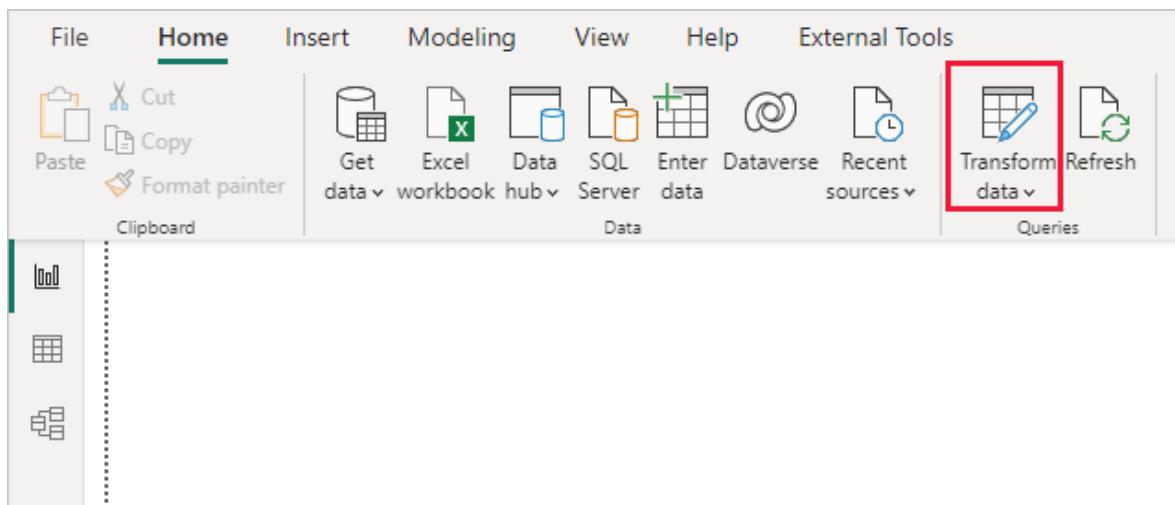
2. Select the file and select Open, and the CSV is displayed in the CSV file dialog.



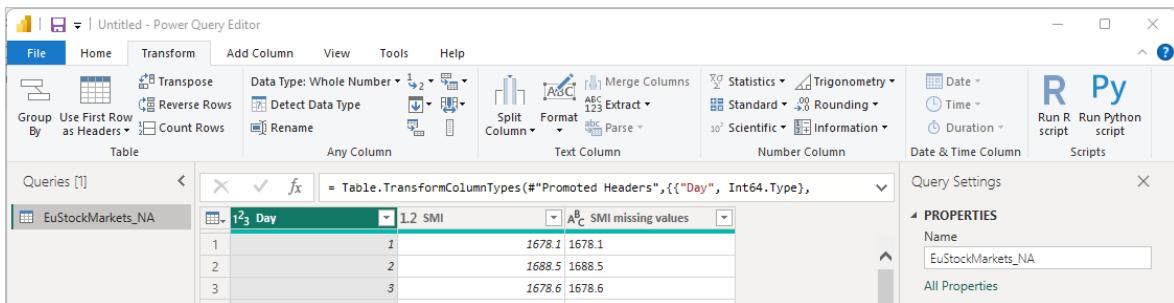
3. Once the data is loaded, you see it in the Fields pane in Power BI Desktop.



4. Open Power Query Editor by selecting **Transform data** from the **Home** tab in **Power BI Desktop**.



5. In the **Transform** tab, select **Run Python Script** and the **Run Python Script** editor appears as shown in the next step. Rows 15 and 20 suffer from missing data, as do other rows you can't see in the following image. The following steps show how Python completes those rows for you.



6. For this example, enter the following script code:

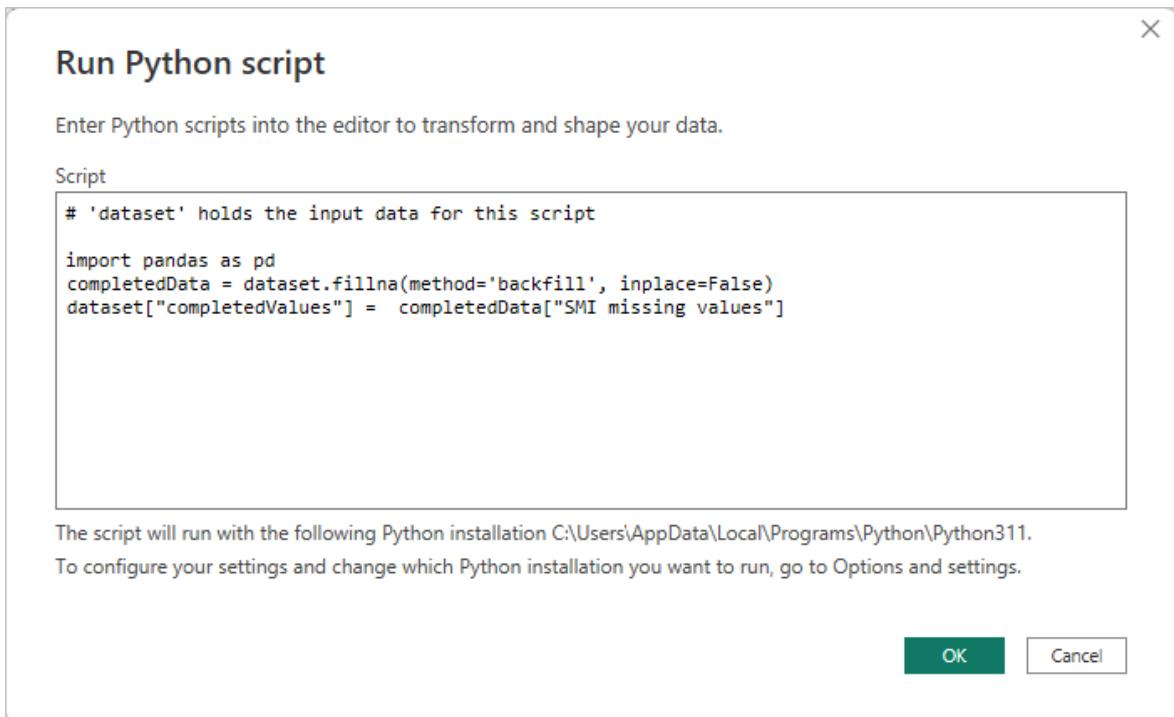
Python

```
import pandas as pd
completedData = dataset.fillna(method='backfill', inplace=False)
dataset["completedValues"] = completedData["SMI missing values"]
```

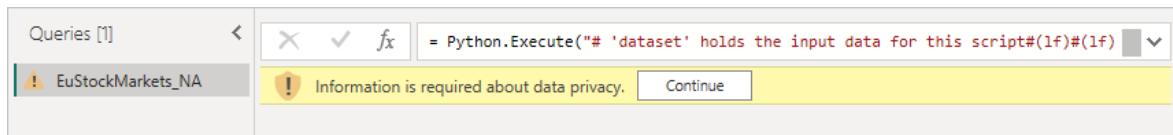
! Note

You need to have the *pandas* library installed in your Python environment for the previous script code to work properly. To install pandas, run the following command in your Python installation: `pip install pandas`

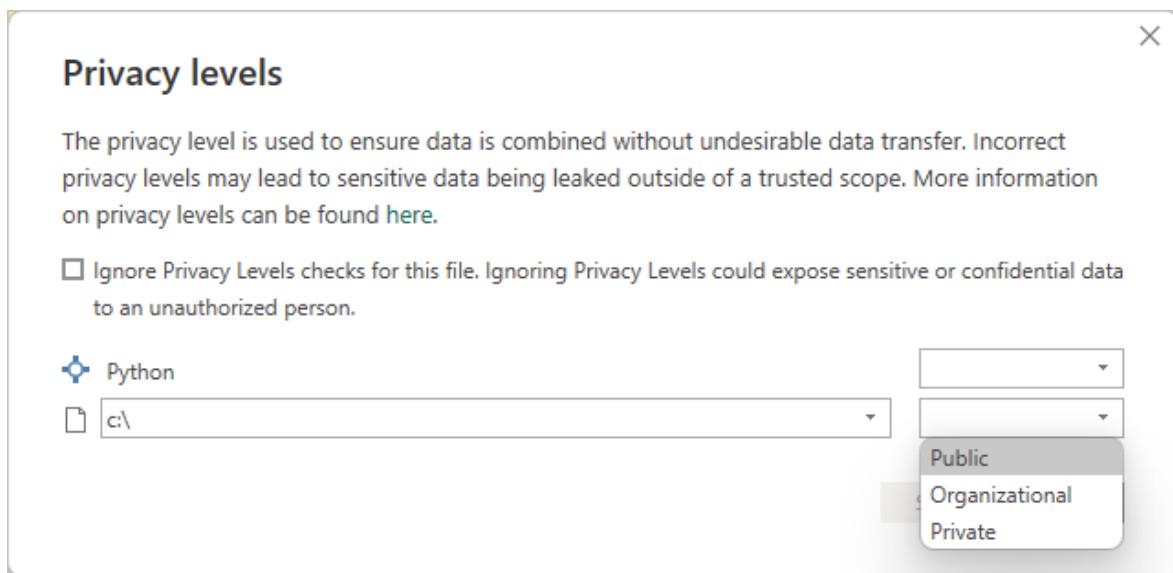
When put into the Run Python Script dialog, the code looks like the following example:



7. After you select OK, Power Query Editor displays a warning about data privacy.



8. For the Python scripts to work properly in the Power BI service, all data sources need to be set to *public*. For more information about privacy settings and their implications, see [Privacy Levels](#).

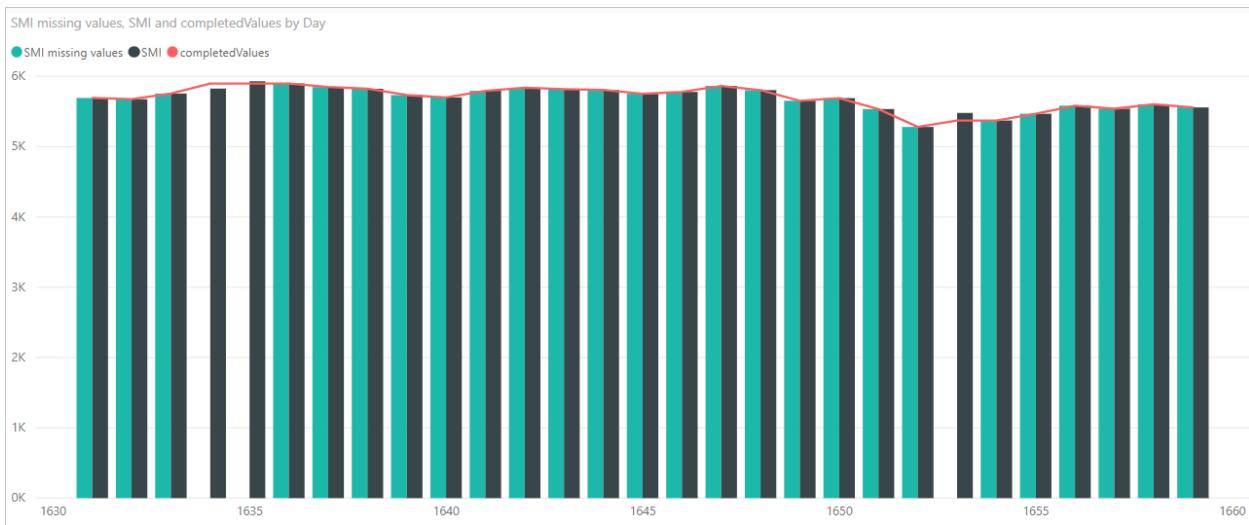


Notice a new column in the **Fields** pane called *completedValues*. Notice there are a few missing data elements, such as on row 15 and 18. Take a look at how Python handles that in the next section.

With just three lines of Python script, **Power Query Editor** filled in the missing values with a predictive model.

Create visuals from Python script data

Now we can create a visual to see how the Python script code using the *pandas* library completed the missing values, as shown in the following image:



Once that visual is complete, and any other visuals you might want to create using **Power BI Desktop**, you can save the **Power BI Desktop** file. **Power BI Desktop** files save with the **.pbix** file name extension. Then use the data model, including the Python scripts that are part of it, in the Power BI service.

! Note

Want to see a completed .pbix file with these steps completed? You're in luck. You can download the completed **Power BI Desktop** file used in these examples [right here ↗](#).

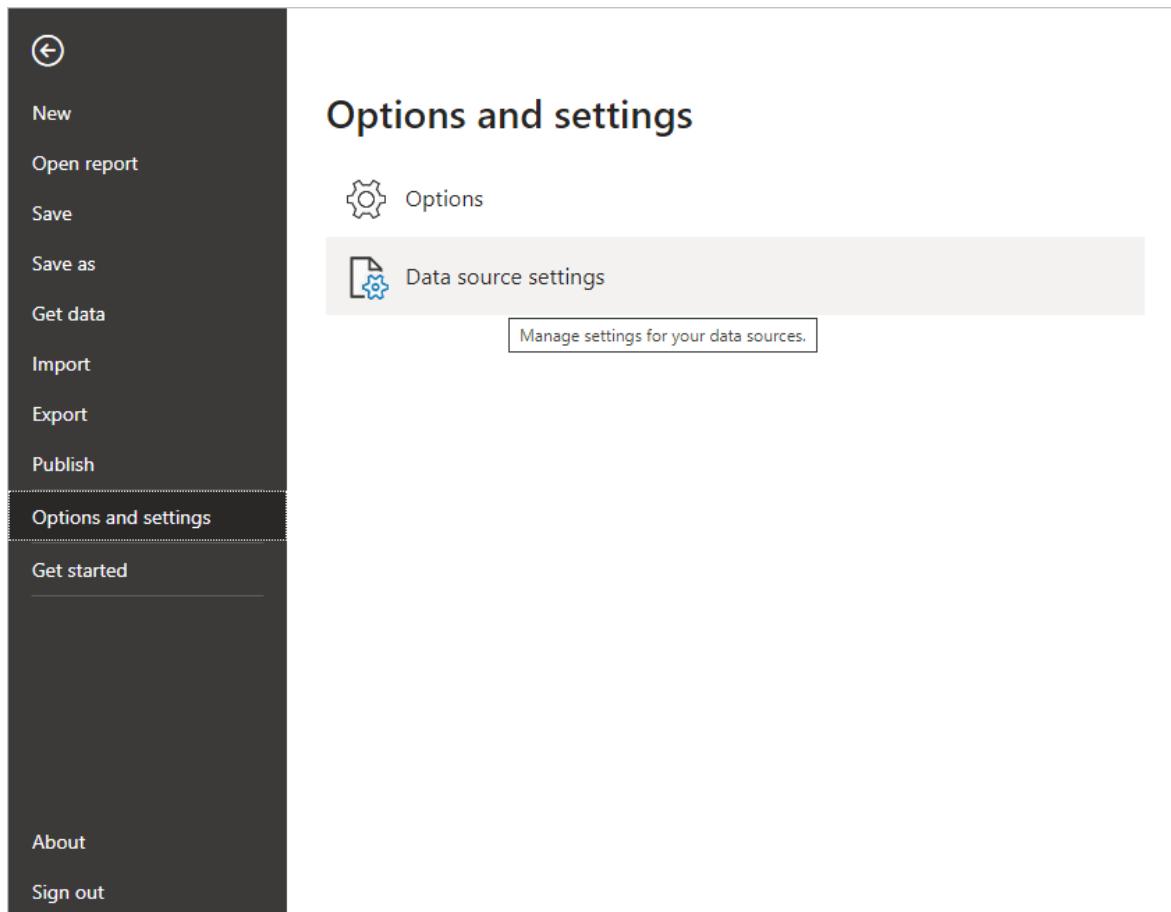
Once you upload the .pbix file to the Power BI service, a couple more steps are necessary to enable data to refresh in the service and to enable visuals to be updated in the service. The data needs access to Python for visuals to be updated. The other steps are the following steps:

- **Enable scheduled refresh for the dataset.** To enable scheduled refresh for the workbook that contains your dataset with Python scripts, see [Configuring scheduled refresh](#), which also includes information about **Personal Gateway**.
- **Install the Personal Gateway.** You need a *Personal Gateway* installed on the machine where the file is located, and where Python is installed. The Power BI service must access that workbook and re-render any updated visuals. For more information, see [install and configure Personal Gateway](#).

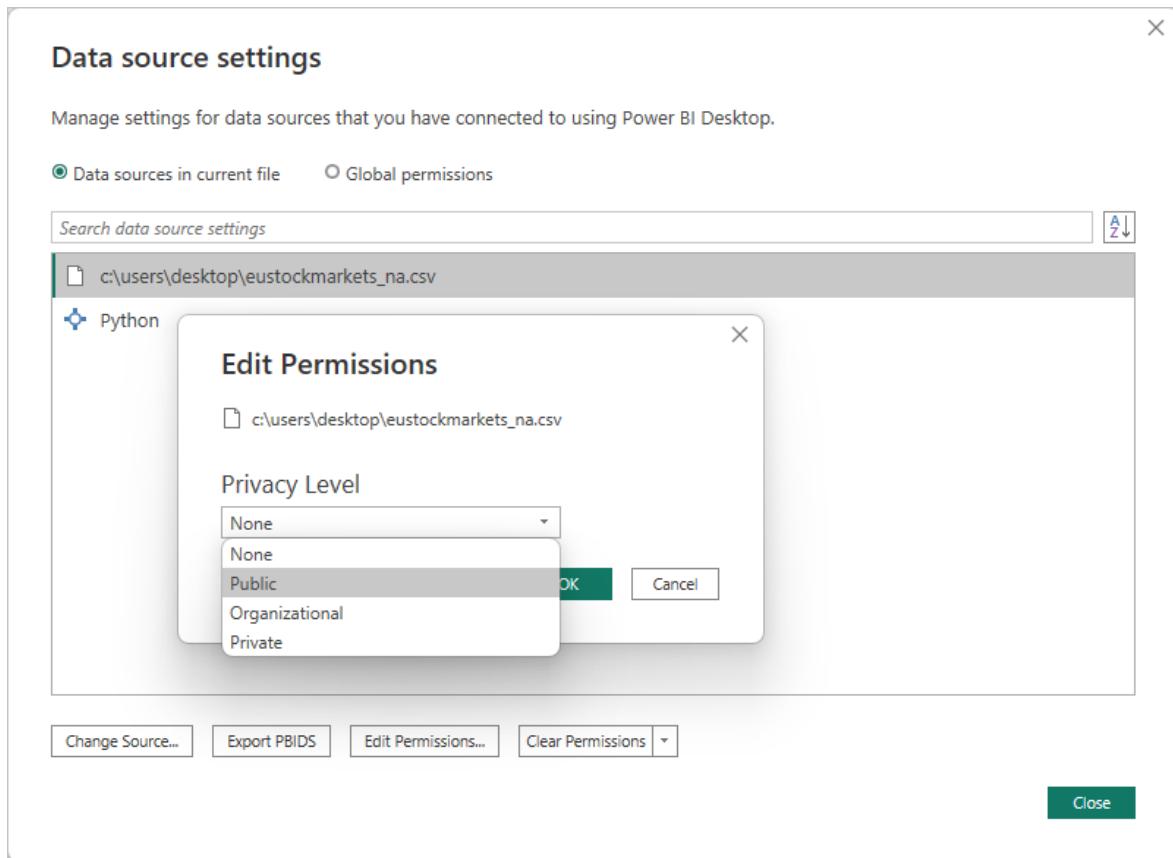
Considerations and limitations

There are some limitations to queries that include Python scripts created in **Power Query Editor**:

- All Python data source settings must be set to *Public*, and all other steps in a query created in **Power Query Editor** must also be public. To get to data source settings, in **Power BI Desktop** select **File > Options and settings > Data source settings**.



From the **Data Source Settings** dialog, select the data sources and then select **Edit Permissions...** and ensure that the **Privacy Level** is set to *Public*.



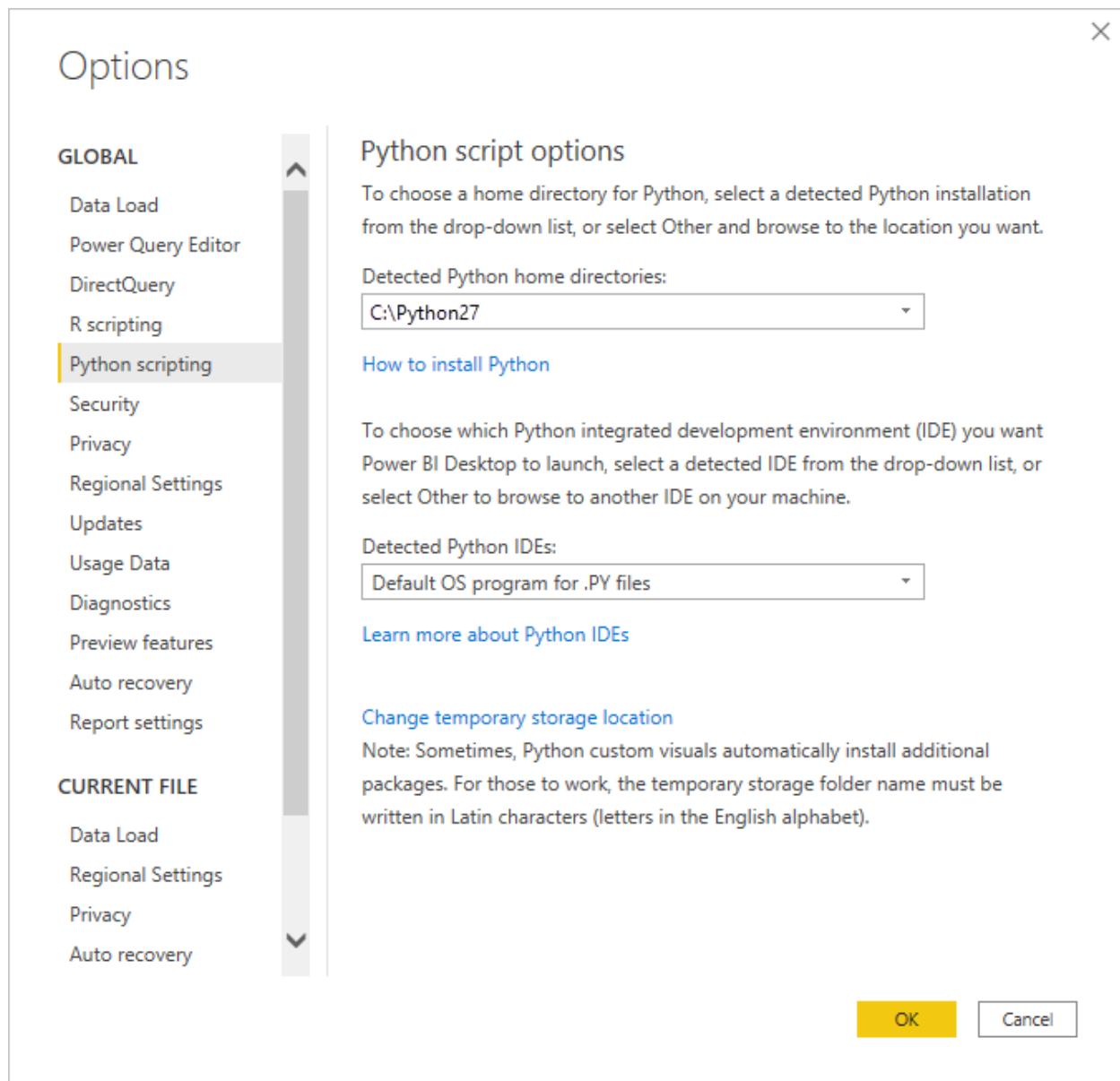
- To enable scheduled refresh of your Python visuals or dataset, you need to enable **Scheduled refresh** and have a Personal Gateway installed on the computer that houses the workbook and the Python installation. For more information on both, see the previous section in this article, which provides links to learn more about each.
- Nested tables, which are table of tables, are currently not supported.

There are all sorts of things you can do with Python and custom queries, so explore and shape your data just the way you want it to appear.

Use an external Python IDE with Power BI

Article • 01/13/2023

With Power BI Desktop, you can use your external Python Integrated Development Environment (IDE) to create and refine Python scripts, then use those scripts in Power BI.



Enable an external Python IDE

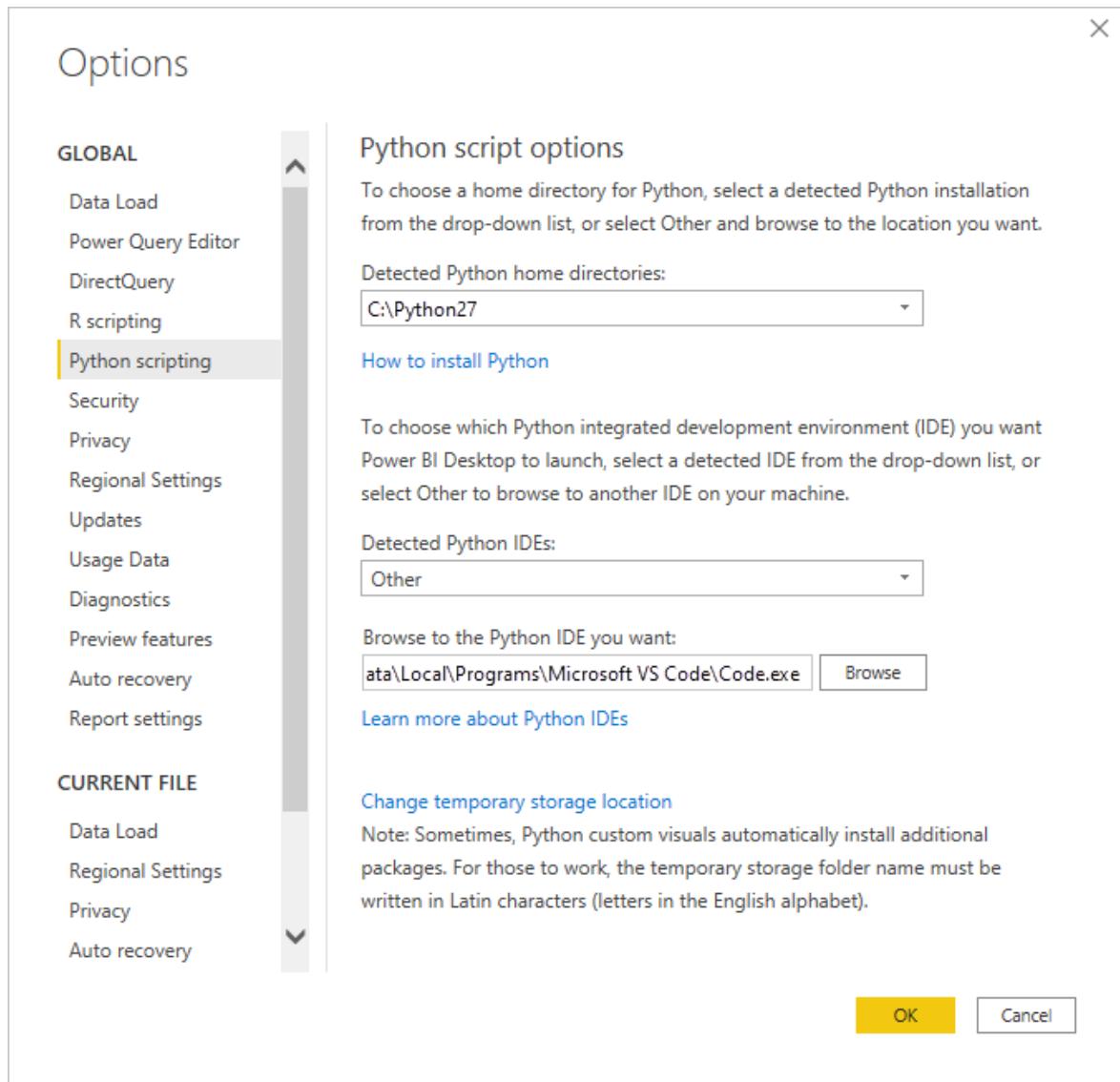
You can launch your external Python IDE from Power BI Desktop and have your data automatically imported and displayed in the Python IDE. From there, you can modify the script in that external Python IDE, then paste it back into Power BI Desktop to create Power BI visuals and reports.

You can specify which Python IDE to use, and have it launch automatically from within Power BI Desktop.

Requirements

To use this feature, you need to install a Python IDE on your local computer. Power BI Desktop doesn't include, deploy, or install the Python engine, so you must separately install Python on your local computer. You can choose which Python IDE to use, with the following options:

- You can install your favorite Python IDE, many of which are available for free, such as the [Visual Studio Code download page ↗](#).
- Power BI Desktop also supports Visual Studio.
- You can also install a different Python IDE and have Power BI Desktop launch that **Python IDE** by doing one of the following:
 - You can associate .PY files with the external IDE you want Power BI Desktop to launch.
 - You can specify the .exe that Power BI Desktop launches by selecting *Other* from the **Python script options** section of the **Options** dialog. You can bring up the **Options** dialog by going to **File > Options and settings > Options**.



If you have multiple Python IDEs installed, you can specify which is launched by selecting it from the **Detected Python IDEs** drop-down in the **Options** dialog.

By default, Power BI Desktop launches Visual Studio Code as the external Python IDE if it's installed on your local computer. If Visual Studio Code isn't installed and you have Visual Studio that is launched instead. If neither of those Python IDEs is installed, the application associated with **.PY** files is launched.

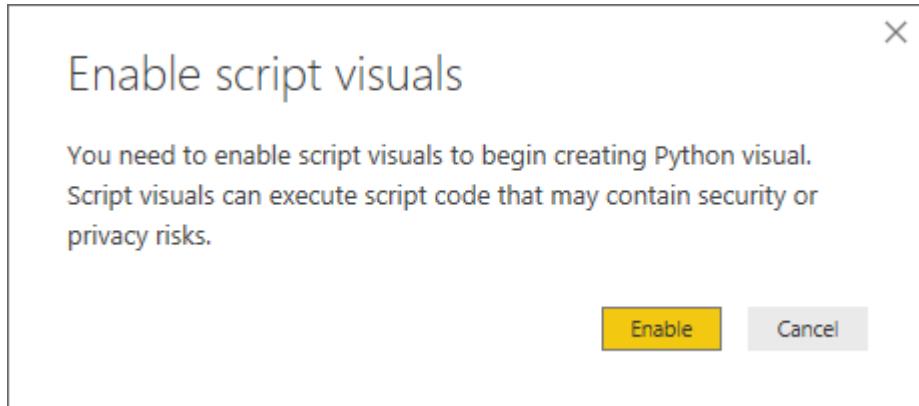
And if no **.PY** file association exists, it's possible to specify a path to a custom IDE in the **Set a Python home directory** section of the **Options** dialog. You can also launch a different Python IDE by selecting the **Settings** gear icon beside the **Launch Python IDE** arrow icon, in Power BI Desktop.

Launch a Python IDE from Power BI Desktop

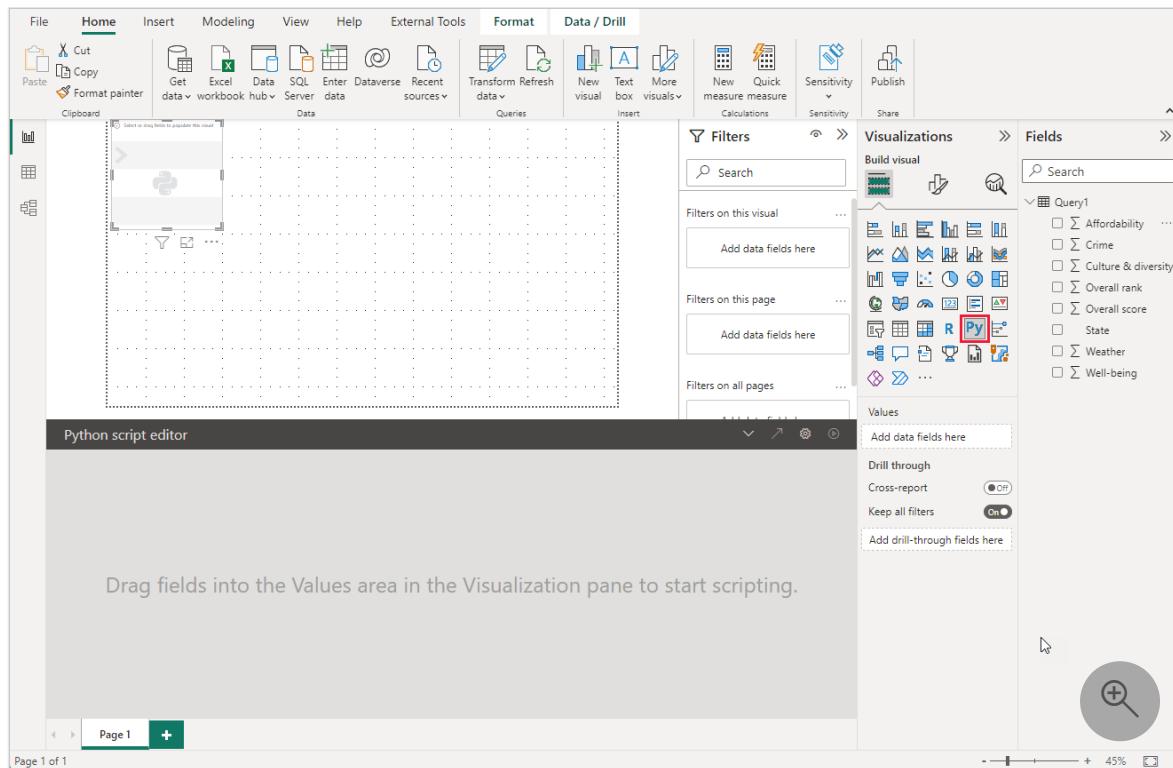
To launch a Python IDE from Power BI Desktop, take the following steps:

1. Load data into Power BI Desktop.

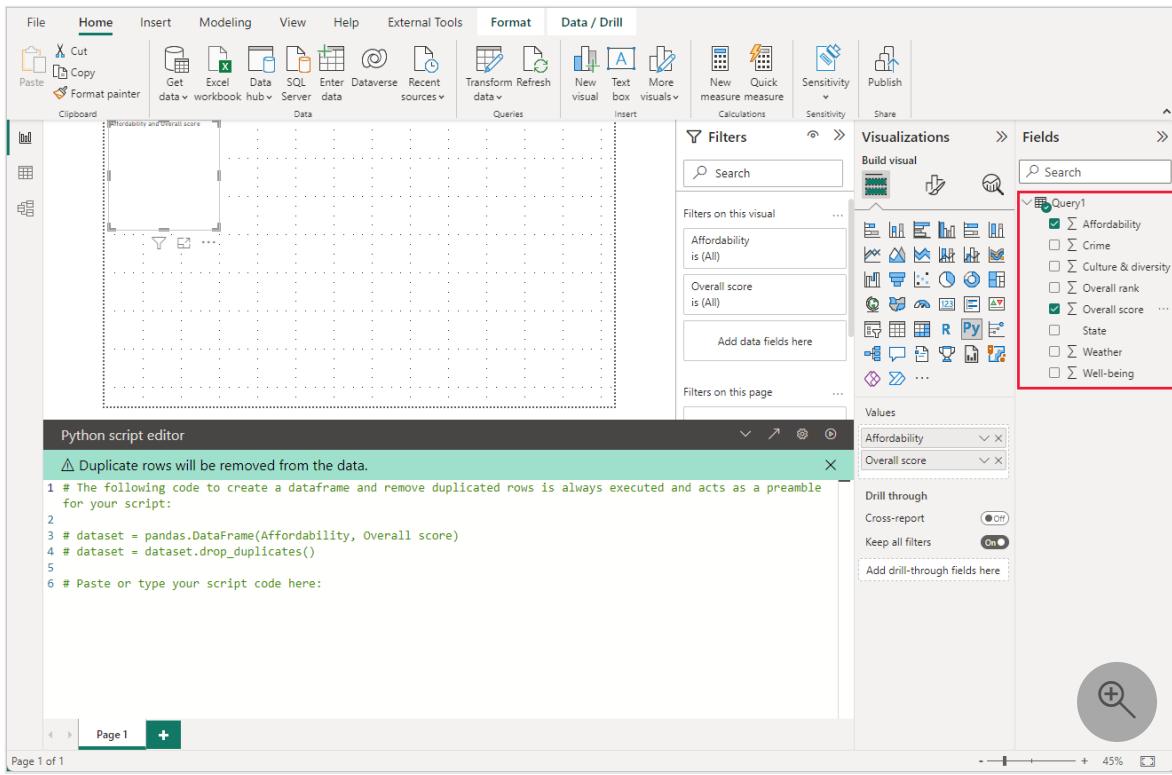
2. Add a Python visualization to your canvas. If you haven't enabled script visuals yet, you're prompted to do so.



3. After script visuals are enabled, a blank Python visual appears that's ready to display the results of your script. The **Python script editor** pane also appears.



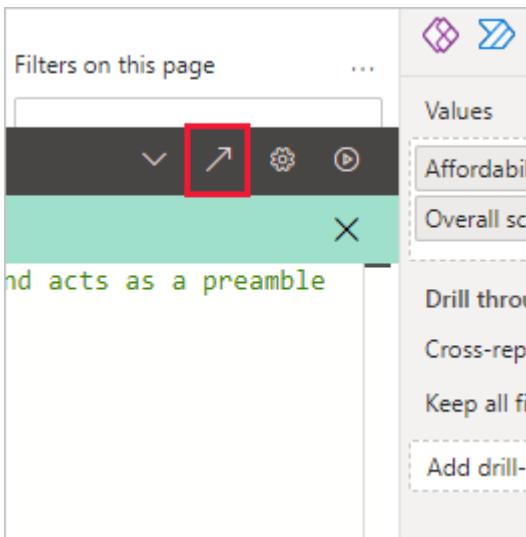
4. Now you can select the fields you want to use in your Python script. When you select a field, the **Python script editor** field automatically creates script code based on the field or fields you select. You can either create or paste your Python script directly in the **Python script editor** pane, or you can leave it empty.



ⓘ Note

The default aggregation type for Python visuals is *do not summarize*.

5. You can now launch your Python IDE directly from Power BI Desktop. Select the **Launch Python IDE** button, found on the right side of the **Python script editor** title bar, as shown in this screenshot.



6. Your specified Python IDE is launched by Power BI Desktop, as shown in the following image. In this image, Visual Studio Code is the default Python IDE.

```
# Prolog - Auto Generated #
import os, pandas
os.chdir(u'C:/Users/xxx/PythonEditorWrapper_74074cf2-f741-4863-b3a5-0dca11842e70')
dataset = pandas.read_csv('input_df_965ba9b6-c41d-4edf-b7a7-883bc0ae7e9f.csv')

# Original Script. Please update your script content here and once completed copy below section back
#####
#####
```

⚠ Note

Power BI Desktop adds the first three lines of the script so it can import your data from Power BI Desktop once you run the script.

7. Any script you created in the **Python script editor pane** of Power BI Desktop appears, starting in line 4, in your Python IDE. At this point, you can create your Python script in the Python IDE. Once your Python script is complete in your Python IDE, you need to copy and paste it back into the **Python script editor pane** in Power BI Desktop, *excluding* the first three lines of the script that Power BI Desktop automatically generated. Don't copy the first three lines of script back into Power BI Desktop, those lines were only used to import your data to your Python IDE from Power BI Desktop.

Known limitations

Launching a Python IDE directly from Power BI Desktop has a few limitations:

- Automatically exporting your script from your Python IDE into Power BI Desktop isn't supported.

Next steps

Take a look at the following additional information about Python in Power BI.

- [Running Python Scripts in Power BI Desktop](#)
- [Create Power BI visuals using Python](#)

Create Power BI visuals with Python

Article • 01/18/2023

This tutorial helps you get started creating visuals with Python data in Power BI Desktop. You use a few of the many available options and capabilities for creating visual reports by using Python, pandas, and the Matplotlib library.

Prerequisites

Work through [Run Python scripts in Power BI Desktop](#) to:

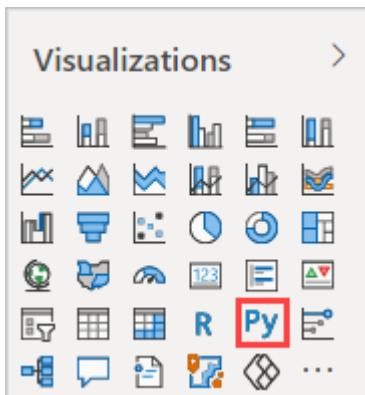
- Install [Python](#) on your local machine.
- Enable Python scripting in Power BI Desktop.
- Install the [pandas](#) and [Matplotlib](#) Python libraries.
- Import the following Python script into Power BI Desktop:

```
Python

import pandas as pd
df = pd.DataFrame({
    'Fname':['Harry','Sally','Paul','Abe','June','Mike','Tom'],
    'Age':[21,34,42,18,24,80,22],
    'Weight': [180, 130, 200, 140, 176, 142, 210],
    'Gender':['M','F','M','M','F','M','M'],
    'State':
    ['Washington','Oregon','California','Washington','Nevada','Texas','Neva
da'],
    'Children':[4,1,2,3,0,2,0],
    'Pets':[3,2,2,5,0,1,5]
})
print (df)
```

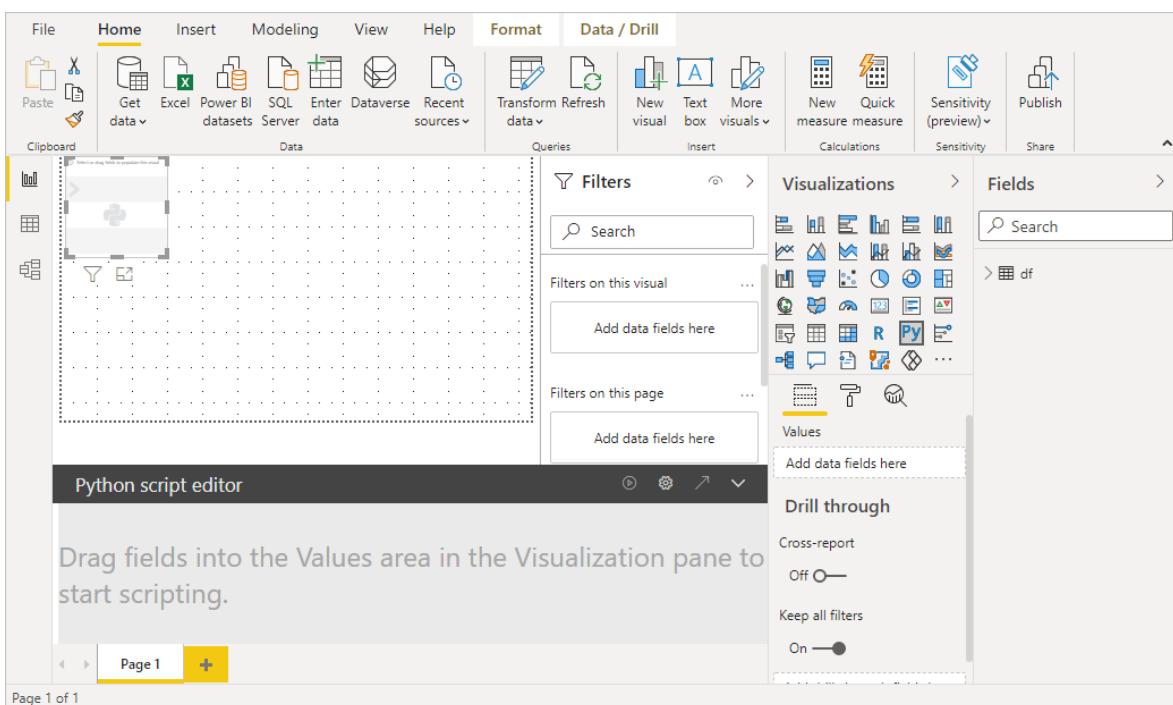
Create a Python visual in Power BI Desktop

1. After you import the Python script, select the **Python visual** icon in the Power BI Desktop **Visualizations** pane.

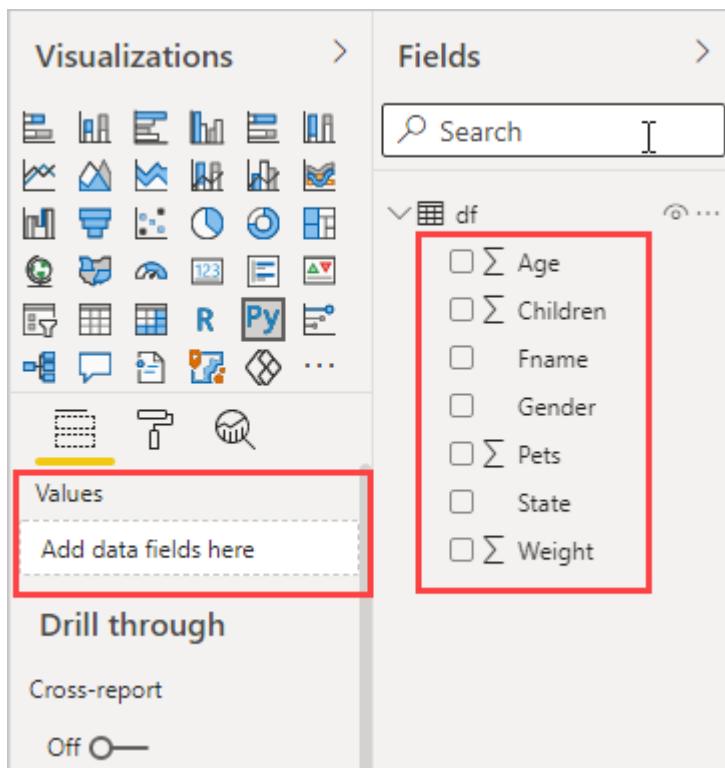


2. In the **Enable script visuals** dialog box that appears, select **Enable**.

A placeholder Python visual image appears on the report canvas, and the **Python script editor** appears along the bottom of the center pane.



3. Drag the **Age**, **Children**, **Fname**, **Gender**, **Pets**, **State**, and **Weight** fields to the **Values** section where it says **Add data fields here**.



Based on your selections, the Python script editor generates the following binding code.

- The editor creates a *dataset* dataframe with the fields you add.
 - The default aggregation is **Don't summarize**.
 - Similar to table visuals, fields are grouped and duplicate rows appear only once.
4. With the dataframe automatically generated by the fields you selected, you can write a Python script that results in plotting to the Python default device. When the script is complete, select the **Run** icon from the **Python script editor** title bar to run the script and generate the visual.

The screenshot shows the 'Python script editor' window. At the top, there's a toolbar with a play button (highlighted with a red box), settings, and other icons. Below the toolbar, a yellow status bar displays the message '⚠ Duplicate rows will be removed from the data.' with a close button. The main area contains a code editor with the following text:

```
1 # The following code to create a dataframe and remove duplicated rows is always
   # executed and acts as a preamble for your script:
2
3 # dataset = pandas.DataFrame(Age, Children, Fname, Gender, Pets, State, Weight)
4 # dataset = dataset.drop_duplicates()
5
6 # Paste or type your script code here:
```

Tips

- Your Python script can use only fields that are added to the **Values** section. You can add or remove fields while you work on your Python script. Power BI Desktop automatically detects field changes. As you select or remove fields from the **Values** section, supporting code in the Python script editor is automatically generated or removed.
- In some cases, you might not want automatic grouping to occur, or you might want all rows to appear, including duplicates. In those cases, you can add an index field to your dataset that causes all rows to be considered unique and prevents grouping.
- You can access columns in the dataset by using their names. For example, you can code `dataset["Age"]` in your Python script to access the age field.
- Power BI Desktop replots the visual when you select **Run** from the **Python script editor** title bar, or whenever a data change occurs due to data refresh, filtering, or highlighting.
- When you run a Python script that results in an error, the Python visual isn't plotted, and an error message appears on the canvas. For error details, select **See details** in the message.
- To get a larger view of the visualizations, you can minimize the **Python script editor**.

Create a scatter plot

Create a scatter plot to see if there's a correlation between age and weight.

1. In the **Python script editor**, under **Paste or type your script code here**, enter this code:

Python

```
import matplotlib.pyplot as plt
dataset.plot(kind='scatter', x='Age', y='Weight', color='red')
plt.show()
```

Your Python script editor pane should now look like the following image:

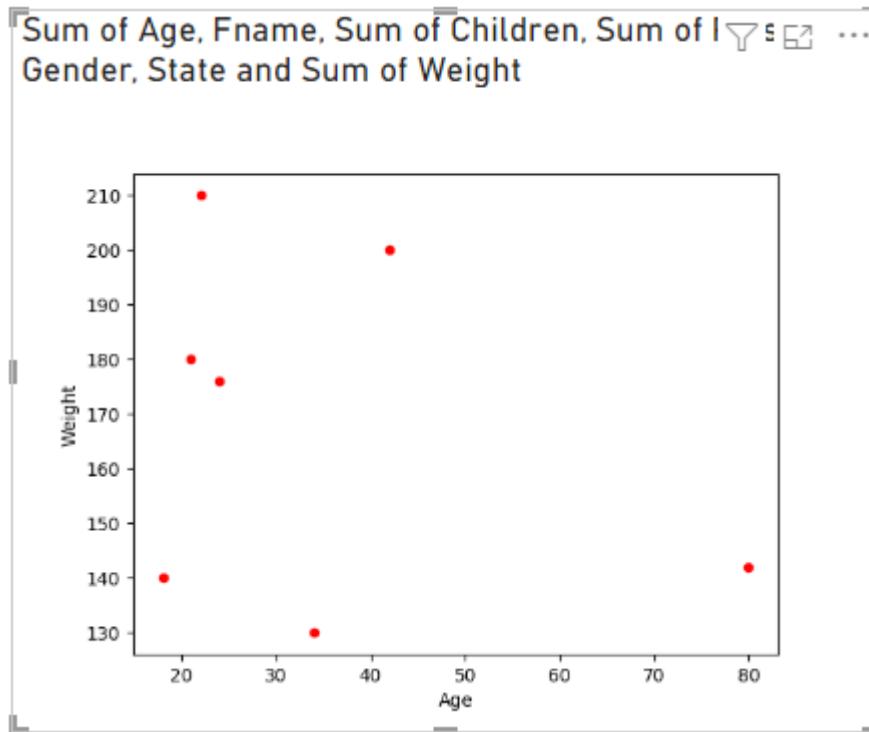
Python script editor

⚠ Duplicate rows will be removed from the data.

```
1 # The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:  
2  
3 # dataset = pandas.DataFrame(Age, Children, Fname, Gender, Pets, State, Weight)  
4 # dataset = dataset.drop_duplicates()  
5  
6 # Paste or type your script code here:  
7 import matplotlib.pyplot as plt  
8 dataset.plot(kind='scatter', x='Age', y='Weight', color='red')  
9 plt.show()
```

The code imports the Matplotlib library, which plots and creates the visual.

2. Select the **Run** script button to generate the following scatter plot in the Python visual.



Create a line plot with multiple columns

Create a line plot for each person that shows their number of children and pets.

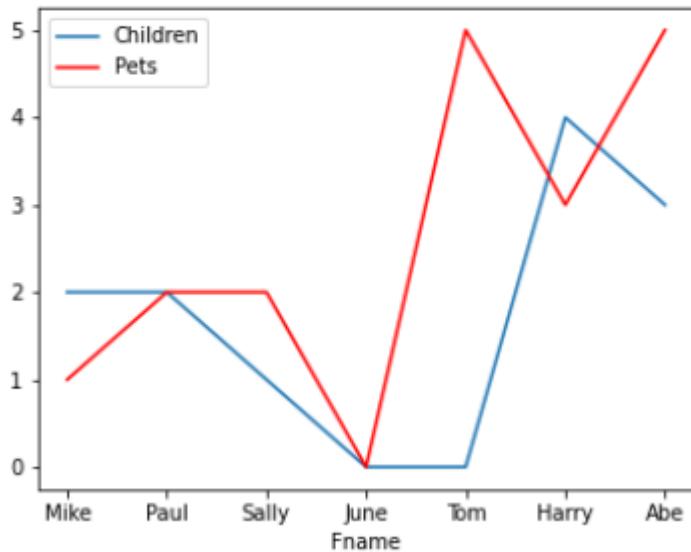
1. Under **Paste or type your script code here**, remove or comment out the previous code, and enter the following Python code:

Python

```
import matplotlib.pyplot as plt  
ax = plt.gca()  
dataset.plot(kind='line',x='Fname',y='Children',ax=ax)  
dataset.plot(kind='line',x='Fname',y='Pets', color='red', ax=ax)  
plt.show()
```

2. Select the **Run** button to generate the following line plot with multiple columns:

Sum of Age, Fname, Sum of Children, Sum of Pets, ...
Gender, State and Sum of Weight



Create a bar plot

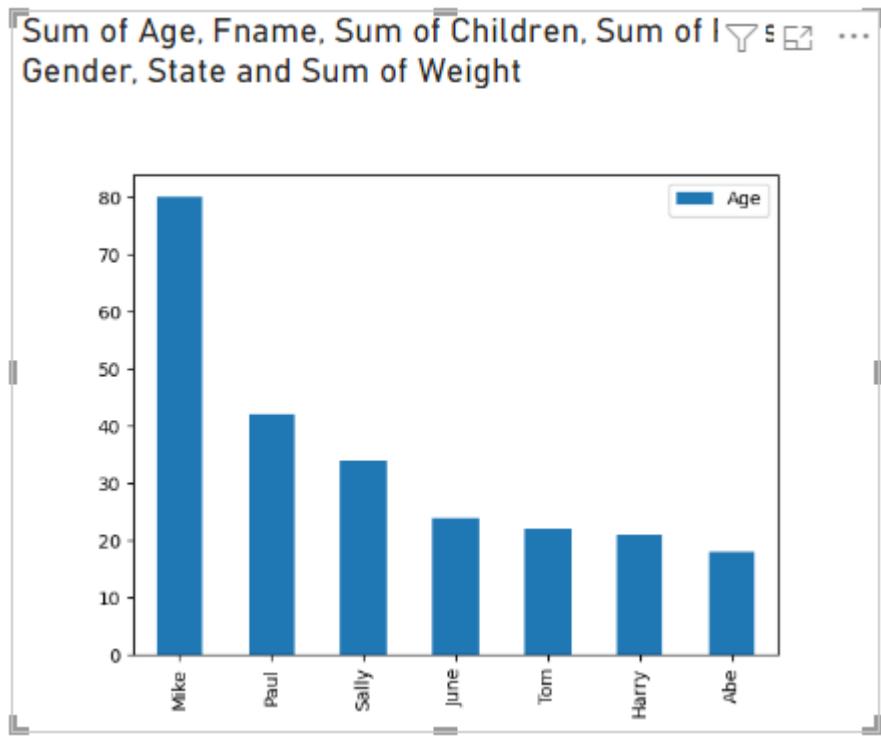
Create a bar plot for each person's age.

- Under **Paste or type your script code here**, remove or comment out the previous code, and enter the following Python code:

Python

```
import matplotlib.pyplot as plt
dataset.plot(kind='bar',x='Fname',y='Age')
plt.show()
```

- Select the **Run** button to generate the following bar plot:



Limitations

Python visuals in Power BI Desktop have the following limitations:

- The data the Python visual uses for plotting is limited to 150,000 rows. If more than 150,000 rows are selected, only the top 150,000 rows are used, and a message appears on the image. The input data also has a limit of 250 MB.
- If the input dataset of a Python visual has a column that contains a string value longer than 32,766 characters, that value is truncated.
- All Python visuals display at 72 DPI resolution.
- If a Python visual calculation exceeds five minutes, the execution times out, which results in an error.
- As with other Power BI Desktop visuals, if you select data fields from different tables with no defined relationship between them, an error occurs.
- Python visuals refresh upon data updates, filtering, and highlighting. The image itself isn't interactive.
- Python visuals respond to highlighting elements in other visuals, but you can't select elements in the Python visual to cross-filter other elements.
- Only plots to the Python default display device display correctly on the canvas. Avoid explicitly using a different Python display device.

- Python visuals don't support renaming input columns. Columns are referred to by their original names during script execution.

Security

Python visuals use Python scripts, which could contain code that has security or privacy risks. When you attempt to view or interact with a Python visual for the first time, you get a security warning. Enable Python visuals only if you trust the author and source, or after you review and understand the Python script.

Licensing

Python visuals require a [Power BI Pro](#) or Premium Per User (PPU) license to render in reports, refresh, filter, and cross-filter. Users of free Power BI can consume only tiles that are shared with them in Premium workspaces.

The following table describes Python visuals capabilities based on licensing.

	Author Python visuals in Power BI Desktop	Create Power BI service reports with Python visuals	View Python visuals in reports
Guest (Power BI embedded)	Supported	Not supported	Supported in Premium/Azure capacity only
Unmanaged tenant (domain not verified)	Supported	Not supported	Not supported
Managed tenant with free license	Supported	Not supported	Supported in Premium capacity only
Managed tenant with Pro or PPU license	Supported	Supported	Supported

For more information about Power BI Pro licenses and how they differ from free licenses, see [Purchase and assign Power BI Pro user licenses](#).

Next steps

This tutorial barely scratches the surface of the options and capabilities for creating visual reports using by Python, pandas, and the Matplotlib library. For more information, see the following resources:

- Documentation at the [Matplotlib](#) website.
- [Matplotlib Tutorial : A Basic Guide to Use Matplotlib with Python](#)
- [Matplotlib Tutorial – Python Matplotlib Library with Examples](#)
- [Pandas API Reference](#)
- [Python visualizations in Power BI service](#)
- [Using Python Visuals in Power BI](#)
- [Comprehensive Python Scripting Tutorial](#)

For more information about Python in Power BI, see:

- [Run Python Scripts in Power BI Desktop](#)
- [Use an external Python IDE with Power BI](#)

Learn which Python packages are supported in Power BI

Article • 01/13/2023

You can use the powerful [Python programming language](#) to create visuals in Power BI. Many Python packages are supported in Power BI and more are being supported all the time.

The following sections provide an alphabetical table of which Python packages are supported in Power BI.

Request support for a new Python package

Supported Python packages for Power BI are found in the [following section](#). If you would like to request support of a Python package not found in that list, submit your request to [Power BI Ideas](#).

Requirements and limitations of Python packages

There are a handful of requirements and limitations for Python packages:

- Current Python runtime: Python 3.7.7.
- Power BI, for the most part, supports Python packages with free and open-source software licenses such as GPL-2, GPL-3, MIT+, and so on.
- Power BI supports packages published in PyPI. The service doesn't support private or custom Python packages. Users are encouraged to make their private packages available on PyPI prior to requesting the package be available in Power BI.
- For Python visuals in Power BI Desktop, you can install any package, including custom Python packages.
- For security and privacy reasons, Python packages that provide client-server queries over the web in the service, aren't supported. Networking is blocked for such attempts.
- The approval process for including a new Python package has a tree of dependencies. Some dependencies required to be installed in the service can't be supported.

Python packages that are supported in Power BI

The following table shows which packages are supported in Power BI.

Package	Version	Link
cycler	0.11.0	https://pypi.org/project/cycler ↗
joblib	1.1.0	https://pypi.org/project/joblib ↗
kiwisolver	1.4.4	https://pypi.org/project/kiwisolver ↗
matplotlib	3.2.2	https://pypi.org/project/matplotlib ↗
numpy	1.21.6	https://pypi.org/project/numpy ↗
packaging	21.3	https://pypi.org/project/packaging ↗
pandas	1.3.5	https://pypi.org/project/pandas ↗
patsy	0.5.2	https://pypi.org/project/patsy ↗
pip	22.1.2	https://pypi.org/project/pip ↗
pyparsing	3.0.9	https://pypi.org/project/pyparsing ↗
python-dateutil	2.8.2	https://pypi.org/project/python-dateutil ↗
pytz	2022.1	https://pypi.org/project/pytz ↗
scikit-learn	1.0.2	https://pypi.org/project/scikit-learn ↗
scipy	1.7.3	https://pypi.org/project/scipy ↗
seaborn	0.11.2	https://pypi.org/project/seaborn ↗
setuptools	63.2.0	https://pypi.org/project/setuptools ↗
six	1.16.0	https://pypi.org/project/six ↗
statsmodels	0.13.2	https://pypi.org/project/statsmodels ↗
threadpoolctl	3.1.0	https://pypi.org/project/threadpoolctl ↗
typing-extensions	4.3.0	https://pypi.org/project/typing-extensions ↗
xgboost	1.6.1	https://pypi.org/project/xgboost ↗

Next steps

For more information about Python in Power BI, take a look at the following articles:

- [Create Power BI visuals using Python](#)
- [Running Python scripts in Power BI Desktop](#)
- [Using Python in Query Editor](#)

Run R scripts in Power BI Desktop

Article • 04/14/2023

You can run R scripts directly in Power BI Desktop and import the resulting datasets into a Power BI Desktop data model.

Install R

To run R scripts in Power BI Desktop, you need to install R on your local machine. You can download and install R for free from many locations, including the [Microsoft R Application Network](#) and the [CRAN Repository](#). The current release supports Unicode characters and spaces (empty characters) in the installation path.

Run R scripts

Using just a few steps in Power BI Desktop, you can run R scripts and create a data model. With the data model, you can create reports and share them on the Power BI service. R scripting in Power BI Desktop now supports number formats that contain decimals (.) and commas (,).

Prepare an R script

To run an R script in Power BI Desktop, create the script in your local R development environment, and make sure it runs successfully.

To run the script in Power BI Desktop, make sure the script runs successfully in a new and unmodified workspace. This prerequisite means that all packages and dependencies must be explicitly loaded and run. You can use `source()` to run dependent scripts.

When you prepare and run an R script in Power BI Desktop, there are a few limitations:

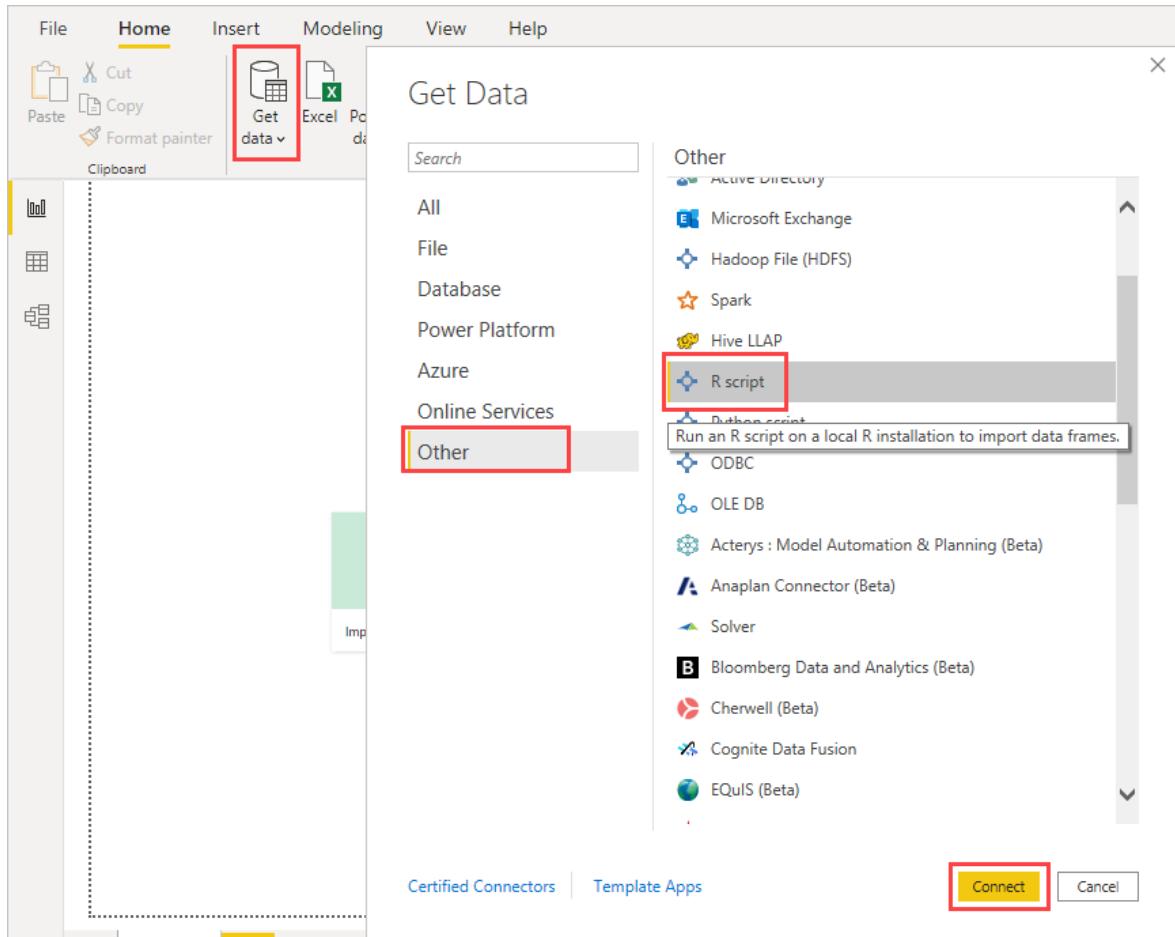
- Because only data frames are imported, remember to represent the data you want to import to Power BI in a data frame.
- Columns typed as *Complex* and *Vector* aren't imported, and they're replaced with error values in the created table.
- Values of `N/A` are translated to `NULL` values in Power BI Desktop.
- If an R script runs longer than 30 minutes, it times out.
- Interactive calls in the R script, such as waiting for user input, halt the script's execution.

- When setting the working directory within the R script, you *must* define a full path to the working directory, rather than a relative path.
- R scripts cannot run in the Power BI service.

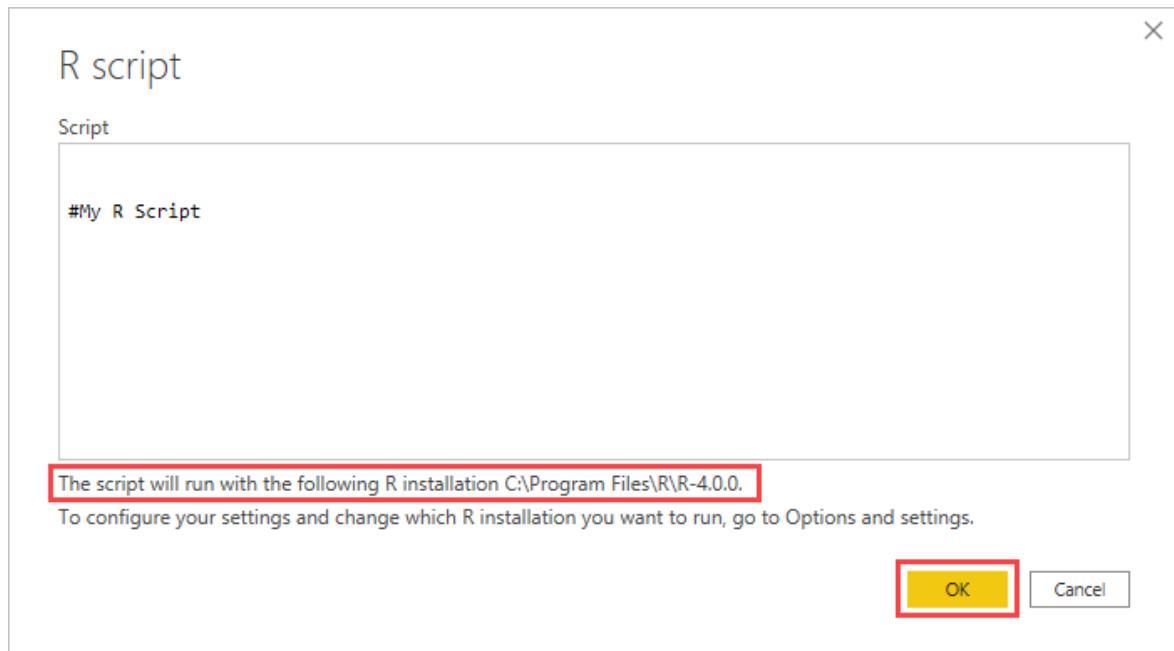
Run your R script and import data

Now you can run your R script to import data into Power BI Desktop:

1. In Power BI Desktop, select **Get data**, choose **Other > R script**, and then select **Connect**:

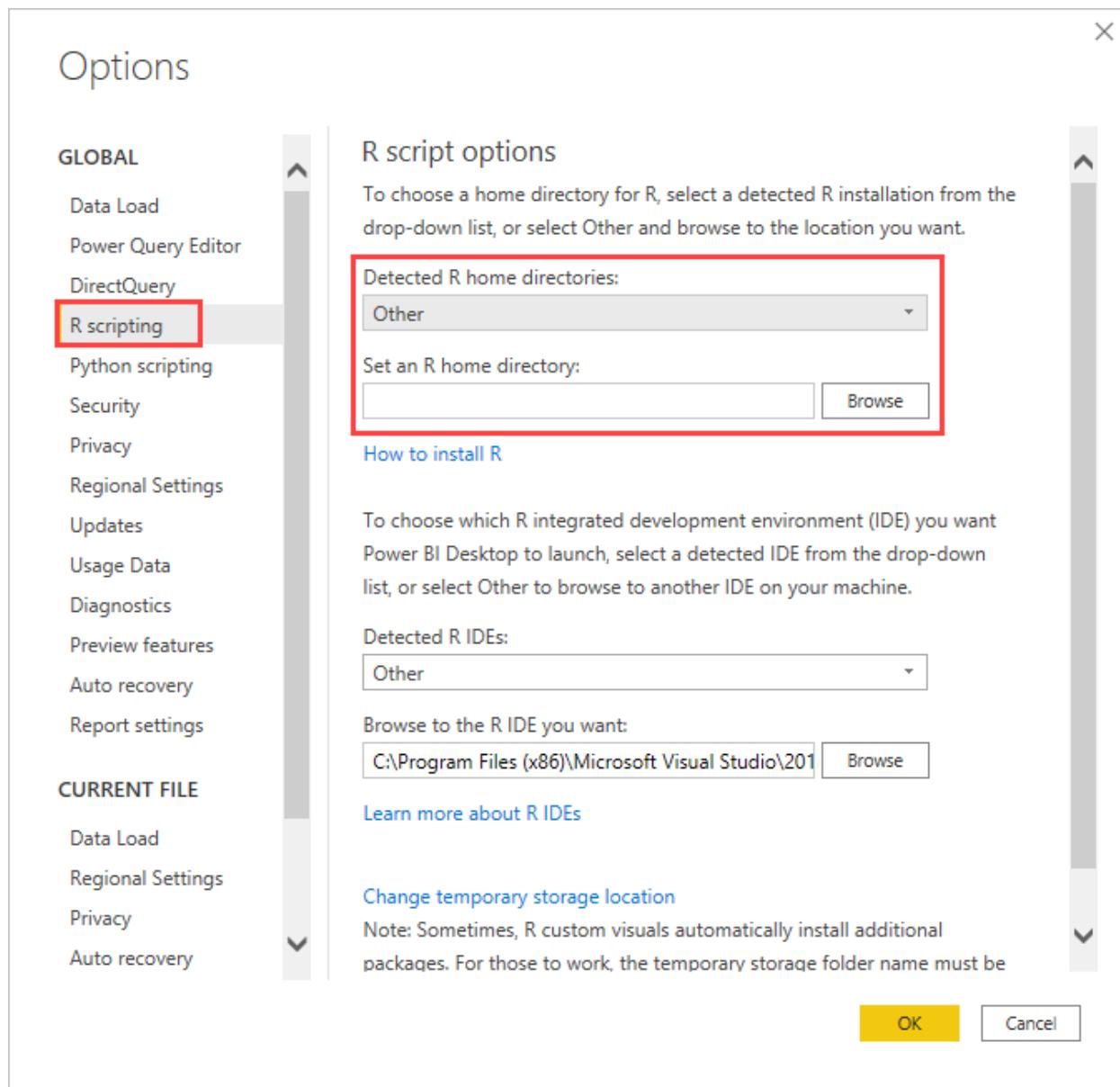


2. If R is installed on your local machine, just copy your script into the script window and select **OK**. The latest installed version is displayed as your R engine.



3. Select **OK** to run the R Script. When the script runs successfully, you can then choose the resulting data frames to add to the Power BI model.

You can control which R installation to use to run your script. To specify your R installation settings, choose **File > Options and settings > Options**, then select **R scripting**. Under **R script options**, the **Detected R home directories** dropdown list shows your current R installation choices. If the R installation you want isn't listed, pick **Other**, and then browse to or enter your preferred R installation folder in **Set an R home directory**.



Refresh

You can refresh an R script in Power BI Desktop. When you refresh an R script, Power BI Desktop runs the R script again in the Power BI Desktop environment.

Next steps

Take a look at the following additional information about R in Power BI.

- [Create Power BI visuals using R](#)
- [Use an external R IDE with Power BI](#)

Use R in Power Query Editor

Article • 08/02/2023

The R language is a powerful programming language that many statisticians, data scientists, and data analysts use. You can use R in Power BI Desktop's Power Query Editor to:

- Prepare data models.
- Create reports.
- Do data cleansing, advanced data shaping, and dataset analytics, which include missing data completion, predictions, clustering, and more.

Install R

You can download R for free from the [CRAN Repository](#).

Install mice

As a prerequisite, you must install the [mice library](#) in your R environment. Without mice, the sample script code doesn't work properly. The mice package implements a method to deal with missing data.

To install the mice library:

1. Launch the `R.exe` program, for example, `C:\Program Files\Microsoft\R Open\R-3.5.3\bin\R.exe`.
2. Run the install command from the R prompt:

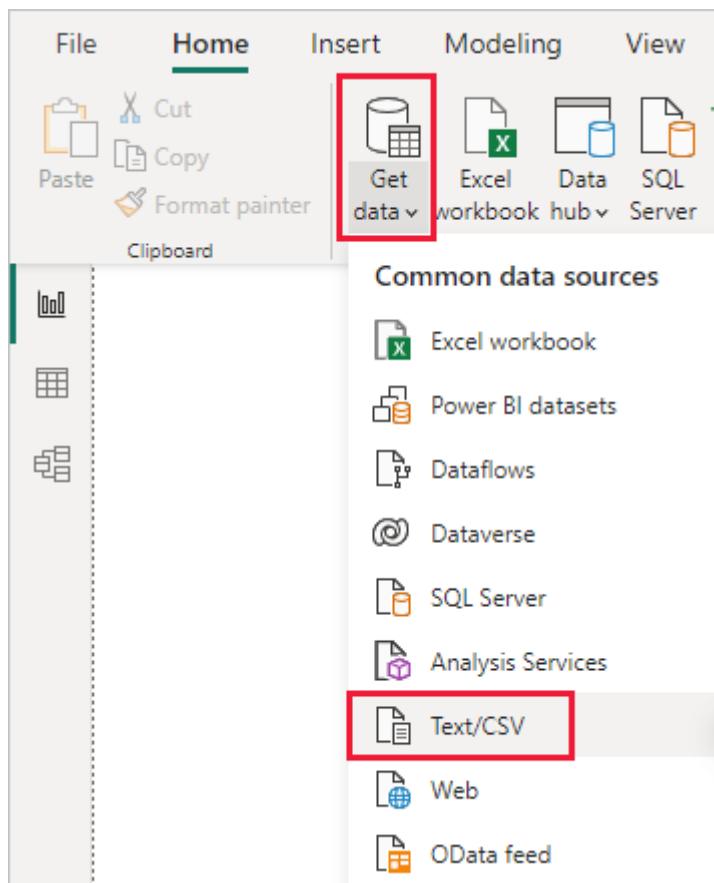
```
R  
install.packages('mice')
```

Use an R script in Power Query Editor

To demonstrate using R in Power Query Editor, this example uses a stock market dataset contained in a .csv file.

1. [Download the EuStockMarkets_NA.csv file](#). Remember where you save it.

2. Load the file into Power BI Desktop. From the Home tab, select Get data > Text/CSV.



3. Select the *EuStockMarkets_NA.csv* file, and then choose Open. The CSV data is displayed in the Text/CSV file dialog.

A screenshot of the 'Text/CSV' file dialog window titled 'EuStockMarkets_NA.csv'. The window contains three tabs at the top: 'File Origin' (set to '1252: Western European (Windows)'), 'Delimiter' (set to 'Comma'), and 'Data Type Detection' (set to 'Based on first 200 rows'). Below these tabs is a preview table with the following data:

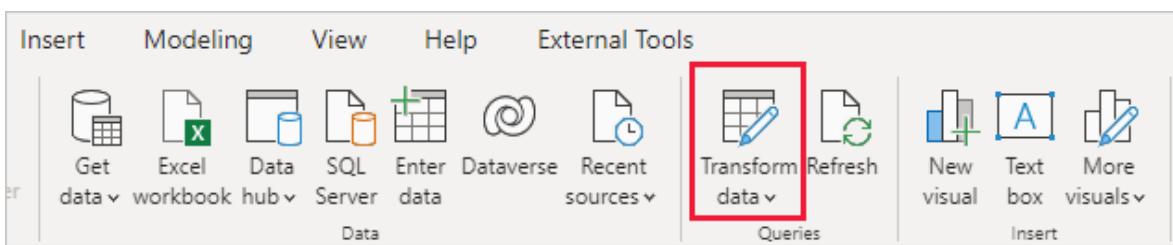
Day	SMI	SMI missing values
1	1678.1	1678.1
2	1688.5	1688.5
3	1678.6	1678.6
4	1684.1	1684.1
5	1686.6	1686.6
6	1671.6	1671.6
7	1682.9	1682.9
8	1703.6	1703.6
9	1697.5	1697.5

At the bottom of the dialog are three buttons: 'Extract Table Using Examples' (disabled), 'Load' (highlighted in green), and 'Transform Data'.

4. Select Load to load the data from the file. After Power BI Desktop has loaded the data, the new table appears in the Fields pane.

The screenshot shows the 'Fields' pane in Power BI. A red box highlights the 'EuStockMarkets_NA' table. Inside the table, there are three columns: 'Day', 'SMI', and 'SMI missing values'. The 'SMI missing values' column is specifically highlighted with a red box.

5. To open Power Query Editor, from the Home ribbon select Transform data.



6. From the Transform tab, select Run R script. The Run R script editor appears. Rows 15 and 20 have missing data, as do other rows you can't see in the image. The following steps show how R completes those rows for you.

The screenshot shows the Power Query Editor window. The 'Run R script' button in the ribbon is highlighted with a red box. The preview pane displays a table with three columns: 'Day', 'SMI', and 'SMI missing values'. The table has 12 rows of data. The 'Properties' pane on the right shows the table is named 'EuStockMarkets_NA' and the last step applied was 'Changed Type'.

	Day	SMI	SMI missing values
1	1	1678.1	1678.1
2	2	1688.5	1688.5
3	3	1678.6	1678.6
4	4	1684.1	1684.1
5	5	1686.6	1686.6
6	6	1671.6	1671.6
7	7	1682.9	1682.9
8	8	1703.6	1703.6
9	9	1697.5	1697.5
10	10	1716.3	1716.3
11	11	1723.8	1723.8
12	12	1730.5	1730.5

7. For this example, enter the following script code in the Script box of the Run R script window. Replace <Your File Path> with the path to *EuStockMarkets_NA.csv* on your local file system, for example, *C:/Users/admin/Documents/Microsoft/EuStockMarkets_NA.csv*.

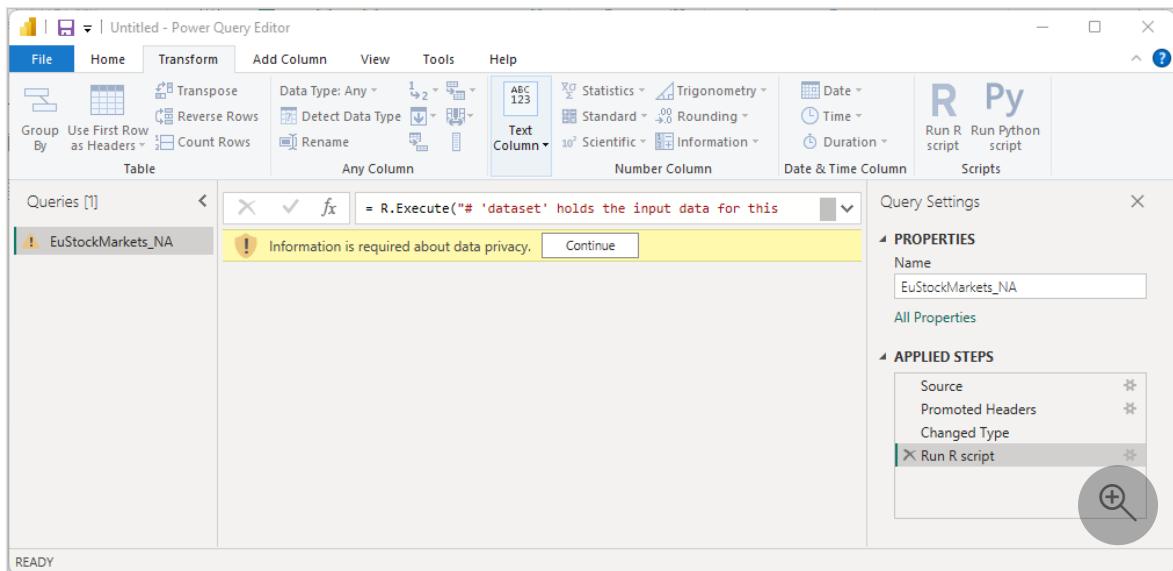
R

```
library(mice)
tempData <- mice(dataset, m=1, maxit=50, meth='pmm', seed=100)
completedData <- complete(tempData, 1)
output <- dataset
output$completedValues <- completedData$"SMI missing values"
```

ⓘ Note

You might need to overwrite a variable named *output* to properly create the new dataset with the filters applied.

8. Select **OK**. Power Query Editor displays a warning about data privacy.



9. Inside the warning message, select **Continue**. In the **Privacy levels** dialog that appears, set all data sources to **Public** for the R scripts to work properly in the Power BI service.

Privacy levels

The privacy level is used to ensure data is combined without undesirable data transfer. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. More information on privacy levels can be found [here](#).

Ignore Privacy Levels checks for this file. Ignoring Privacy Levels could expose sensitive or confidential data to an unauthorized person.



For more information about privacy settings and their implications, see [Power BI Desktop privacy levels](#).

10. Select **Save** to run the script.

When you run the script, you see the following result:

	Name	Type
1	completedData	Table
2	output	Table

When you select **Table** next to **Output** in the table that appears, the table is presented, as shown in the following image.

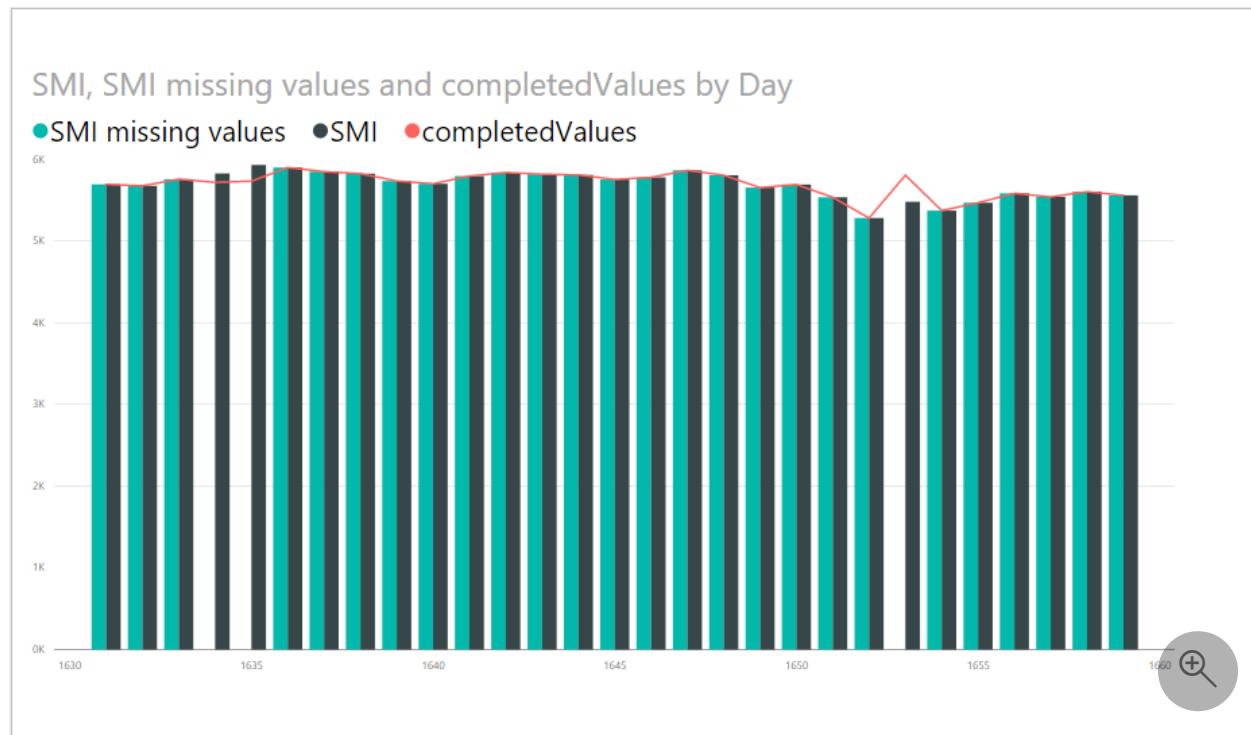
Queries [1]				
	X	✓	fx	= # "Run R script" {[Name = "output"]}[Value]
EuStockMarkets_NA				
	123 Day	1.2 SMI	1.2 SMI missing values	1.2 completedValues
1	1	1678.1	1678.1	1678.1
2	2	1688.5	1688.5	1688.5
3	3	1678.6	1678.6	1678.6
4	4	1684.1	1684.1	1684.1
5	5	1686.6	1686.6	1686.6
6	6	1671.6	1671.6	1671.6
7	7	1682.9	1682.9	1682.9
8	8	1703.6	1703.6	1703.6
9	9	1697.5	1697.5	1697.5
10	10	1716.3	1716.3	1716.3
11	11	1723.8	1723.8	1723.8
12	12	1730.5	1730.5	1730.5
13	13	1727.4	1727.4	1727.4
14	14	1733.3	1733.3	1733.3
15	15	1734	null	1723.8
16	16	1728.3	1728.3	1728.3
17	17	1737.1	1737.1	1737.1
18	18	1723.1	1723.1	1723.1
19	19	1723.6	1723.6	1723.6
20	20	1719	null	1723.1
21	21	1721.2	1721.2	1721.2
22	22	1725.3	1725.3	1725.3
23	23	1727.2	1727.2	1727.2
24	24	1727.2	1727.2	1727.2

Notice the new column in the **Fields** pane called **completedValues**. The SMI missing values column has a few missing data elements. Take a look at how R handles that in the next section.

With just five lines of R script, Power Query Editor filled in the missing values with a predictive model.

Create visuals from R script data

We can now create a visual to see how the R script code with the mice library completes the missing values.



You can save all completed visuals in one Power BI Desktop .pbix file and use the data model and its R scripts in the Power BI service.

ⓘ Note

You can download a [.pbix file](#) with all these steps completed.

After you've uploaded the .pbix file to the Power BI service, you need to take other steps to enable service data refresh and updated visuals:

- **Enable scheduled refresh for the dataset:** To enable scheduled refresh for the workbook containing your dataset with R scripts, see [Configuring scheduled refresh](#). This article also includes information about on-premises data gateways.

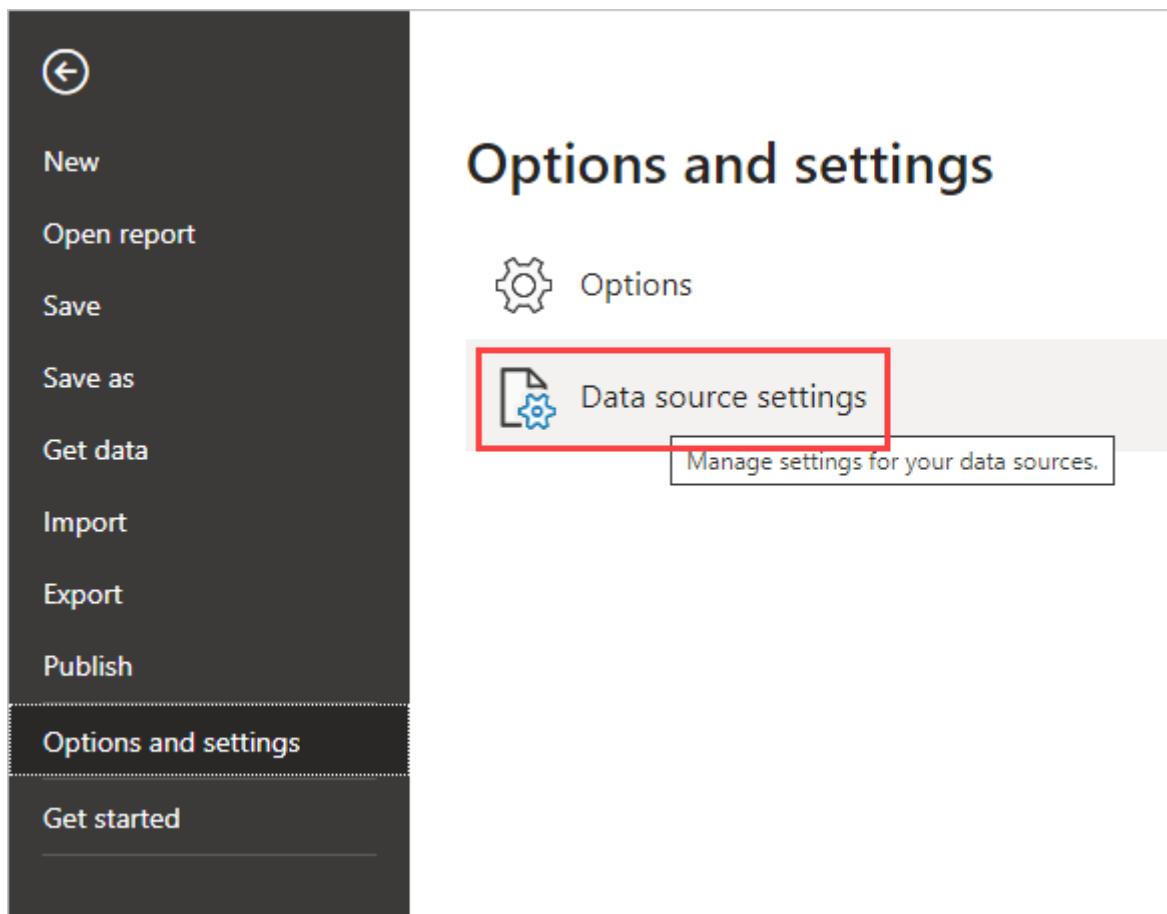
- **Install a gateway:** You need an on-premises data gateway (personal mode) installed on the machine where the file and R are located. The Power BI service accesses that workbook and re-renders any updated visuals. For more information, see [use personal gateways in Power BI](#).

Considerations and limitations

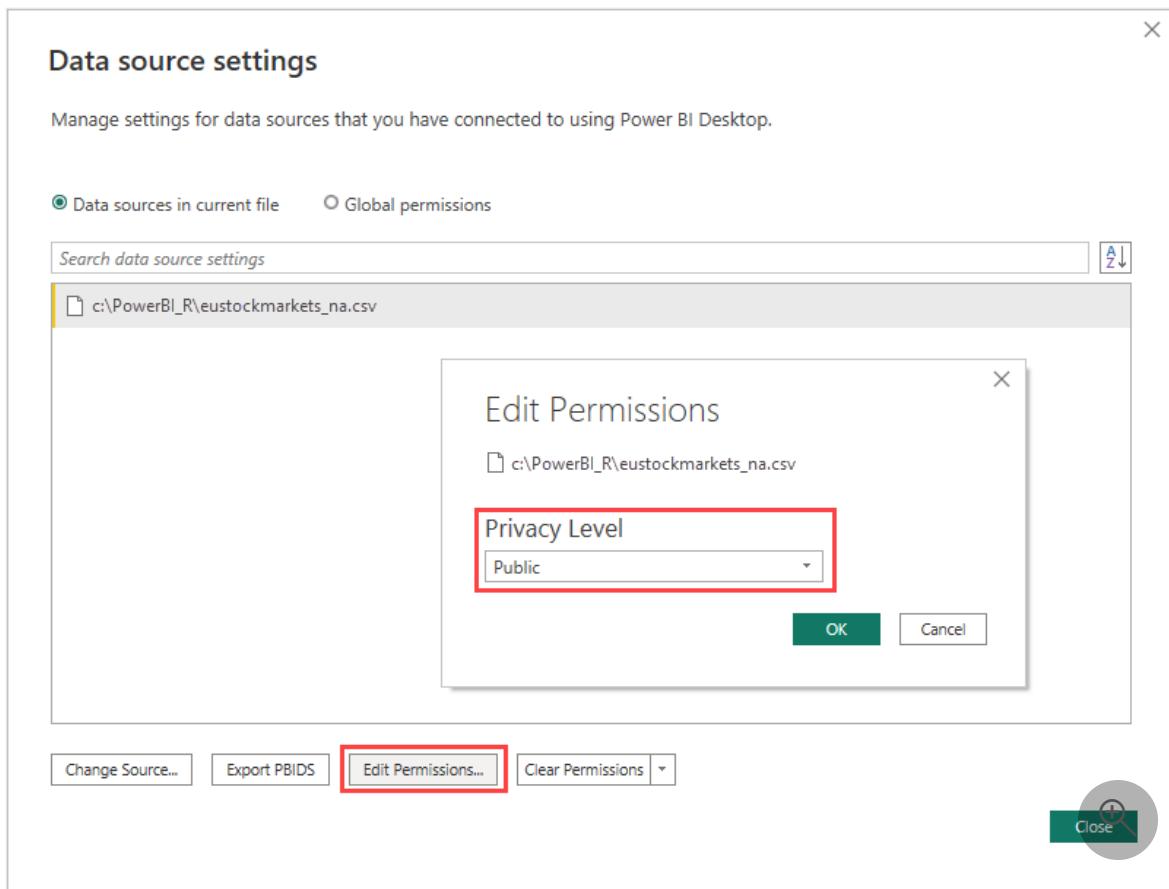
There are some limitations to queries that include R scripts created in Power Query Editor:

- All R data source settings must be set to **Public**. All other steps in a Power Query Editor query must also be public.

To get to the data source settings, in Power BI Desktop, select **File > Options and settings > Data source settings**.



In the **Data source settings** dialog, select one or more data sources, and then select **Edit Permissions**. Set the **Privacy Level** to **Public**.



- To schedule refresh of your R visuals or dataset, enable scheduled refresh and install an on-premises data gateway (personal mode) on the computer containing the workbook and R. You can't use an enterprise gateway to refresh datasets containing R scripts in Power Query.

Next Steps

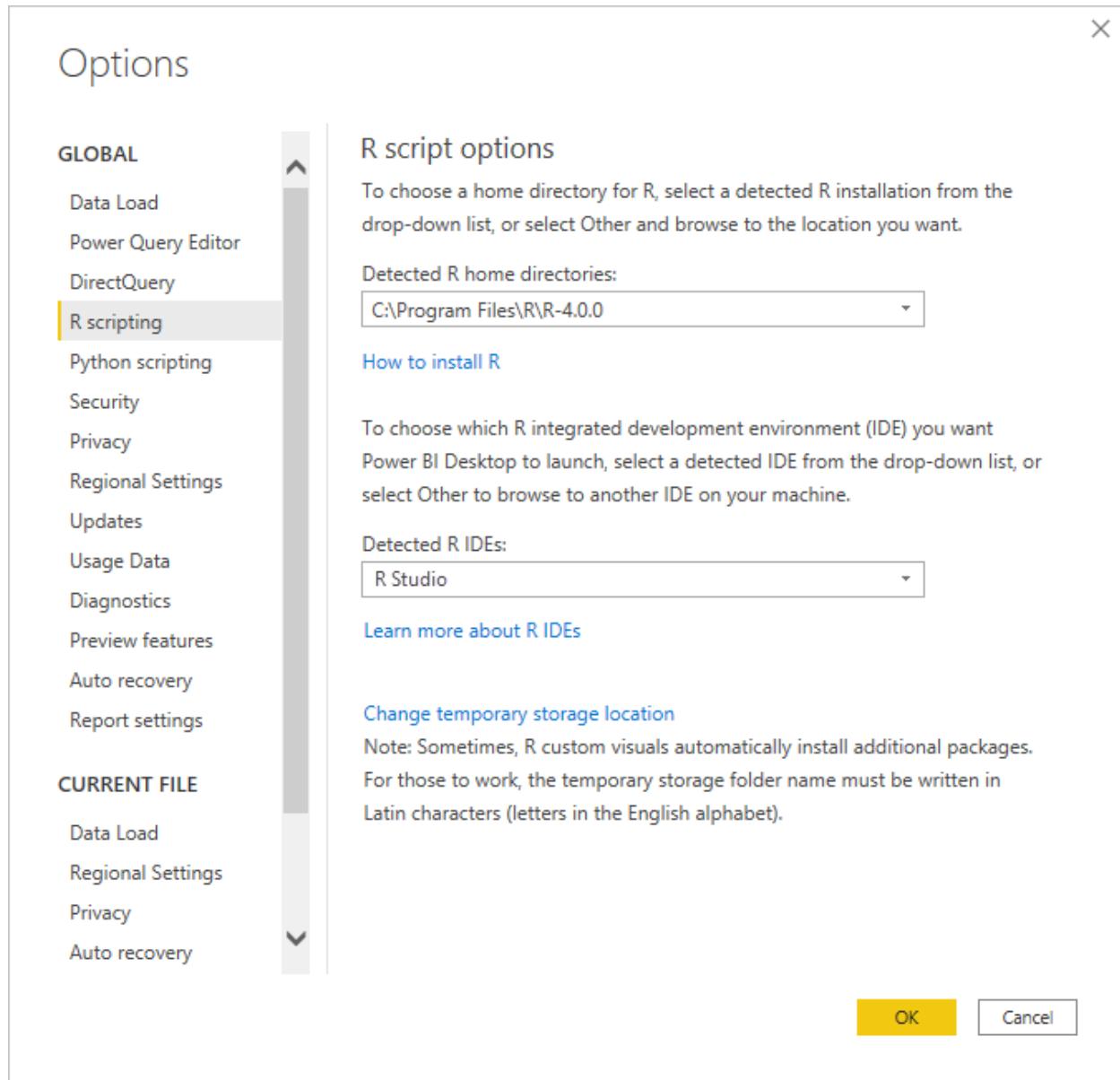
There are all sorts of things you can do with R and custom queries. Explore and shape your data just the way you want it to appear.

- [Run R scripts in Power BI Desktop](#)
- [Use an external R IDE with Power BI](#)
- [Create visuals by using R packages in the Power BI service](#)

Use an external R IDE with Power BI

Article • 12/07/2021

With **Power BI Desktop**, you can use your external R IDE (Integrated Development Environment) to create and refine R scripts, then use those scripts in Power BI.



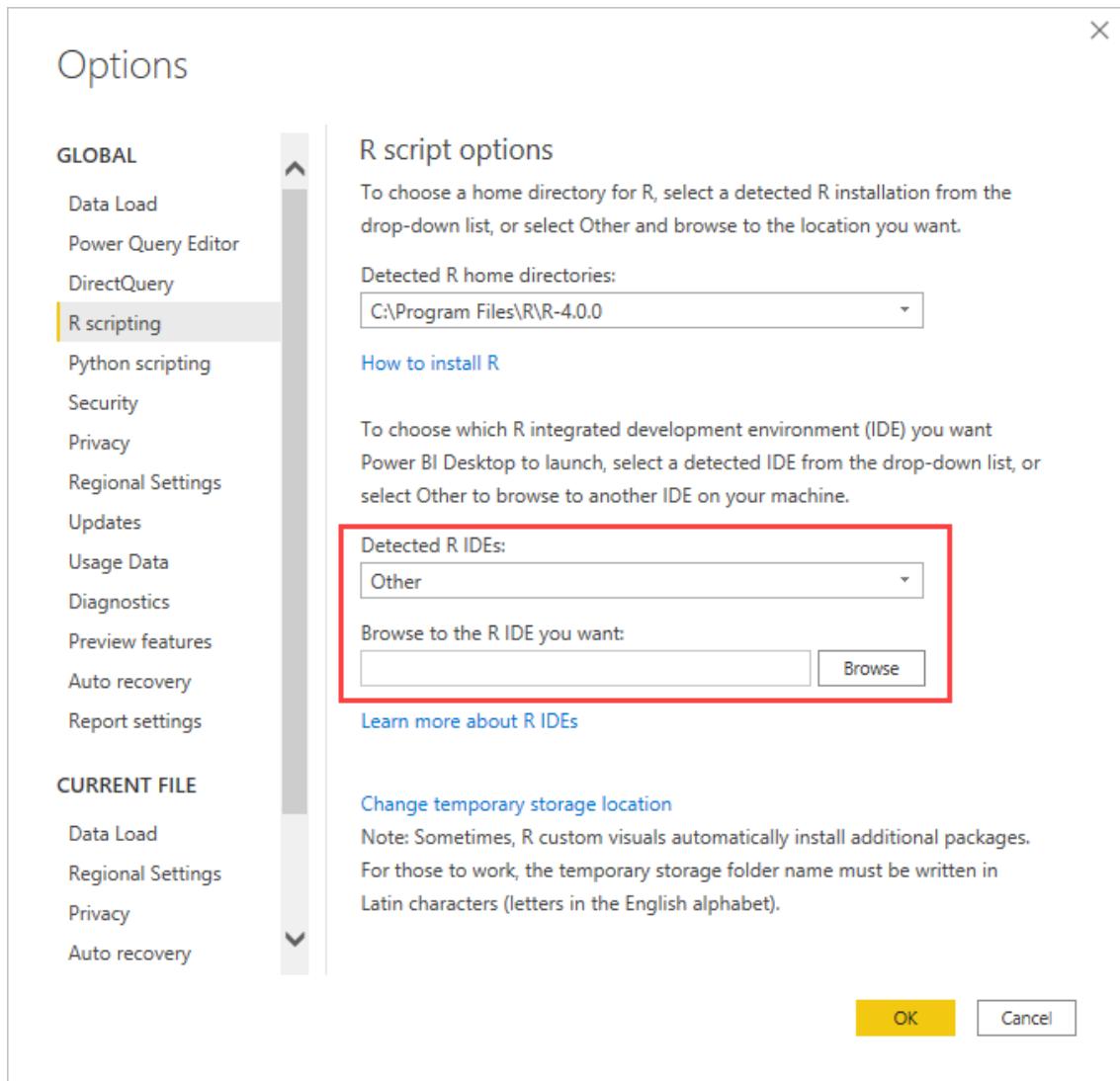
Enable an external R IDE

Launch your external R IDE from **Power BI Desktop** and have your data automatically imported and displayed in the R IDE. From there, you can modify the script in that external R IDE, then paste it back into **Power BI Desktop** to create Power BI visuals and reports. Specify which R IDE you would like to use, and have it launch automatically from within **Power BI Desktop**.

Requirements

To use this feature, you need to install an **R IDE** on your local computer. **Power BI Desktop** does not include, deploy, or install the R engine, so you must separately install R on your local computer. You can choose which R IDE to use, with the following options:

- You can install your favorite R IDE, many of which are available for free, such as the [Revolution Open download page](#), and the [CRAN Repository](#).
- **Power BI Desktop** also supports [R Studio](#) and [Visual Studio 2015](#) with [*R Tools for Visual Studio*](#) editors.
- You can also install a different R IDE and have **Power BI Desktop** launch that R IDE by doing one of the following:
 - You can associate .R files with the external IDE you want **Power BI Desktop** to launch.
 - You can specify the .exe that **Power BI Desktop** should launch by selecting *Other* from the **R Script Options** section of the **Options** dialog. You can bring up the **Options** dialog by going to **File > Options and settings > Options**.



If you have multiple R IDEs installed, you can specify which will be launched by selecting it from the *Detected R IDEs* drop-down in the **Options** dialog.

By default, **Power BI Desktop** will launch **R Studio** as the external R IDE if it's installed on your local computer; if **R Studio** is not installed and you have **Visual Studio 2015** with **R Tools for Visual Studio**, that will be launched instead. If neither of those R IDEs is installed, the application associated with **.R** files is launched.

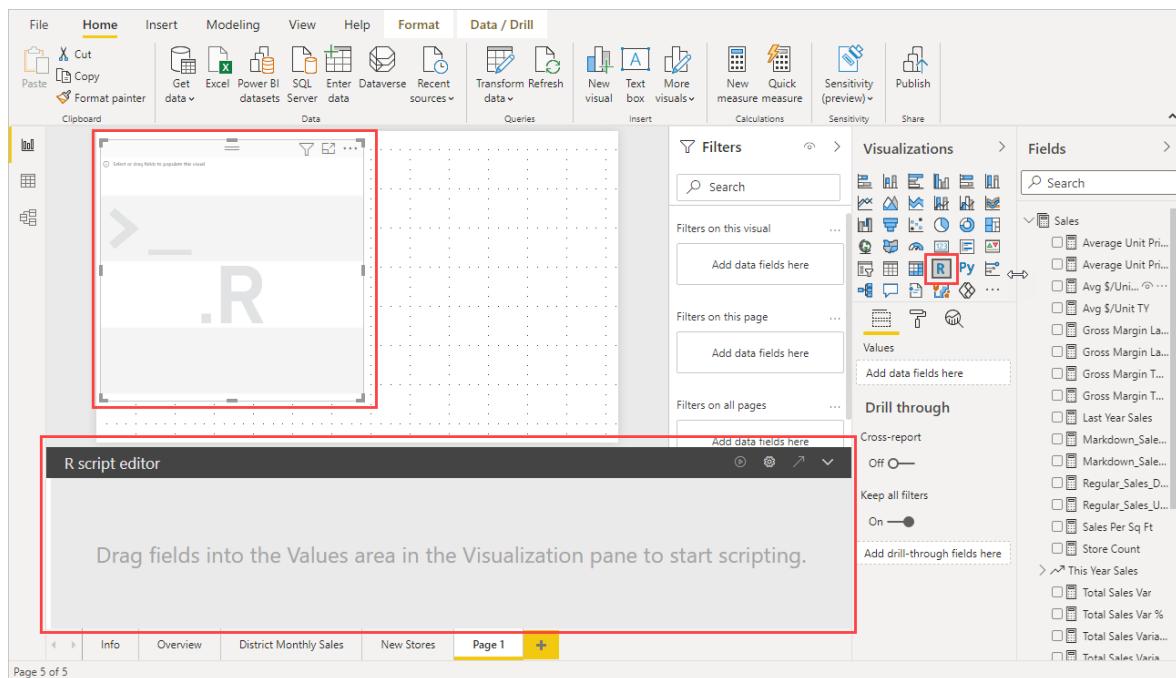
And if no **.R** file association exists, it's possible to specify a path to a custom IDE in the *Browse to your preferred R IDE* section of the **Options** dialog. You can also launch a different R IDE by selecting the **Settings** gear icon beside the **Edit script in external IDE** arrow icon, in **Power BI Desktop**.

Launch an R IDE from Power BI Desktop

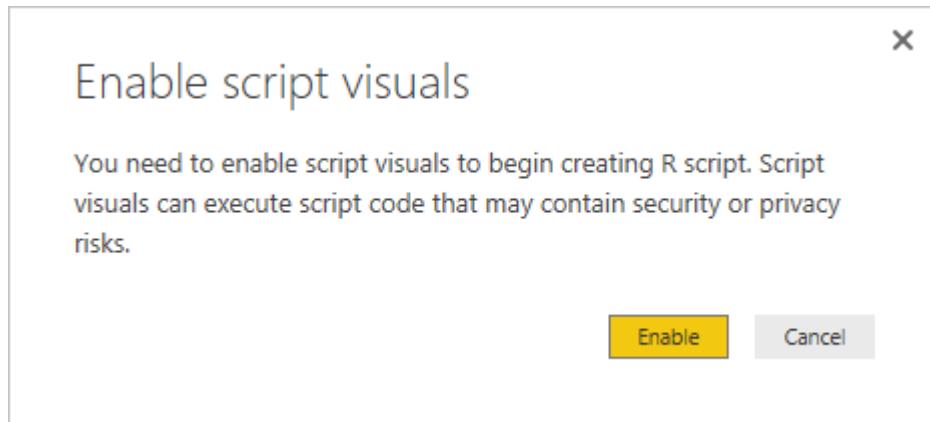
To launch an R IDE from **Power BI Desktop**, take the following steps:

1. Load data into **Power BI Desktop**.

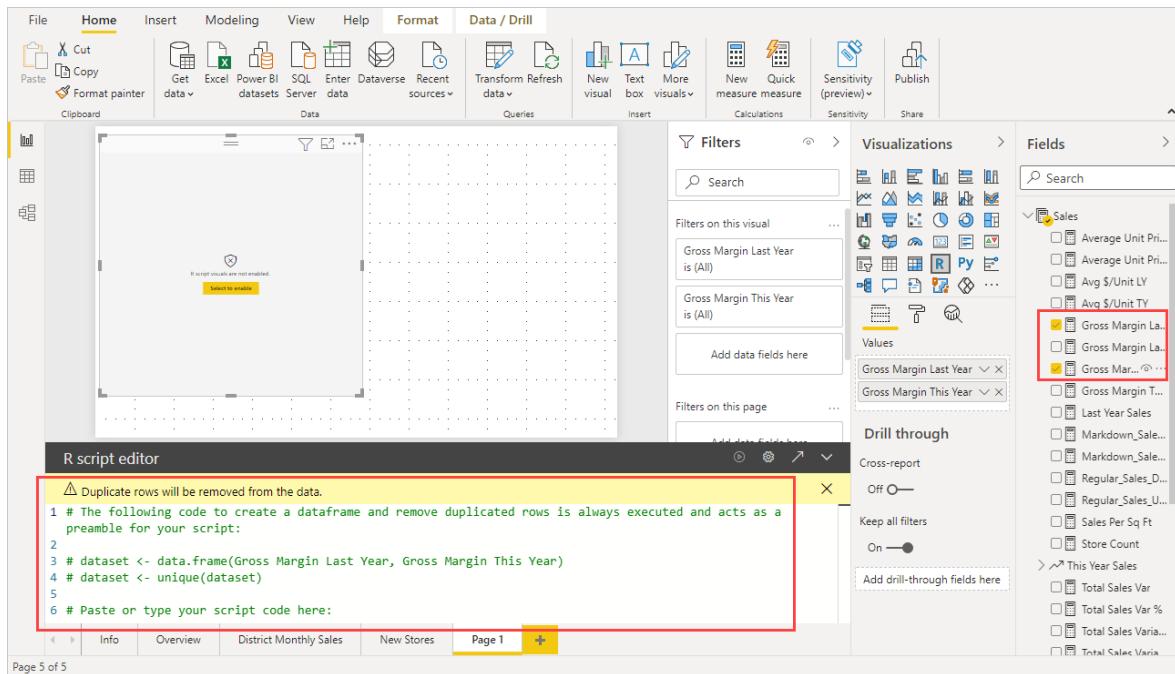
2. When script visuals are enabled, you can select an R visual from the **Visualizations** pane, which creates a blank R visual that's ready to display the results of your script. The **R script editor** pane also appears.



3. Select some fields from the **Fields** pane that you want to work with. If you haven't enabled script visuals yet, you'll be prompted to do so.



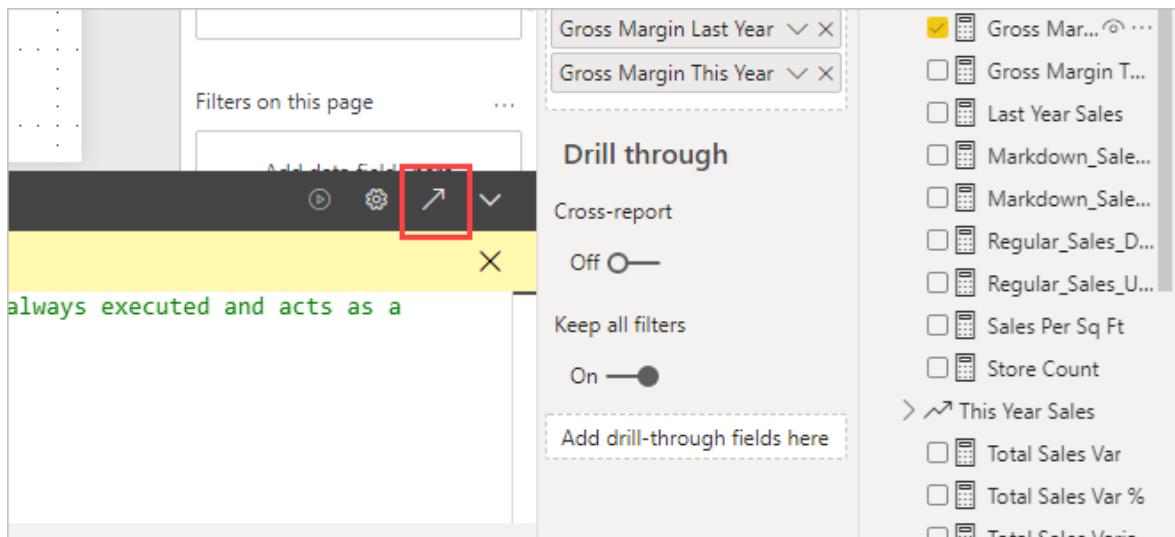
4. Now you can select the fields you want to use in your R script. When you select a field, the **R script editor** field automatically creates script code based on the field or fields you select. You can either create (or paste) your R script directly in the **R script editor** pane, or you can leave it empty.



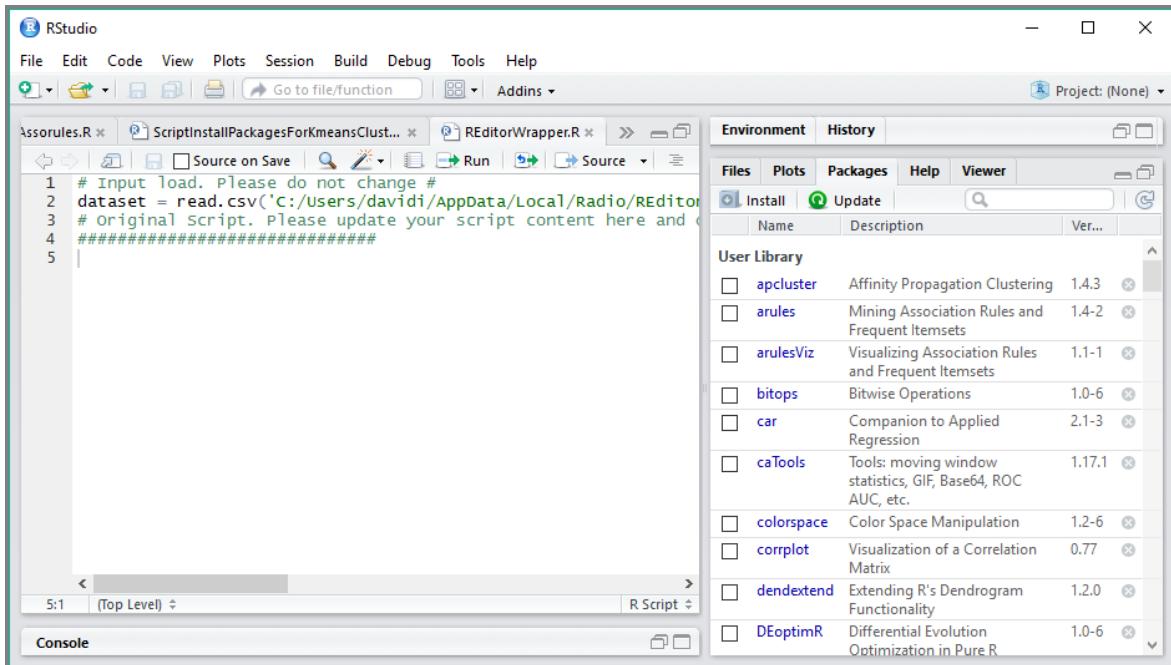
➊ Note

The default aggregation type for R visuals is *do not summarize*.

5. You can now launch your R IDE directly from **Power BI Desktop**. Select the **Edit script in external IDE** button, found on the right side of the **R script editor** title bar, as shown below.



6. Your specified R IDE is launched by Power BI Desktop, as shown in the following image (in this image, RStudio is the default R IDE).



(!) Note

Power BI Desktop adds the first three lines of the script so it can import your data from **Power BI Desktop** once you run the script.

7. Any script you created in the **R script editor pane** of **Power BI Desktop** appears starting in line 4 in your R IDE. At this point, you can create your R script in the R IDE. Once your R script is complete in your R IDE, you need to copy and paste it back into the **R script editor pane** in **Power BI Desktop**, *excluding* the first three lines of the script that **Power BI Desktop** automatically generated. Do not copy the first three lines of script back into **Power BI Desktop**, those lines were only used to import your data to your R IDE from **Power BI Desktop**.

Known limitations

Launching an R IDE directly from Power BI Desktop has a few limitations:

- Automatically exporting your script from your R IDE into **Power BI Desktop** is not supported.
- **R Client** editor (RGui.exe) is not supported, because the editor itself does not support opening files.

Next steps

Take a look at the following additional information about R in Power BI.

- Running R Scripts in Power BI Desktop
- Create Power BI visuals using R

Create visuals by using R packages in the Power BI service

Article • 01/25/2023

You can use the powerful [R programming language](#) to create visuals in the Power BI service. Many R packages are supported in the Power BI service and more are being supported all the time. Some packages aren't supported.

The following sections provide an alphabetical table of which R packages are supported in Power BI, and which aren't. For more information about R in Power BI, see the [R visuals](#) article.

Request support for a new R package

Supported R packages for the Power BI service are found in [the following section](#). If you would like to request support of an R package not found in that list, submit your request to [Power BI Ideas](#).

Requirements and limitations of R packages

There are a handful of requirements and limitations for R packages:

- Current R runtime: Microsoft R 3.4.4
- The Power BI service usually supports R packages with free and open-source software licenses such as GPL-2, GPL-3, MIT+, and so on.
- The Power BI service supports packages published in the Comprehensive R Archive Network (CRAN). The service doesn't support private or custom R packages. Users are encouraged to make their private packages available on CRAN prior to requesting the package be available in the Power BI service.
- The Power BI Desktop has two variations for R packages:
 - For R visuals, you can install any package, including custom R packages
 - For Custom R visuals, only public CRAN packages are supported for auto-installation of the packages
- For security and privacy reasons, R packages that provide client-server queries over the web, such as *RgoogleMaps*, in the service, aren't supported. Networking is

blocked for such attempts. See the following section for a list of supported and unsupported R packages.

- The approval process for including a new R package has a tree of dependencies. Some dependencies required to be installed in the service can't be supported.

R packages that are supported in Power BI

The following table shows which packages are supported in the Power BI service.

Package	Version	Link
abc	2.1	https://cran.r-project.org/web/packages/abc/index.html ↗
abc.data	1.0	https://cran.r-project.org/web/packages/abc.data/index.html ↗
abind	1.4-5	https://cran.r-project.org/web/packages/abind/index.html ↗
acepack	1.4.1	https://cran.r-project.org/web/packages/acepack/index.html ↗
actuar	2.3-1	https://cran.r-project.org/web/packages/actuar/index.html ↗
ade4	1.7-10	https://cran.r-project.org/web/packages/ade4/index.html ↗
adegenet	2.1.2	https://cran.r-project.org/web/packages/adegenet/index.html ↗
AdMit	2.1.3	https://cran.r-project.org/web/packages/AdMit/index.html ↗
AER	1.2-5	https://cran.r-project.org/web/packages/AER/index.html ↗
agricolae	1.3-1	https://cran.r-project.org/web/packages/agricolae/index.html ↗
AlgDesign	1.1-7.3	https://cran.r-project.org/web/packages/AlgDesign/index.html ↗
alluvial	0.1-2	https://cran.r-project.org/web/packages/alluvial/index.html ↗
andrews	1.0	https://cran.r-project.org/web/packages/andrews/index.html ↗
anomalize	0.1.1	https://cran.r-project.org/web/packages/anomalize/index.html ↗
anytime	0.3.3	https://cran.r-project.org/web/packages/anytime/index.html ↗
aod	1.3	https://cran.r-project.org/web/packages/aod/index.html ↗
apcluster	1.4.5	https://cran.r-project.org/web/packages/apcluster/index.html ↗
ape	5.0	https://cran.r-project.org/web/packages/ape/index.html ↗

Package	Version	Link
aplpack	1.3.0	https://cran.r-project.org/web/packages/aplpack/index.html ↗
approximator	1.2-6	https://cran.r-project.org/web/packages/approximator/index.html ↗
arm	1.9-3	https://cran.r-project.org/web/packages/arm/index.html ↗
arules	1.6-0	https://cran.r-project.org/web/packages/arules/index.html ↗
arulesViz	1.3-0	https://cran.r-project.org/web/packages/arulesViz/index.html ↗
ash	1.0-15	https://cran.r-project.org/web/packages/ash/index.html ↗
assertthat	0.2.0	https://cran.r-project.org/web/packages/assertthat/index.html ↗
autocogs	0.1.2	https://cran.r-project.org/web/packages/autocogs/index.html ↗
automap	1.0-14	https://cran.r-project.org/web/packages/automap/index.html ↗
aweek	1.0.1	https://cran.r-project.org/web/packages/aweek/index.html ↗
AzureML	0.2.14	https://cran.r-project.org/web/packages/AzureML/index.html ↗
BaBooN	0.2-0	https://cran.r-project.org/web/packages/BaBooN/index.html ↗
BACCO	2.0-9	https://cran.r-project.org/web/packages/BACCO/index.html ↗
backports	1.1.2	https://cran.r-project.org/web/packages/backports/index.html ↗
BaM	1.0.1	https://cran.r-project.org/web/packages/BaM/index.html ↗
BAS	1.4.9	https://cran.r-project.org/web/packages/BAS/index.html ↗
base	3.4.4	NA
base2grob	0.0.2	https://cran.r-project.org/web/packages/base2grob/index.html ↗
base64	2.0	https://cran.r-project.org/web/packages/base64/index.html ↗
base64enc	0.1-3	https://cran.r-project.org/web/packages/base64enc/index.html ↗
BayesDA	2012.04-1	https://cran.r-project.org/web/packages/BayesDA/index.html ↗
BayesFactor	0.9.12-2	https://cran.r-project.org/web/packages/BayesFactor/index.html ↗

Package	Version	Link
bayesGARCH	2.1.3	https://cran.r-project.org/web/packages/bayesGARCH/index.html ↗
bayesm	3.1-0.1	https://cran.r-project.org/web/packages/bayesm/index.html ↗
bayesmix	0.7-4	https://cran.r-project.org/web/packages/bayesmix/index.html ↗
bayesplot	1.5.0	https://cran.r-project.org/web/packages/bayesplot/index.html ↗
bayesQR	2.3	https://cran.r-project.org/web/packages/bayesQR/index.html ↗
bayesSurv	3.2	https://cran.r-project.org/web/packages/bayesSurv/index.html ↗
Bayesthresh	2.0.1	https://cran.r-project.org/web/packages/Bayesthresh/index.html ↗
BayesTree	0.3-1.4	https://cran.r-project.org/web/packages/BayesTree/index.html ↗
BayesValidate	0.0	https://cran.r-project.org/web/packages/BayesValidate/index.html ↗
BayesX	0.2-9	https://cran.r-project.org/web/packages/BayesX/index.html ↗
BayHaz	0.1-3	https://cran.r-project.org/web/packages/BayHaz/index.html ↗
bbemkr	2.0	https://cran.r-project.org/web/packages/bbemkr/index.html ↗
BCBCSF	1.0-1	https://cran.r-project.org/web/packages/BCBCSF/index.html ↗
BCE	2.1	https://cran.r-project.org/web/packages/BCE/index.html ↗
bclust	1.5	https://cran.r-project.org/web/packages/bclust/index.html ↗
bcp	4.0.0	https://cran.r-project.org/web/packages/bcp/index.html ↗
BDgraph	2.45	https://cran.r-project.org/web/packages/BDgraph/index.html ↗
beanplot	1.2	https://cran.r-project.org/web/packages/beanplot/index.html ↗
beeswarm	0.2.3	https://cran.r-project.org/web/packages/beeswarm/index.html ↗
benford.analysis	0.1.4.1	https://cran.r-project.org/web/packages/benford.analysis/index.html ↗
BenfordTests	1.2.0	https://cran.r-project.org/web/packages/BenfordTests/index.html ↗

Package	Version	Link
bfp	0.0-38	https://cran.r-project.org/web/packages/bfp/index.html ↗
BH	1.66.0-1	https://cran.r-project.org/web/packages/BH/index.html ↗
biglm	0.9-1	https://cran.r-project.org/web/packages/biglm/index.html ↗
bindr	0.1.1	https://cran.r-project.org/web/packages/bindr/index.html ↗
bindrcpp	0.2.2	https://cran.r-project.org/web/packages/bindrcpp/index.html ↗
binom	1.1-1	https://cran.r-project.org/web/packages/binom/index.html ↗
bisoreg	1.4	https://cran.r-project.org/web/packages/bisoreg/index.html ↗
bit	1.1-12	https://cran.r-project.org/web/packages/bit/index.html ↗
bit64	0.9-7	https://cran.r-project.org/web/packages/bit64/index.html ↗
bitops	1.0-6	https://cran.r-project.org/web/packages/bitops/index.html ↗
bizdays	1.0.6	https://cran.r-project.org/web/packages/bizdays/index.html ↗
blandr	0.5.1	https://cran.r-project.org/web/packages/blandr/index.html ↗
blme	1.0-4	https://cran.r-project.org/web/packages/blme/index.html ↗
blob	1.1.1	https://cran.r-project.org/web/packages/blob/index.html ↗
BLR	1.4	https://cran.r-project.org/web/packages/BLR/index.html ↗
BMA	3.18.8	https://cran.r-project.org/web/packages/BMA/index.html ↗
Bmix	0.6	https://cran.r-project.org/web/packages/Bmix/index.html ↗
bmp	0.3	https://cran.r-project.org/web/packages/bmp/index.html ↗
BMS	0.3.4	https://cran.r-project.org/web/packages/BMS/index.html ↗
bnlearn	4.3	https://cran.r-project.org/web/packages/bnlearn/index.html ↗
boa	1.1.8-2	https://cran.r-project.org/web/packages/boa/index.html ↗
bomrang	0.1.4	https://cran.r-project.org/web/packages/bomrang/index.html ↗
BoolNet	2.1.5	https://cran.r-project.org/web/packages/BoolNet/index.html ↗
Boom	0.7	https://cran.r-project.org/web/packages/Boom/index.html ↗
BoomSpikeSlab	0.9.0	https://cran.r-project.org/web/packages/BoomSpikeSlab/index.html ↗
boot	1.3-20	https://cran.r-project.org/web/packages/boot/index.html ↗

Package	Version	Link
bootstrap	2017.2	https://cran.r-project.org/web/packages/bootstrap/index.html ↗
Boruta	5.3.0	https://cran.r-project.org/web/packages/Boruta/index.html ↗
bqtl	1.0-32	https://cran.r-project.org/web/packages/bqtl/index.html ↗
BradleyTerry2	1.0-8	https://cran.r-project.org/web/packages/BradleyTerry2/index.html ↗
brew	1.0-6	https://cran.r-project.org/web/packages/brew/index.html ↗
brglm	0.6.1	https://cran.r-project.org/web/packages/brglm/index.html ↗
broom	0.4.4	https://cran.r-project.org/web/packages/broom/index.html ↗
bspec	1.5	https://cran.r-project.org/web/packages/bspec/index.html ↗
bspmma	0.1-1	https://cran.r-project.org/web/packages/bspmma/index.html ↗
bsts	0.7.1	https://cran.r-project.org/web/packages/bsts/index.html ↗
bupaR	0.4.4	https://cran.r-project.org/web/packages/bupaR/index.html ↗
BVS	4.12.1	https://cran.r-project.org/web/packages/BVS/index.html ↗
C50	0.1.1	https://cran.r-project.org/web/packages/C50/index.html ↗
Cairo	1.5-9	https://cran.r-project.org/web/packages/Cairo/index.html ↗
cairoDevice	2.24	https://cran.r-project.org/web/packages/cairoDevice/index.html ↗
calibrate	1.7.2	https://cran.r-project.org/web/packages/calibrate/index.html ↗
calibrator	1.2-6	https://cran.r-project.org/web/packages/calibrator/index.html ↗
callr	2.0.2	https://cran.r-project.org/web/packages/callr/index.html ↗
car	2.1-6	https://cran.r-project.org/web/packages/car/index.html ↗
carData	3.0-1	https://cran.r-project.org/web/packages/carData/index.html ↗
caret	6.0-78	https://cran.r-project.org/web/packages/caret/index.html ↗
catnet	1.15.3	https://cran.r-project.org/web/packages/catnet/index.html ↗
caTools	1.17.1	https://cran.r-project.org/web/packages/caTools/index.html ↗
cclust	0.6-21	https://cran.r-project.org/web/packages/cclust/index.html ↗

Package	Version	Link
cellranger	1.1.0	https://cran.r-project.org/web/packages/cellranger/index.html ↗
ChainLadder	0.2.5	https://cran.r-project.org/web/packages/ChainLadder/index.html ↗
changepoint	2.2.2	https://cran.r-project.org/web/packages/changepoint/index.html ↗
checkmate	1.8.5	https://cran.r-project.org/web/packages/checkmate/index.html ↗
checkpoint	0.4.3	https://cran.r-project.org/web/packages/checkpoint/index.html ↗
choroplethrMaps	1.0.1	https://cran.r-project.org/web/packages/choroplethrMaps/index.html ↗
chron	2.3-52	https://cran.r-project.org/web/packages/chron/index.html ↗
circlize	0.4.3	https://cran.r-project.org/web/packages/circlize/index.html ↗
Ckmeans.1d.dp	4.2.1	https://cran.r-project.org/web/packages/Ckmeans.1d.dp/index.html ↗
class	7.3-14	https://cran.r-project.org/web/packages/class/index.html ↗
classInt	0.3-3	https://cran.r-project.org/web/packages/classInt/index.html ↗
CLI	1.0.0	https://cran.r-project.org/web/packages/cli/index.html ↗
ClickClust	1.1.5	https://cran.r-project.org/web/packages/ClickClust/index.html ↗
clickstream	1.3.0	https://cran.r-project.org/web/packages/clickstream/index.html ↗
clue	0.3-54	https://cran.r-project.org/web/packages/clue/index.html ↗
cluster	2.0.6	https://cran.r-project.org/web/packages/cluster/index.html ↗
clv	0.3-2.1	https://cran.r-project.org/web/packages/clv/index.html ↗
cmpblk	2.2-7	https://cran.r-project.org/web/packages/cmpblk/index.html ↗
coda	0.19-1	https://cran.r-project.org/web/packages/coda/index.html ↗
codetools	0.2-15	https://cran.r-project.org/web/packages/codetools/index.html ↗
coefplot	1.2.6	https://cran.r-project.org/web/packages/coefplot/index.html ↗

Package	Version	Link
coin	1.2-2	https://cran.r-project.org/web/packages/coin/index.html ↗
collapsibleTree	0.1.6	https://cran.r-project.org/web/packages/collapsibleTree/index.html ↗
colorRamps	2.3	https://cran.r-project.org/web/packages/colorRamps/index.html ↗
colorspace	1.3-2	https://cran.r-project.org/web/packages/colorspace/index.html ↗
colourpicker	1.0	https://cran.r-project.org/web/packages/colourpicker/index.html ↗
combinat	0.0-8	https://cran.r-project.org/web/packages/combinat/index.html ↗
commonmark	1.4	https://cran.r-project.org/web/packages/commonmark/index.html ↗
compiler	3.4.4	NA
compositions	1.40-1	https://cran.r-project.org/web/packages/compositions/index.html ↗
CORElearn	1.52.0	https://cran.r-project.org/web/packages/CORElearn/index.html ↗
corpcor	1.6.9	https://cran.r-project.org/web/packages/corpcor/index.html ↗
corrgram	1.12	https://cran.r-project.org/web/packages/corrgram/index.html ↗
corrplot	0.84	https://cran.r-project.org/web/packages/corrplot/index.html ↗
covr	3.0.1	https://cran.r-project.org/web/packages/covr/index.html ↗
cowplot	0.9.2	https://cran.r-project.org/web/packages/cowplot/index.html ↗
cplm	0.7-5	https://cran.r-project.org/web/packages/cplm/index.html ↗
cpp11	0.4.2	https://cran.r-project.org/web/packages/cpp11/index.html ↗
crayon	1.3.4	https://cran.r-project.org/web/packages/crayon/index.html ↗
crosstalk	1.0.0	https://cran.r-project.org/web/packages/crosstalk/index.html ↗
cslogistic	0.1-3	https://cran.r-project.org/web/packages/cslogistic/index.html ↗
cts	1.0-21	https://cran.r-project.org/web/packages/cts/index.html ↗
ctv	0.8-4	https://cran.r-project.org/web/packages/ctv/index.html ↗

Package	Version	Link
cubature	1.3-11	https://cran.r-project.org/web/packages/cubature/index.html ↗
Cubist	0.2.1	https://cran.r-project.org/web/packages/Cubist/index.html ↗
curl	3.2	https://cran.r-project.org/web/packages/curl/index.html ↗
CVST	0.2-1	https://cran.r-project.org/web/packages/CVST/index.html ↗
cvTools	0.3.2	https://cran.r-project.org/web/packages/cvTools/index.html ↗
d3heatmap	0.6.1.2	https://cran.r-project.org/web/packages/d3heatmap/index.html ↗
d3Network	0.5.2.1	https://cran.r-project.org/web/packages/d3Network/index.html ↗
d3r	0.8.0	https://cran.r-project.org/web/packages/d3r/index.html ↗
data.table	1.12.6	https://cran.r-project.org/web/packages/data.table/index.html ↗
data.tree	0.7.5	https://cran.r-project.org/web/packages/data.tree/index.html ↗
datasauRus	0.1.4	https://cran.r-project.org/web/packages/datasauRus/index.html ↗
datasets	3.4.4	NA
date	1.2-38	https://cran.r-project.org/web/packages/date/index.html ↗
DBI	0.8	https://cran.r-project.org/web/packages/DBI/index.html ↗
dbplyr	1.2.1	https://cran.r-project.org/web/packages/dbplyr/index.html ↗
dbSCAN	1.1-1	https://cran.r-project.org/web/packages/dbSCAN/index.html ↗
dclone	2.2-0	https://cran.r-project.org/web/packages/dclone/index.html ↗
ddalpha	1.3.1.1	https://cran.r-project.org/web/packages/ddalpha/index.html ↗
deal	1.2-37	https://cran.r-project.org/web/packages/deal/index.html ↗
debugme	1.1.0	https://cran.r-project.org/web/packages/debugme/index.html ↗
deepnet	0.2	https://cran.r-project.org/web/packages/deepnet/index.html ↗
deldir	0.1-14	https://cran.r-project.org/web/packages/deldir/index.html ↗
dendextend	1.12.0	https://cran.r-project.org/web/packages/dendextend/index.html ↗

Package	Version	Link
DEoptimR	1.0-8	https://cran.r-project.org/web/packages/DEoptimR/index.html ↗
deployrRserve	9.0.0	NA
Deriv	3.8.4	https://cran.r-project.org/web/packages/Deriv/index.html ↗
desc	1.1.1	https://cran.r-project.org/web/packages/desc/index.html ↗
descr	1.1.4	https://cran.r-project.org/web/packages/descr/index.html ↗
deSolve	1.20	https://cran.r-project.org/web/packages/deSolve/index.html ↗
devtools	1.13.5	https://cran.r-project.org/web/packages/devtools/index.html ↗
DiagrammeR	1.0.0	https://cran.r-project.org/web/packages/DiagrammeR/index.html ↗
DiagrammeRsvg	0.1	https://cran.r-project.org/web/packages/DiagrammeRsvg/index.html ↗
dichromat	2.0-0	https://cran.r-project.org/web/packages/dichromat/index.html ↗
digest	0.6.15	https://cran.r-project.org/web/packages/digest/index.html ↗
dimRed	0.1.0	https://cran.r-project.org/web/packages/dimRed/index.html ↗
diptest	0.75-7	https://cran.r-project.org/web/packages/diptest/index.html ↗
distcrete	1.0.3	https://cran.r-project.org/web/packages/distcrete/index.html ↗
DistributionUtils	0.6-0	https://cran.r-project.org/web/packages/DistributionUtils/index.html ↗
distrom	1.0	https://cran.r-project.org/web/packages/distrom/index.html ↗
dlm	1.1-4	https://cran.r-project.org/web/packages/dlm/index.html ↗
DMwR	0.4.1	https://cran.r-project.org/web/packages/DMwR/index.html ↗
doBy	4.6-1	https://cran.r-project.org/web/packages/doBy/index.html ↗
doParallel	1.0.12	https://cran.r-project.org/web/packages/doParallel/index.html ↗
doSNOW	1.0.16	https://cran.r-project.org/web/packages/doSNOW/index.html ↗
dotCall64	0.9-5.2	https://cran.r-project.org/web/packages/dotCall64/index.html ↗

Package	Version	Link
downloader	0.4	https://cran.r-project.org/web/packages/downloader/index.html ↗
dplyr	0.8.3	https://cran.r-project.org/web/packages/dplyr/index.html ↗
DPPackage	1.1-7.4	https://cran.r-project.org/web/packages/DPPackage/index.html ↗
DRR	0.0.3	https://cran.r-project.org/web/packages/DRR/index.html ↗
dse	2015.12-1	https://cran.r-project.org/web/packages/dse/index.html ↗
DT	0.4	https://cran.r-project.org/web/packages/DT/index.html ↗
dtt	0.1-2	https://cran.r-project.org/web/packages/dtt/index.html ↗
dtw	1.18-1	https://cran.r-project.org/web/packages/dtw/index.html ↗
dygraphs	1.1.1.4	https://cran.r-project.org/web/packages/dygraphs/index.html ↗
dynlm	0.3-5	https://cran.r-project.org/web/packages/dynlm/index.html ↗
e1071	1.6-8	https://cran.r-project.org/web/packages/e1071/index.html ↗
earth	4.6.2	https://cran.r-project.org/web/packages/earth/index.html ↗
EbayesThresh	1.4-12	https://cran.r-project.org/web/packages/EbayesThresh/index.html ↗
ebdbNet	1.2.5	https://cran.r-project.org/web/packages/ebdbNet/index.html ↗
ecm	4.4.0	https://cran.r-project.org/web/packages/ecm/index.html ↗
edeaR	0.8.0	https://cran.r-project.org/web/packages/edeaR/index.html ↗
effects	4.0-1	https://cran.r-project.org/web/packages/effects/index.html ↗
ellipse	0.4.1	https://cran.r-project.org/web/packages/ellipse/index.html ↗
ellipsis	0.3.0	https://cran.r-project.org/web/packages/ellipsis/index.html ↗
emmeans	1.1.2	https://cran.r-project.org/web/packages/emmeans/index.html ↗
emulator	1.2-15	https://cran.r-project.org/web/packages/emulator/index.html ↗
energy	1.7-2	https://cran.r-project.org/web/packages/energy/index.html ↗
english	1.2-3	https://cran.r-project.org/web/packages/english/index.html ↗

Package	Version	Link
ensembleBMA	5.1.5	https://cran.r-project.org/web/packages/ensembleBMA/index.html ↗
entropy	1.2.1	https://cran.r-project.org/web/packages/entropy/index.html ↗
epitools	0.5-10.1	https://cran.r-project.org/web/packages/epitools/index.html ↗
epitrix	0.2.2	https://cran.r-project.org/web/packages/epitrix/index.html ↗
estimability	1.3	https://cran.r-project.org/web/packages/estimability/index.html ↗
eulerr	5.1.0	https://cran.r-project.org/web/packages/eulerr/index.html ↗
EvalEst	2015.4-2	https://cran.r-project.org/web/packages/EvalEst/index.html ↗
evaluate	0.10.1	https://cran.r-project.org/web/packages/evaluate/index.html ↗
evd	2.3-2	https://cran.r-project.org/web/packages/evd/index.html ↗
evdbayes	1.1-1	https://cran.r-project.org/web/packages/evdbayes/index.html ↗
eventdataR	0.2.0	https://cran.r-project.org/web/packages eventdataR/index.html ↗
exactLoglinTest	1.4.2	https://cran.r-project.org/web/packages/exactLoglinTest/index.html ↗
exactRankTests	0.8-29	https://cran.r-project.org/web/packages/exactRankTests/index.html ↗
expint	0.1-4	https://cran.r-project.org/web/packages/expint/index.html ↗
expm	0.999-2	https://cran.r-project.org/web/packages/expm/index.html ↗
extraDistr	1.8.8	https://cran.r-project.org/web/packages/extraDistr/index.html ↗
extrafont	0.17	https://cran.r-project.org/web/packages/extrafont/index.html ↗
extrafontdb	1.0	https://cran.r-project.org/web/packages/extrafontdb/index.html ↗
extremevalues	2.3.2	https://cran.r-project.org/web/packages/extremevalues/index.html ↗
ez	4.4-0	https://cran.r-project.org/web/packages/ez/index.html ↗
factoextra	1.0.5	https://cran.r-project.org/web/packages/factoextra/index.html ↗

Package	Version	Link
FactoMineR	1.40	https://cran.r-project.org/web/packages/FactoMineR/index.html ↗
factorQR	0.1-4	https://cran.r-project.org/web/packages/factorQR/index.html ↗
fansi	0.4.0	https://cran.r-project.org/web/packages/fansi/index.html ↗
faoutlier	0.7.2	https://cran.r-project.org/web/packages/faoutlier/index.html ↗
farver	1.1.0	https://cran.r-project.org/web/packages/farver/index.html ↗
fastICA	1.2-1	https://cran.r-project.org/web/packages/fastICA/index.html ↗
fastmatch	1.1-0	https://cran.r-project.org/web/packages/fastmatch/index.html ↗
fBasics	3042.89	https://cran.r-project.org/web/packages/fBasics/index.html ↗
fdrtool	1.2.15	https://cran.r-project.org/web/packages/fdrtool/index.html ↗
fGarch	3042.83.1	https://cran.r-project.org/web/packages/fGarch/index.html ↗
fields	9.6	https://cran.r-project.org/web/packages/fields/index.html ↗
filehash	2.4-1	https://cran.r-project.org/web/packages/filehash/index.html ↗
FinCal	0.6.3	https://cran.r-project.org/web/packages/FinCal/index.html ↗
fitdistrplus	1.0-9	https://cran.r-project.org/web/packages/fitdistrplus/index.html ↗
flashClust	1.01-2	https://cran.r-project.org/web/packages/flashClust/index.html ↗
flexclust	1.3-5	https://cran.r-project.org/web/packages/flexclust/index.html ↗
flexmix	2.3-14	https://cran.r-project.org/web/packages/flexmix/index.html ↗
FME	1.3.5	https://cran.r-project.org/web/packages/FME/index.html ↗
fmsb	0.6.1	https://cran.r-project.org/web/packages/fmsb/index.html ↗
FNN	1.1	https://cran.r-project.org/web/packages/FNN/index.html ↗
fontBitstreamVera	0.1.1	https://cran.r-project.org/web/packages/fontBitstreamVera/index.html ↗
fontLiberation	0.1.0	https://cran.r-project.org/web/packages/fontLiberation/index.html ↗

Package	Version	Link
fontquiver	0.2.1	https://cran.r-project.org/web/packages/fontquiver/index.html ↗
forcats	0.3.0	https://cran.r-project.org/web/packages/forcats/index.html ↗
foreach	1.4.4	https://cran.r-project.org/web/packages/foreach/index.html ↗
forecast	8.7	https://cran.r-project.org/web/packages/forecast/index.html ↗
forecastHybrid	2.1.11	https://cran.r-project.org/web/packages/forecastHybrid/index.html ↗
foreign	0.8-69	https://cran.r-project.org/web/packages/foreign/index.html ↗
formatR	1.5	https://cran.r-project.org/web/packages/formatR/index.html ↗
formattable	0.2.0.1	https://cran.r-project.org/web/packages/formattable/index.html ↗
Formula	1.2-2	https://cran.r-project.org/web/packages/Formula/index.html ↗
fpc	2.1-11	https://cran.r-project.org/web/packages/fpc/index.html ↗
fracdiff	1.4-2	https://cran.r-project.org/web/packages/fracdiff/index.html ↗
fTrading	3042.79	https://cran.r-project.org/web/packages/fTrading/index.html ↗
fUnitRoots	3042.79	https://cran.r-project.org/web/packages/fUnitRoots/index.html ↗
futile.logger	1.4.3	https://cran.r-project.org/web/packages/futile.logger/index.html ↗
futile.options	1.0.0	https://cran.r-project.org/web/packages/futile.options/index.html ↗
future	1.15.0	https://cran.r-project.org/web/packages/future/index.html ↗
future.apply	1.3.0	https://cran.r-project.org/web/packages/future.apply/index.html ↗
gam	1.15	https://cran.r-project.org/web/packages/gam/index.html ↗
gamlr	1.13-4	https://cran.r-project.org/web/packages/gamlr/index.html ↗
gamlss	5.0-6	https://cran.r-project.org/web/packages/gamlss/index.html ↗
gamlss.data	5.0-0	https://cran.r-project.org/web/packages/gamlss.data/index.html ↗

Package	Version	Link
gamlss.dist	5.0-4	https://cran.r-project.org/web/packages/gamlss.dist/index.html ↗
gbm	2.1.3	https://cran.r-project.org/web/packages/gbm/index.html ↗
gclus	1.3.1	https://cran.r-project.org/web/packages/gclus/index.html ↗
gdalUtils	2.0.1.7	https://cran.r-project.org/web/packages/gdalUtils/index.html ↗
gdata	2.18.0	https://cran.r-project.org/web/packages/gdata/index.html ↗
gdtools	0.1.7	https://cran.r-project.org/web/packages/gdtools/index.html ↗
gee	4.13-19	https://cran.r-project.org/web/packages/gee/index.html ↗
genalg	0.2.0	https://cran.r-project.org/web/packages/genalg/index.html ↗
generics	0.1.2	https://cran.r-project.org/web/packages/generics/index.html ↗
genetics	1.3.8.1	https://cran.r-project.org/web/packages/genetics/index.html ↗
GenSA	1.1.7	https://cran.r-project.org/web/packages/GenSA/index.html ↗
geojson	0.2.0	https://cran.r-project.org/web/packages/geojson/index.html ↗
geojsonio	0.6.0	https://cran.r-project.org/web/packages/geojsonio/index.html ↗
geojsonlint	0.2.0	https://cran.r-project.org/web/packages/geojsonlint/index.html ↗
geoR	1.7-5.2	https://cran.r-project.org/web/packages/geoR/index.html ↗
geoRglm	0.9-11	https://cran.r-project.org/web/packages/geoRglm/index.html ↗
geosphere	1.5-7	https://cran.r-project.org/web/packages/geosphere/index.html ↗
GGally	2.0.0	https://cran.r-project.org/web/packages/GGally/index.html ↗
ggalt	0.4.0	https://cran.r-project.org/web/packages/ggalt/index.html ↗
ggridge	0.1.3	https://cran.r-project.org/web/packages/ggridge/index.html ↗
ggridges	0.5.0	https://cran.r-project.org/web/packages/ggridges/index.html ↗
ggnetwork	2.0.1	https://cran.r-project.org/web/packages/ggnetwork/index.html ↗
ggnetwork2	0.1.0	https://cran.r-project.org/web/packages/ggnetwork2/index.html ↗
gganimate	1.0.3	https://cran.r-project.org/web/packages/gganimate/index.html ↗
ggcorrplot	0.1.1	https://cran.r-project.org/web/packages/ggcorrplot/index.html ↗
ggdendro	0.1-20	https://cran.r-project.org/web/packages/ggdendro/index.html ↗

Package	Version	Link
ggeffects	0.3.2	https://cran.r-project.org/web/packages/ggeffects/index.html ↗
ggExtra	0.9	https://cran.r-project.org/web/packages/ggExtra/index.html ↗
ggforce	0.1.1	https://cran.r-project.org/web/packages/ggforce/index.html ↗
ggformula	0.6.2	https://cran.r-project.org/web/packages/ggformula/index.html ↗
ggfortify	0.4.3	https://cran.r-project.org/web/packages/ggfortify/index.html ↗
gghighlight	0.3.0	https://cran.r-project.org/web/packages/gghighlight/index.html ↗
ggimage	0.1.2	https://cran.r-project.org/web/packages/ggimage/index.html ↗
ggiraph	0.6.1	https://cran.r-project.org/web/packages/ggiraph/index.html ↗
ggjoy	0.4.0	https://cran.r-project.org/web/packages/ggjoy/index.html ↗
ggm	2.3	https://cran.r-project.org/web/packages/ggm/index.html ↗
ggmap	3.0.0	https://cran.r-project.org/web/packages/ggmap/index.html ↗
ggmcmc	1.1	https://cran.r-project.org/web/packages/ggmcmc/index.html ↗
ggplot2	3.3.3	https://cran.r-project.org/web/packages/ggplot2/index.html ↗
ggplot2movies	0.0.1	https://cran.r-project.org/web/packages/ggplot2movies/index.html ↗
ggpmisc	0.2.16	https://cran.r-project.org/web/packages/ggpmisc/index.html ↗
ggpubr	0.2.3	https://cran.r-project.org/web/packages/ggpubr/index.html ↗
ggQC	0.0.31	https://cran.r-project.org/web/packages/ggQC/index.html ↗
ggRandomForests	2.0.1	https://cran.r-project.org/web/packages/ggRandomForests/index.html ↗
ggraph	1.0.1	https://cran.r-project.org/web/packages/ggraph/index.html ↗
ggrepel	0.8.0	https://cran.r-project.org/web/packages/ggrepel/index.html ↗
ggridges	0.4.1	https://cran.r-project.org/web/packages/ggridges/index.html ↗
ggsci	2.8	https://cran.r-project.org/web/packages/ggsci/index.html ↗
ggsignif	0.4.0	https://cran.r-project.org/web/packages/ggsignif/index.html ↗
ggsoccer	0.1.4	https://cran.r-project.org/web/packages/ggsoccer/index.html ↗

Package	Version	Link
ggstance	0.3	https://cran.r-project.org/web/packages/ggstance/index.html ↗
ggtern	2.2.1	https://cran.r-project.org/web/packages/ggtern/index.html ↗
ggthemes	3.4.0	https://cran.r-project.org/web/packages/ggthemes/index.html ↗
gistr	0.4.0	https://cran.r-project.org/web/packages/gistr/index.html ↗
git2r	0.21.0	https://cran.r-project.org/web/packages/git2r/index.html ↗
glasso	1.8	https://cran.r-project.org/web/packages/glasso/index.html ↗
glmmBUGS	2.4.0	https://cran.r-project.org/web/packages/glmmBUGS/index.html ↗
glmmTMB	0.2.0	https://cran.r-project.org/web/packages/glmmTMB/index.html ↗
glmnet	2.0-13	https://cran.r-project.org/web/packages/glmnet/index.html ↗
GlobalOptions	0.0.13	https://cran.r-project.org/web/packages/GlobalOptions/index.html ↗
globals	0.12.4	https://cran.r-project.org/web/packages/globals/index.html ↗
glue	1.3.1	https://cran.r-project.org/web/packages/glue/index.html ↗
gmodels	2.16.2	https://cran.r-project.org/web/packages/gmodels/index.html ↗
gmp	0.5-13.1	https://cran.r-project.org/web/packages/gmp/index.html ↗
gnm	1.0-8	https://cran.r-project.org/web/packages/gnm/index.html ↗
goftest	1.1-1	https://cran.r-project.org/web/packages/goftest/index.html ↗
googleVis	0.6.2	https://cran.r-project.org/web/packages/googleVis/index.html ↗
gower	0.1.2	https://cran.r-project.org/web/packages/gower/index.html ↗
GPArotation	2014.11-1	https://cran.r-project.org/web/packages/GPArotation/index.html ↗
gplots	3.0.1	https://cran.r-project.org/web/packages/gplots/index.html ↗
graphics	3.4.4	NA
grDevices	3.4.4	NA
grid	3.4.4	NA

Package	Version	Link
gridBase	0.4-7	https://cran.r-project.org/web/packages/gridBase/index.html ↗
gridExtra	2.3	https://cran.r-project.org/web/packages/gridExtra/index.html ↗
gridGraphics	0.2-1	https://cran.r-project.org/web/packages/gridGraphics/index.html ↗
growcurves	0.2.4.1	https://cran.r-project.org/web/packages/growcurves/index.html ↗
grpreg	3.1-2	https://cran.r-project.org/web/packages/grpreg/index.html ↗
gss	2.1-7	https://cran.r-project.org/web/packages/gss/index.html ↗
gstat	1.1-5	https://cran.r-project.org/web/packages/gstat/index.html ↗
gsubfn	0.7	https://cran.r-project.org/web/packages/gsubfn/index.html ↗
gtable	0.2.0	https://cran.r-project.org/web/packages/gtable/index.html ↗
gtools	3.5.0	https://cran.r-project.org/web/packages/gtools/index.html ↗
gtrendsR	1.4.3	https://cran.r-project.org/web/packages/gtrendsR/index.html ↗
gWidgets	0.0-54	https://cran.r-project.org/web/packages/gWidgets/index.html ↗
gWidgetsRGtk2	0.0-86	https://cran.r-project.org/web/packages/gWidgetsRGtk2/index.html ↗
gWidgetscltk	0.0-55	https://cran.r-project.org/web/packages/gWidgetscltk/index.html ↗
haplo.stats	1.7.7	https://cran.r-project.org/web/packages/haplo.stats/index.html ↗
hash	2.2.6	https://cran.r-project.org/web/packages/hash/index.html ↗
haven	1.1.1	https://cran.r-project.org/web/packages/haven/index.html ↗
hbsae	1.0	https://cran.r-project.org/web/packages/hbsae/index.html ↗
HDInterval	0.2.0	https://cran.r-project.org/web/packages/HDInterval/index.html ↗
hdrcde	3.2	https://cran.r-project.org/web/packages/hdrcde/index.html ↗
heatmaply	0.16.0	https://cran.r-project.org/web/packages/heatmaply/index.html ↗
heavy	0.38.19	https://cran.r-project.org/web/packages/heavy/index.html ↗

Package	Version	Link
hexbin	1.27.2	https://cran.r-project.org/web/packages/hexbin/index.html ↗
hflights	0.1	https://cran.r-project.org/web/packages/hflights/index.html ↗
HH	3.1-34	https://cran.r-project.org/web/packages/HH/index.html ↗
HI	0.4	https://cran.r-project.org/web/packages/HI/index.html ↗
highcharter	0.5.0	https://cran.r-project.org/web/packages/highcharter/index.html ↗
highr	0.6	https://cran.r-project.org/web/packages/highr/index.html ↗
HistData	0.8-2	https://cran.r-project.org/web/packages/HistData/index.html ↗
Hmisc	4.1-1	https://cran.r-project.org/web/packages/Hmisc/index.html ↗
hms	0.4.2	https://cran.r-project.org/web/packages/hms/index.html ↗
hoardr	0.2.0	https://cran.r-project.org/web/packages/hoardr/index.html ↗
hrbrthemes	0.6.0	https://cran.r-project.org/web/packages/hrbrthemes/index.html ↗
HSAUR	1.3-9	https://cran.r-project.org/web/packages/HSAUR/index.html ↗
htmlTable	1.11.2	https://cran.r-project.org/web/packages/htmlTable/index.html ↗
htmltools	0.3.6	https://cran.r-project.org/web/packages/htmltools/index.html ↗
htmlwidgets	1.3	https://cran.r-project.org/web/packages/htmlwidgets/index.html ↗
hts	5.1.5	https://cran.r-project.org/web/packages/hts/index.html ↗
httpuv	1.3.6.2	https://cran.r-project.org/web/packages/httpuv/index.html ↗
httr	1.3.1	https://cran.r-project.org/web/packages/httr/index.html ↗
huge	1.2.7	https://cran.r-project.org/web/packages/huge/index.html ↗
hunspell	2.9	https://cran.r-project.org/web/packages/hunspell/index.html ↗
hydroTSM	0.5-1	https://cran.r-project.org/web/packages/hydroTSM/index.html ↗
IBrokers	0.9-12	https://cran.r-project.org/web/packages/IBrokers/index.html ↗
ifultools	2.0-4	https://cran.r-project.org/web/packages/ifultools/index.html ↗

Package	Version	Link
igraph	1.2.1	https://cran.r-project.org/web/packages/igraph/index.html ↗
imager	0.40.2	https://cran.r-project.org/web/packages/imager/index.html ↗
imputeTS	2.7	https://cran.r-project.org/web/packages/imputeTS/index.html ↗
incidence	1.7.2	https://cran.r-project.org/web/packages/incidence/index.html ↗
influenceR	0.1.0	https://cran.r-project.org/web/packages/influenceR/index.html ↗
InformationValue	1.2.3	https://cran.r-project.org/web/packages/InformationValue/index.html ↗
inline	0.3.14	https://cran.r-project.org/web/packages/inline/index.html ↗
intervals	0.15.1	https://cran.r-project.org/web/packages/intervals/index.html ↗
inum	1.0-0	https://cran.r-project.org/web/packages/inum/index.html ↗
investr	1.4.2	https://cran.r-project.org/web/packages/investr/index.html ↗
ipred	0.9-6	https://cran.r-project.org/web/packages/ipred/index.html ↗
irlba	2.3.2	https://cran.r-project.org/web/packages/irlba/index.html ↗
irr	0.84	https://cran.r-project.org/web/packages/irr/index.html ↗
isoband	0.2.0	https://cran.r-project.org/web/packages/isoband/index.html ↗
ISOcodes	2017.09.27	https://cran.r-project.org/web/packages/ISOcodes/index.html ↗
iterators	1.0.9	https://cran.r-project.org/web/packages/iterators/index.html ↗
janeaustenr	0.1.5	https://cran.r-project.org/web/packages/janeaustenr/index.html ↗
janitor	1.0.0	https://cran.r-project.org/web/packages/janitor/index.html ↗
jmvcore	1.0.8	https://cran.r-project.org/web/packages/jmvcore/index.html ↗
jpeg	0.1-8	https://cran.r-project.org/web/packages/jpeg/index.html ↗
jqr	1.0.0	https://cran.r-project.org/web/packages/jqr/index.html ↗
jsonlite	1.6	https://cran.r-project.org/web/packages/jsonlite/index.html ↗

Package	Version	Link
jsonvalidate	1.0.0	https://cran.r-project.org/web/packages/jsonvalidate/index.html ↗
jtools	0.9.4	https://cran.r-project.org/web/packages/jtools/index.html ↗
kableExtra	0.7.0	https://cran.r-project.org/web/packages/kableExtra/index.html ↗
Kendall	2.2	https://cran.r-project.org/web/packages/Kendall/index.html ↗
kernlab	0.9-25	https://cran.r-project.org/web/packages/kernlab/index.html ↗
KernSmooth	2.23-15	https://cran.r-project.org/web/packages/KernSmooth/index.html ↗
KFKSDS	1.6	https://cran.r-project.org/web/packages/KFKSDS/index.html ↗
kinship2	1.6.4	https://cran.r-project.org/web/packages/kinship2/index.html ↗
kknn	1.3.1	https://cran.r-project.org/web/packages/kknn/index.html ↗
klaR	0.6-14	https://cran.r-project.org/web/packages/klaR/index.html ↗
km.ci	0.5-2	https://cran.r-project.org/web/packages/km.ci/index.html ↗
KMsurv	0.1-5	https://cran.r-project.org/web/packages/KMsurv/index.html ↗
knitr	1.20	https://cran.r-project.org/web/packages/knitr/index.html ↗
ks	1.11.0	https://cran.r-project.org/web/packages/ks/index.html ↗
labeling	0.3	https://cran.r-project.org/web/packages/labeling/index.html ↗
labelled	1.0.1	https://cran.r-project.org/web/packages/labelled/index.html ↗
laeken	0.4.6	https://cran.r-project.org/web/packages/laeken/index.html ↗
Lahman	6.0-0	https://cran.r-project.org/web/packages/Lahman/index.html ↗
lambda.r	1.2	https://cran.r-project.org/web/packages/lambda.r/index.html ↗
lars	1.2	https://cran.r-project.org/web/packages/lars/index.html ↗
later	1.0.0	https://cran.r-project.org/web/packages/later/index.html ↗
latex2exp	0.4.0	https://cran.r-project.org/web/packages/latex2exp/index.html ↗
lattice	0.20-35	https://cran.r-project.org/web/packages/lattice/index.html ↗

Package	Version	Link
latticeExtra	0.6-28	https://cran.r-project.org/web/packages/latticeExtra/index.html ↗
lava	1.6.1	https://cran.r-project.org/web/packages/lava/index.html ↗
lavaan	0.5-23.1097	https://cran.r-project.org/web/packages/lavaan/index.html ↗
lazyeval	0.2.1	https://cran.r-project.org/web/packages/lazyeval/index.html ↗
lda	1.4.2	https://cran.r-project.org/web/packages/lda/index.html ↗
leaflet	2.0.2	https://cran.r-project.org/web/packages/leaflet/index.html ↗
leaflet.esri	0.2	https://cran.r-project.org/web/packages/leaflet.esri/index.html ↗
leaflet.extras	0.2	https://cran.r-project.org/web/packages/leaflet.extras/index.html ↗
leaps	3.0	https://cran.r-project.org/web/packages/leaps/index.html ↗
LearnBayes	2.15.1	https://cran.r-project.org/web/packages/LearnBayes/index.html ↗
lexicon	1.2.1	https://cran.r-project.org/web/packages/lexicon/index.html ↗
libcoin	1.0-1	https://cran.r-project.org/web/packages/libcoin/index.html ↗
LiblineaR	2.10-8	https://cran.r-project.org/web/packages/LiblineaR/index.html ↗
LICORS	0.2.0	https://cran.r-project.org/web/packages/LICORS/index.html ↗
lifecycle	0.1.0	https://cran.r-project.org/web/packages/lifecycle/index.html ↗
likert	1.3.5	https://cran.r-project.org/web/packages/likert/index.html ↗
limSolve	1.5.5.3	https://cran.r-project.org/web/packages/limSolve/index.html ↗
linelist	0.0.40.9000	https://cran.r-project.org/web/packages/linelist/index.html ↗
linprog	0.9-2	https://cran.r-project.org/web/packages/linprog/index.html ↗
listenv	0.7.0	https://cran.r-project.org/web/packages/listenv/index.html ↗
lm.beta	1.5-1	https://cran.r-project.org/web/packages/lm.beta/index.html ↗
lme4	1.1-16	https://cran.r-project.org/web/packages/lme4/index.html ↗
lmm	1.0	https://cran.r-project.org/web/packages/lmm/index.html ↗

Package	Version	Link
lmtest	0.9-35	https://cran.r-project.org/web/packages/lmtest/index.html ↗
locfit	1.5-9.1	https://cran.r-project.org/web/packages/locfit/index.html ↗
locpol	0.6-0	https://cran.r-project.org/web/packages/locpol/index.html ↗
LogicReg	1.5.9	https://cran.r-project.org/web/packages/LogicReg/index.html ↗
lpSolve	5.6.13	https://cran.r-project.org/web/packages/lpSolve/index.html ↗
lsa	0.73.1	https://cran.r-project.org/web/packages/lsa/index.html ↗
lsmeans	2.27-61	https://cran.r-project.org/web/packages/lsmeans/index.html ↗
lubridate	1.7.2	https://cran.r-project.org/web/packages/lubridate/index.html ↗
magic	1.5-8	https://cran.r-project.org/web/packages/magic/index.html ↗
magick	1.8	https://cran.r-project.org/web/packages/magick/index.html ↗
magrittr	1.5	https://cran.r-project.org/web/packages/magrittr/index.html ↗
manipulateWidget	0.9.0	https://cran.r-project.org/web/packages/manipulateWidget/index.html ↗
MAPA	2.0.4	https://cran.r-project.org/web/packages/MAPA/index.html ↗
mapdata	2.3.0	https://cran.r-project.org/web/packages/mapdata/index.html ↗
mapproj	1.2.6	https://cran.r-project.org/web/packages/mapproj/index.html ↗
maps	3.2.0	https://cran.r-project.org/web/packages/maps/index.html ↗
maptools	0.9-2	https://cran.r-project.org/web/packages/maptools/index.html ↗
maptree	1.4-7	https://cran.r-project.org/web/packages/maptree/index.html ↗
mapview	2.3.0	https://cran.r-project.org/web/packages/mapview/index.html ↗
marima	2.2	https://cran.r-project.org/web/packages/marima/index.html ↗
markdown	0.8	https://cran.r-project.org/web/packages/markdown/index.html ↗
MASS	7.3-49	https://cran.r-project.org/web/packages/MASS/index.html ↗
MasterBayes	2.55	https://cran.r-project.org/web/packages/MasterBayes/index.html ↗

Package	Version	Link
Matching	4.9-5	https://cran.r-project.org/web/packages/Matching/index.html ↗
MatchIt	3.0.2	https://cran.r-project.org/web/packages/MatchIt/index.html ↗
matchmaker	0.1.1	https://cran.r-project.org/web/packages/matchmaker/index.html ↗
Matrix	1.2-12	https://cran.r-project.org/web/packages/Matrix/index.html ↗
matrixcalc	1.0-3	https://cran.r-project.org/web/packages/matrixcalc/index.html ↗
MatrixModels	0.4-1	https://cran.r-project.org/web/packages/MatrixModels/index.html ↗
matrixStats	0.54.0	https://cran.r-project.org/web/packages/matrixStats/index.html ↗
maxent	1.3.3.1	https://cran.r-project.org/web/packages/maxent/index.html ↗
maxLik	1.3-4	https://cran.r-project.org/web/packages/maxLik/index.html ↗
maxstat	0.7-25	https://cran.r-project.org/web/packages/maxstat/index.html ↗
mboost	2.8-1	https://cran.r-project.org/web/packages/mboost/index.html ↗
mclust	5.4	https://cran.r-project.org/web/packages/mclust/index.html ↗
mcmc	0.9-5	https://cran.r-project.org/web/packages/mcmc/index.html ↗
MCMCglmm	2.25	https://cran.r-project.org/web/packages/MCMCglmm/index.html ↗
mda	0.4-10	https://cran.r-project.org/web/packages/mda/index.html ↗
memoise	1.1.0	https://cran.r-project.org/web/packages/memoise/index.html ↗
merTools	0.3.0	https://cran.r-project.org/web/packages/merTools/index.html ↗
meta	4.9-1	https://cran.r-project.org/web/packages/meta/index.html ↗
metafor	2.0-0	https://cran.r-project.org/web/packages/metafor/index.html ↗
methods	3.4.4	NA
metricsgraphics	0.9.0	https://cran.r-project.org/web/packages/metricsgraphics/index.html ↗
mgcv	1.8-23	https://cran.r-project.org/web/packages/mgcv/index.html ↗

Package	Version	Link
mgsub	1.7.1	https://cran.r-project.org/web/packages/mgsub/index.html ↗
mi	1.0	https://cran.r-project.org/web/packages/mi/index.html ↗
mice	2.46.0	https://cran.r-project.org/web/packages/mice/index.html ↗
microbenchmark	1.4-4	https://cran.r-project.org/web/packages/microbenchmark/index.html ↗
MicrosoftR	3.4.4.0105	NA
mime	0.5	https://cran.r-project.org/web/packages/mime/index.html ↗
miniCRAN	0.2.11	https://cran.r-project.org/web/packages/miniCRAN/index.html ↗
miniUI	0.1.1	https://cran.r-project.org/web/packages/miniUI/index.html ↗
minpack.lm	1.2-1	https://cran.r-project.org/web/packages/minpack.lm/index.html ↗
minqa	1.2.4	https://cran.r-project.org/web/packages/minqa/index.html ↗
mirt	1.27.1	https://cran.r-project.org/web/packages/mirt/index.html ↗
misc3d	0.8-4	https://cran.r-project.org/web/packages/misc3d/index.html ↗
miscTools	0.6-22	https://cran.r-project.org/web/packages/miscTools/index.html ↗
mitools	2.3	https://cran.r-project.org/web/packages/mitools/index.html ↗
mixtools	1.1.0	https://cran.r-project.org/web/packages/mixtools/index.html ↗
mlapi	0.1.0	https://cran.r-project.org/web/packages/mlapi/index.html ↗
mlbench	2.1-1	https://cran.r-project.org/web/packages/mlbench/index.html ↗
mlogitBMA	0.1-6	https://cran.r-project.org/web/packages/mlogitBMA/index.html ↗
mnormt	1.5-5	https://cran.r-project.org/web/packages/mnormt/index.html ↗
MNP	3.1-0	https://cran.r-project.org/web/packages/MNP/index.html ↗
ModelMetrics	1.1.0	https://cran.r-project.org/web/packages/ModelMetrics/index.html ↗
modelr	0.1.1	https://cran.r-project.org/web/packages/modelr/index.html ↗

Package	Version	Link
modeltools	0.2-21	https://cran.r-project.org/web/packages/modeltools/index.html ↗
mombf	1.9.6	https://cran.r-project.org/web/packages/mombf/index.html ↗
moments	0.14	https://cran.r-project.org/web/packages/moments/index.html ↗
monomvn	1.9-7	https://cran.r-project.org/web/packages/monomvn/index.html ↗
monreg	0.1.3	https://cran.r-project.org/web/packages/monreg/index.html ↗
mosaic	1.1.1	https://cran.r-project.org/web/packages/mosaic/index.html ↗
mosaicCore	0.4.2	https://cran.r-project.org/web/packages/mosaicCore/index.html ↗
mosaicData	0.16.0	https://cran.r-project.org/web/packages/mosaicData/index.html ↗
MSBVAR	0.9-3	https://cran.r-project.org/web/packages/MSBVAR/index.html ↗
msir	1.3.2	https://cran.r-project.org/web/packages/msir/index.html ↗
msm	1.6.6	https://cran.r-project.org/web/packages/msm/index.html ↗
multcomp	1.4-8	https://cran.r-project.org/web/packages/multcomp/index.html ↗
multicool	0.1-10	https://cran.r-project.org/web/packages/multicool/index.html ↗
munsell	0.5.0	https://cran.r-project.org/web/packages/munsell/index.html ↗
mvoutlier	2.0.9	https://cran.r-project.org/web/packages/mvoutlier/index.html ↗
mvtnorm	1.0-7	https://cran.r-project.org/web/packages/mvtnorm/index.html ↗
NbClust	3.0	https://cran.r-project.org/web/packages/NbClust/index.html ↗
ncvreg	3.9-1	https://cran.r-project.org/web/packages/ncvreg/index.html ↗
network	1.13.0	https://cran.r-project.org/web/packages/network/index.html ↗
networkD3	0.4	https://cran.r-project.org/web/packages/networkD3/index.html ↗
neuralnet	1.33	https://cran.r-project.org/web/packages/neuralnet/index.html ↗

Package	Version	Link
ngram	3.0.4	https://cran.r-project.org/web/packages/ngram/index.html ↗
nlme	3.1-131.1	https://cran.r-project.org/web/packages/nlme/index.html ↗
nloptr	1.0.4	https://cran.r-project.org/web/packages/nloptr/index.html ↗
NLP	0.1-11	https://cran.r-project.org/web/packages/NLP/index.html ↗
nls.multstart	1.2.0	https://cran.r-project.org/web/packages/nls.multstart/index.html ↗
NMF	0.21.0	https://cran.r-project.org/web/packages/NMF/index.html ↗
nnet	7.3-12	https://cran.r-project.org/web/packages/nnet/index.html ↗
nnls	1.4	https://cran.r-project.org/web/packages/nnls/index.html ↗
nortest	1.0-4	https://cran.r-project.org/web/packages/nortest/index.html ↗
numbers	0.6-6	https://cran.r-project.org/web/packages/numbers/index.html ↗
numDeriv	2016.8-1	https://cran.r-project.org/web/packages/numDeriv/index.html ↗
numform	0.4.0	https://cran.r-project.org/web/packages/numform/index.html ↗
OceanView	1.0.4	https://cran.r-project.org/web/packages/OceanView/index.html ↗
openair	2.3-0	https://cran.r-project.org/web/packages/openair/index.html ↗
openssl	1.0.1	https://cran.r-project.org/web/packages/openssl/index.html ↗
osmar	1.1-7	https://cran.r-project.org/web/packages/osmar/index.html ↗
outbreaks	1.5.0	https://cran.r-project.org/web/packages/outbreaks/index.html ↗
OutlierDC	0.3-0	https://cran.r-project.org/web/packages/OutlierDC/index.html ↗
OutlierDM	1.1.1	https://cran.r-project.org/web/packages/OutlierDM/index.html ↗
outliers	0.14	https://cran.r-project.org/web/packages/outliers/index.html ↗
pacbpred	0.92.2	https://cran.r-project.org/web/packages/pacbpred/index.html ↗

Package	Version	Link
packcircles	0.3.3	https://cran.r-project.org/web/packages/packcircles/index.html ↗
padr	0.4.0	https://cran.r-project.org/web/packages/padr/index.html ↗
parallel	3.4.4	NA
partitions	1.9-19	https://cran.r-project.org/web/packages/partitions/index.html ↗
party	1.2-4	https://cran.r-project.org/web/packages/party/index.html ↗
partykit	1.2-0	https://cran.r-project.org/web/packages/partykit/index.html ↗
PAWL	0.5	https://cran.r-project.org/web/packages/PAWL/index.html ↗
pbapply	1.3-4	https://cran.r-project.org/web/packages/pbapply/index.html ↗
pbivnorm	0.6.0	https://cran.r-project.org/web/packages/pbivnorm/index.html ↗
pbkrtest	0.4-7	https://cran.r-project.org/web/packages/pbkrtest/index.html ↗
PCAmixdata	3.1	https://cran.r-project.org/web/packages/PCAmixdata/index.html ↗
pcaPP	1.9-73	https://cran.r-project.org/web/packages/pcaPP/index.html ↗
pdc	1.0.3	https://cran.r-project.org/web/packages/pdc/index.html ↗
pegas	0.12	https://cran.r-project.org/web/packages/pegas/index.html ↗
PerformanceAnalytics	1.5.2	https://cran.r-project.org/web/packages/PerformanceAnalytics/index.html ↗
permute	0.9-4	https://cran.r-project.org/web/packages/permute/index.html ↗
perry	0.2.0	https://cran.r-project.org/web/packages/perry/index.html ↗
petrinetR	0.1.0	https://cran.r-project.org/web/packages/petrinetR/index.html ↗
pheatmap	1.0.8	https://cran.r-project.org/web/packages/pheatmap/index.html ↗
pillar	1.4.2	https://cran.r-project.org/web/packages/pillar/index.html ↗
pixmap	0.4-11	https://cran.r-project.org/web/packages/pixmap/index.html ↗
pkgconfig	2.0.2	https://cran.r-project.org/web/packages/pkgconfig/index.html ↗

Package	Version	Link
pkgmaker	0.22	https://cran.r-project.org/web/packages/pkgmaker/index.html ↗
platetools	0.1.0	https://cran.r-project.org/web/packages/platetools/index.html ↗
plogr	0.2.0	https://cran.r-project.org/web/packages/plogr/index.html ↗
plot3D	1.1.1	https://cran.r-project.org/web/packages/plot3D/index.html ↗
plot3Drgl	1.0.1	https://cran.r-project.org/web/packages/plot3Drgl/index.html ↗
plotly	4.9.2.2	https://cran.r-project.org/web/packages/plotly/index.html ↗
plotmo	3.3.6	https://cran.r-project.org/web/packages/plotmo/index.html ↗
plotrix	3.7	https://cran.r-project.org/web/packages/plotrix/index.html ↗
pls	2.6-0	https://cran.r-project.org/web/packages/pls/index.html ↗
plyr	1.8.4	https://cran.r-project.org/web/packages/plyr/index.html ↗
png	0.1-7	https://cran.r-project.org/web/packages/png/index.html ↗
polyspline	1.1.12	https://cran.r-project.org/web/packages/polyspline/index.html ↗
polyclip	1.6-1	https://cran.r-project.org/web/packages/polyclip/index.html ↗
polylabelr	0.1.0	https://cran.r-project.org/web/packages/polylabelr/index.html ↗
polynom	1.3-9	https://cran.r-project.org/web/packages/polynom/index.html ↗
ppcor	1.1	https://cran.r-project.org/web/packages/ppcor/index.html ↗
prabclus	2.2-6	https://cran.r-project.org/web/packages/prabclus/index.html ↗
pracma	2.1.4	https://cran.r-project.org/web/packages/pracma/index.html ↗
praise	1.0.0	https://cran.r-project.org/web/packages/praise/index.html ↗
precrec	0.10.1	https://cran.r-project.org/web/packages/precrec/index.html ↗
prediction	0.2.0	https://cran.r-project.org/web/packages/prediction/index.html ↗
predmixcor	1.1-1	https://cran.r-project.org/web/packages/predmixcor/index.html ↗

Package	Version	Link
PresenceAbsence	1.1.9	https://cran.r-project.org/web/packages/PresenceAbsence/index.html ↗
prettyunits	1.0.2	https://cran.r-project.org/web/packages/prettyunits/index.html ↗
pROC	1.11.0	https://cran.r-project.org/web/packages/pROC/index.html ↗
processmapR	0.3.3	https://cran.r-project.org/web/packages/processmapR/index.html ↗
processmonitR	0.1.0	https://cran.r-project.org/web/packages/processmonitR/index.html ↗
processx	2.0.0.1	https://cran.r-project.org/web/packages/processx/index.html ↗
prodlim	1.6.1	https://cran.r-project.org/web/packages/prodlim/index.html ↗
profDPM	3.3	https://cran.r-project.org/web/packages/profDPM/index.html ↗
profileModel	0.5-9	https://cran.r-project.org/web/packages/profileModel/index.html ↗
progress	1.1.2	https://cran.r-project.org/web/packages/progress/index.html ↗
proj4	1.0-8	https://cran.r-project.org/web/packages/proj4/index.html ↗
promises	1.1.0	https://cran.r-project.org/web/packages/promises/index.html ↗
prophet	0.2.1	https://cran.r-project.org/web/packages/prophet/index.html ↗
proto	1.0.0	https://cran.r-project.org/web/packages/proto/index.html ↗
protolite	1.7	https://cran.r-project.org/web/packages/protolite/index.html ↗
proxy	0.4-21	https://cran.r-project.org/web/packages/proxy/index.html ↗
pryr	0.1.4	https://cran.r-project.org/web/packages/pryr/index.html ↗
pscl	1.5.2	https://cran.r-project.org/web/packages/pscl/index.html ↗
psych	1.8.3.3	https://cran.r-project.org/web/packages/psych/index.html ↗
purrr	0.3.3	https://cran.r-project.org/web/packages/purrr/index.html ↗
pwr	1.2-2	https://cran.r-project.org/web/packages/pwr/index.html ↗
qap	0.1-1	https://cran.r-project.org/web/packages/qap/index.html ↗
qcc	2.7	https://cran.r-project.org/web/packages/qcc/index.html ↗

Package	Version	Link
qdapDictionaries	1.0.7	https://cran.r-project.org/web/packages/qdapDictionaries/index.html ↗
qdapRegex	0.7.2	https://cran.r-project.org/web/packages/qdapRegex/index.html ↗
qdapTools	1.3.3	https://cran.r-project.org/web/packages/qdapTools/index.html ↗
qgraph	1.4.4	https://cran.r-project.org/web/packages/qgraph/index.html ↗
qicharts	0.5.5	https://cran.r-project.org/web/packages/qicharts/index.html ↗
qicharts2	0.6.0	https://cran.r-project.org/web/packages/qicharts2/index.html ↗
quadprog	1.5-5	https://cran.r-project.org/web/packages/quadprog/index.html ↗
qualityTools	1.55	https://cran.r-project.org/web/packages/qualityTools/index.html ↗
quanteda	1.1.1	https://cran.r-project.org/web/packages/quanteda/index.html ↗
quantmod	0.4-12	https://cran.r-project.org/web/packages/quantmod/index.html ↗
quantreg	5.35	https://cran.r-project.org/web/packages/quantreg/index.html ↗
questionr	0.6.2	https://cran.r-project.org/web/packages/questionr/index.html ↗
qvcalc	0.9-1	https://cran.r-project.org/web/packages/qvcalc/index.html ↗
R.matlab	3.6.1	https://cran.r-project.org/web/packages/R.matlab/index.html ↗
R.methodsS3	1.7.1	https://cran.r-project.org/web/packages/R.methodsS3/index.html ↗
R.oo	1.21.0	https://cran.r-project.org/web/packages/R.oo/index.html ↗
R.utils	2.6.0	https://cran.r-project.org/web/packages/R.utils/index.html ↗
r2d3	0.2.3	https://cran.r-project.org/web/packages/r2d3/index.html ↗
R2HTML	2.3.2	https://cran.r-project.org/web/packages/R2HTML/index.html ↗
R2jags	0.5-7	https://cran.r-project.org/web/packages/R2jags/index.html ↗
R2OpenBUGS	3.2-3.2	https://cran.r-project.org/web/packages/R2OpenBUGS/index.html ↗

Package	Version	Link
R2WinBUGS	2.1-21	https://cran.r-project.org/web/packages/R2WinBUGS/index.html ↗
R6	2.2.2	https://cran.r-project.org/web/packages/R6/index.html ↗
ramps	0.6-15	https://cran.r-project.org/web/packages/ramps/index.html ↗
RandomFields	3.1.50	https://cran.r-project.org/web/packages/RandomFields/index.html ↗
RandomFieldsUtils	0.3.25	https://cran.r-project.org/web/packages/RandomFieldsUtils/index.html ↗
randomForest	4.6-14	https://cran.r-project.org/web/packages/randomForest/index.html ↗
ranger	0.9.0	https://cran.r-project.org/web/packages/ranger/index.html ↗
RApiDatetime	0.0.4	https://cran.r-project.org/web/packages/RApiDatetime/index.html ↗
rapdirs	0.3.1	https://cran.r-project.org/web/packages/rapdirs/index.html ↗
RArcInfo	0.4-12	https://cran.r-project.org/web/packages/RArcInfo/index.html ↗
raster	2.6-7	https://cran.r-project.org/web/packages/raster/index.html ↗
rattle	5.1.0	https://cran.r-project.org/web/packages/rattle/index.html ↗
rayshader	0.10.1	https://cran.r-project.org/web/packages/rayshader/index.html ↗
rbenchmark	1.0.0	https://cran.r-project.org/web/packages/rbenchmark/index.html ↗
Rblpapi	0.3.8	https://cran.r-project.org/web/packages/Rblpapi/index.html ↗
rbokeh	0.5.0	https://cran.r-project.org/web/packages/rbokeh/index.html ↗
rbugs	0.5-9	https://cran.r-project.org/web/packages/rbugs/index.html ↗
RColorBrewer	1.1-2	https://cran.r-project.org/web/packages/RColorBrewer/index.html ↗
Rcpp	1.0.1	https://cran.r-project.org/web/packages/Rcpp/index.html ↗
RcppArmadillo	0.8.400.0.0	https://cran.r-project.org/web/packages/RcppArmadillo/index.html ↗
rcppbugs	0.1.4.2	https://cran.r-project.org/web/packages/rcppbugs/index.html ↗

Package	Version	Link
RcppDE	0.1.5	https://cran.r-project.org/web/packages/RcppDE/index.html ↗
RcppEigen	0.3.3.4.0	https://cran.r-project.org/web/packages/RcppEigen/index.html ↗
RcppExamples	0.1.8	https://cran.r-project.org/web/packages/RcppExamples/index.html ↗
RcppParallel	4.4.0	https://cran.r-project.org/web/packages/RcppParallel/index.html ↗
RcppProgress	0.4	https://cran.r-project.org/web/packages/RcppProgress/index.html ↗
RcppRoll	0.2.2	https://cran.r-project.org/web/packages/RcppRoll/index.html ↗
RCurl	1.95-4.10	https://cran.r-project.org/web/packages/RCurl/index.html ↗
readbitmap	0.1-4	https://cran.r-project.org/web/packages/readbitmap/index.html ↗
readr	1.1.1	https://cran.r-project.org/web/packages/readr/index.html ↗
readxl	1.0.0	https://cran.r-project.org/web/packages/readxl/index.html ↗
recipes	0.1.2	https://cran.r-project.org/web/packages/recipes/index.html ↗
Redmonder	0.2.0	https://cran.r-project.org/web/packages/Redmonder/index.html ↗
registry	0.5	https://cran.r-project.org/web/packages/registry/index.html ↗
relaimpo	2.2-3	https://cran.r-project.org/web/packages/relaimpo/index.html ↗
relimp	1.0-5	https://cran.r-project.org/web/packages/relimp/index.html ↗
rematch	1.0.1	https://cran.r-project.org/web/packages/rematch/index.html ↗
Renext	3.1-0	https://cran.r-project.org/web/packages/Renext/index.html ↗
reports	0.1.4	https://cran.r-project.org/web/packages/reports/index.html ↗
reprex	0.1.2	https://cran.r-project.org/web/packages/reprex/index.html ↗
reshape	0.8.7	https://cran.r-project.org/web/packages/reshape/index.html ↗
reshape2	1.4.3	https://cran.r-project.org/web/packages/reshape2/index.html ↗
reticulate	1.6	https://cran.r-project.org/web/packages/reticulate/index.html ↗

Package	Version	Link
RevoIOQ	8.0.10	NA
RevoMods	11.0.0	NA
RevoUtils	10.0.9	NA
RevoUtilsMath	10.0.1	NA
rex	1.1.2	https://cran.r-project.org/web/packages/rex/index.html ↗
rFerns	2.0.3	https://cran.r-project.org/web/packages/rFerns/index.html ↗
rfm	0.2.0	https://cran.r-project.org/web/packages/rfm/index.html ↗
RGA	0.4.2	https://cran.r-project.org/web/packages/RGA/index.html ↗
rgdal	1.2-18	https://cran.r-project.org/web/packages/rgdal/index.html ↗
rgeos	0.3-26	https://cran.r-project.org/web/packages/rgeos/index.html ↗
rgexf	0.15.3	https://cran.r-project.org/web/packages/rgexf/index.html ↗
rgl	0.99.16	https://cran.r-project.org/web/packages/rgl/index.html ↗
RgoogleMaps	1.4.1	https://cran.r-project.org/web/packages/RgoogleMaps/index.html ↗
RGraphics	2.0-14	https://cran.r-project.org/web/packages/RGraphics/index.html ↗
RGtk2	2.20.34	https://cran.r-project.org/web/packages/RGtk2/index.html ↗
RInside	0.2.14	https://cran.r-project.org/web/packages/RInside/index.html ↗
RJaCGH	2.0.4	https://cran.r-project.org/web/packages/RJaCGH/index.html ↗
rjags	4-6	https://cran.r-project.org/web/packages/rjags/index.html ↗
rjson	0.2.15	https://cran.r-project.org/web/packages/rjson/index.html ↗
RJSONIO	1.3-0	https://cran.r-project.org/web/packages/RJSONIO/index.html ↗
rlang	0.4.1	https://cran.r-project.org/web/packages/rlang/index.html ↗
rlecuyer	0.3-4	https://cran.r-project.org/web/packages/rlecuyer/index.html ↗
rlist	0.4.6.1	https://cran.r-project.org/web/packages/rlist/index.html ↗
rmapshaper	0.3.0	https://cran.r-project.org/web/packages/rmapshaper/index.html ↗

Package	Version	Link
rmarkdown	1.9	https://cran.r-project.org/web/packages/rmarkdown/index.html ↗
Rmisc	1.5	https://cran.r-project.org/web/packages/Rmisc/index.html ↗
Rmpfr	0.7-0	https://cran.r-project.org/web/packages/Rmpfr/index.html ↗
rms	5.1-2	https://cran.r-project.org/web/packages/rms/index.html ↗
RMySQL	0.10.14	https://cran.r-project.org/web/packages/RMySQL/index.html ↗
rngtools	1.2.4	https://cran.r-project.org/web/packages/rngtools/index.html ↗
robCompositions	2.0.6	https://cran.r-project.org/web/packages/robCompositions/index.html ↗
robfilter	4.1	https://cran.r-project.org/web/packages/robfilter/index.html ↗
robustbase	0.92-8	https://cran.r-project.org/web/packages/robustbase/index.html ↗
robustHD	0.5.1	https://cran.r-project.org/web/packages/robustHD/index.html ↗
ROCR	1.0-7	https://cran.r-project.org/web/packages/ROCR/index.html ↗
RODBC	1.3-15	https://cran.r-project.org/web/packages/RODBC/index.html ↗
Rook	1.1-1	https://cran.r-project.org/web/packages/Rook/index.html ↗
rootSolve	1.7	https://cran.r-project.org/web/packages/rootSolve/index.html ↗
roxygen2	6.0.1	https://cran.r-project.org/web/packages/roxygen2/index.html ↗
rpart	4.1-13	https://cran.r-project.org/web/packages/rpart/index.html ↗
rpart.plot	2.1.2	https://cran.r-project.org/web/packages/rpart.plot/index.html ↗
rpivotTable	0.3.0	https://cran.r-project.org/web/packages/rpivotTable/index.html ↗
rprojroot	1.3-2	https://cran.r-project.org/web/packages/rprojroot/index.html ↗
rrcov	1.4-3	https://cran.r-project.org/web/packages/rrcov/index.html ↗
rscproxy	2.0-5	https://cran.r-project.org/web/packages/rscproxy/index.html ↗
rsdmx	0.5-11	https://cran.r-project.org/web/packages/rsdmx/index.html ↗

Package	Version	Link
RSGHB	1.1.2	https://cran.r-project.org/web/packages/RSGHB/index.html ↗
RSiteCatalyst	1.4.14	https://cran.r-project.org/web/packages/RSiteCatalyst/index.html ↗
RSNNS	0.4-10	https://cran.r-project.org/web/packages/RSNNS/index.html ↗
Rsolnp	1.16	https://cran.r-project.org/web/packages/Rsolnp/index.html ↗
RSpectra	0.12-0	https://cran.r-project.org/web/packages/RSpectra/index.html ↗
RSQLite	2.1.0	https://cran.r-project.org/web/packages/RSQLite/index.html ↗
rstan	2.17.3	https://cran.r-project.org/web/packages/rstan/index.html ↗
rstudioapi	0.7	https://cran.r-project.org/web/packages/rstudioapi/index.html ↗
rsvg	1.1	https://cran.r-project.org/web/packages/rsvg/index.html ↗
RTextTools	1.4.2	https://cran.r-project.org/web/packages/RTextTools/index.html ↗
Rttf2pt1	1.3.6	https://cran.r-project.org/web/packages/Rttf2pt1/index.html ↗
RUnit	0.4.31	https://cran.r-project.org/web/packages/RUnit/index.html ↗
runjags	2.0.4-2	https://cran.r-project.org/web/packages/runjags/index.html ↗
Runuran	0.24	https://cran.r-project.org/web/packages/Runuran/index.html ↗
rvcheck	0.0.9	https://cran.r-project.org/web/packages/rvcheck/index.html ↗
rvest	0.3.2	https://cran.r-project.org/web/packages/rvest/index.html ↗
rworldmap	1.3-6	https://cran.r-project.org/web/packages/rworldmap/index.html ↗
rworldxtra	1.01	https://cran.r-project.org/web/packages/rworldxtra/index.html ↗
SampleSizeMeans	1.1	https://cran.r-project.org/web/packages/SampleSizeMeans/index.html ↗
SampleSizeProportions	1.0	https://cran.r-project.org/web/packages/SampleSizeProportions/index.html ↗
sandwich	2.4-0	https://cran.r-project.org/web/packages/sandwich/index.html ↗
sas7bdat	0.5	https://cran.r-project.org/web/packages/sas7bdat/index.html ↗

Package	Version	Link
satellite	1.0.1	https://cran.r-project.org/web/packages/satellite/index.html ↗
sbgcop	0.975	https://cran.r-project.org/web/packages/sbgcop/index.html ↗
scales	1.0.0	https://cran.r-project.org/web/packages/scales/index.html ↗
scatterplot3d	0.3-41	https://cran.r-project.org/web/packages/scatterplot3d/index.html ↗
sciplot	1.1-1	https://cran.r-project.org/web/packages/sciplot/index.html ↗
segmented	0.5-3.0	https://cran.r-project.org/web/packages/segmented/index.html ↗
selectr	0.4-0	https://cran.r-project.org/web/packages/selectr/index.html ↗
sem	3.1-9	https://cran.r-project.org/web/packages/sem/index.html ↗
sentimentr	2.7.1	https://cran.r-project.org/web/packages/sentimentr/index.html ↗
seqinr	3.6-1	https://cran.r-project.org/web/packages/seqinr/index.html ↗
seriation	1.2-3	https://cran.r-project.org/web/packages/seriation/index.html ↗
setRNG	2013.9-1	https://cran.r-project.org/web/packages/setRNG/index.html ↗
sf	0.7-4	https://cran.r-project.org/web/packages/sf/index.html ↗
sfsmisc	1.1-2	https://cran.r-project.org/web/packages/sfsmisc/index.html ↗
sgeostat	1.0-27	https://cran.r-project.org/web/packages/sgeostat/index.html ↗
shape	1.4.4	https://cran.r-project.org/web/packages/shape/index.html ↗
shapefiles	0.7	https://cran.r-project.org/web/packages/shapefiles/index.html ↗
shiny	1.0.5	https://cran.r-project.org/web/packages/shiny/index.html ↗
shinyBS	0.61	https://cran.r-project.org/web/packages/shinyBS/index.html ↗
shinycssloaders	0.2.0	https://cran.r-project.org/web/packages/shinycssloaders/index.html ↗
shinyjs	1.0	https://cran.r-project.org/web/packages/shinyjs/index.html ↗
shinyTime	0.2.1	https://cran.r-project.org/web/packages/shinyTime/index.html ↗
showtext	0.5-1	https://cran.r-project.org/web/packages/showtext/index.html ↗

Package	Version	Link
showtextdb	2.0	https://cran.r-project.org/web/packages/showtextdb/index.html ↗
SIS	0.8-6	https://cran.r-project.org/web/packages/SIS/index.html ↗
SixSigma	0.9-51	https://cran.r-project.org/web/packages/SixSigma/index.html ↗
sjlabelled	1.0.8	https://cran.r-project.org/web/packages/sjlabelled/index.html ↗
sjmisc	2.7.1	https://cran.r-project.org/web/packages/sjmisc/index.html ↗
sjPlot	2.4.1	https://cran.r-project.org/web/packages/sjPlot/index.html ↗
sjstats	0.14.2-3	https://cran.r-project.org/web/packages/sjstats/index.html ↗
skmeans	0.2-11	https://cran.r-project.org/web/packages/skmeans/index.html ↗
slam	0.1-42	https://cran.r-project.org/web/packages/slam/index.html ↗
sm	2.2-5.4	https://cran.r-project.org/web/packages/sm/index.html ↗
smooth	2.4.1	https://cran.r-project.org/web/packages/smooth/index.html ↗
smoothSurv	2.0	https://cran.r-project.org/web/packages/smoothSurv/index.html ↗
sna	2.4	https://cran.r-project.org/web/packages/sna/index.html ↗
snakecase	0.9.1	https://cran.r-project.org/web/packages/snakecase/index.html ↗
snow	0.4-2	https://cran.r-project.org/web/packages/snow/index.html ↗
SnowballC	0.5.1	https://cran.r-project.org/web/packages/SnowballC/index.html ↗
snowFT	1.6-0	https://cran.r-project.org/web/packages/snowFT/index.html ↗
sodium	1.1	https://cran.r-project.org/web/packages/sodium/index.html ↗
sourcetools	0.1.6	https://cran.r-project.org/web/packages/sourcetools/index.html ↗
sp	1.2-7	https://cran.r-project.org/web/packages/sp/index.html ↗
spacetime	1.2-1	https://cran.r-project.org/web/packages/spacetime/index.html ↗
spacyr	0.9.6	https://cran.r-project.org/web/packages/spacyr/index.html ↗

Package	Version	Link
spam	2.1-3	https://cran.r-project.org/web/packages/spam/index.html ↗
SparseM	1.77	https://cran.r-project.org/web/packages/SparseM/index.html ↗
sparsepp	0.2.0	https://cran.r-project.org/web/packages/sparsepp/index.html ↗
spatial	7.3-11	https://cran.r-project.org/web/packages/spatial/index.html ↗
spatstat	1.55-0	https://cran.r-project.org/web/packages/spatstat/index.html ↗
spatstat.data	1.2-0	https://cran.r-project.org/web/packages/spatstat.data/index.html ↗
spatstat.utils	1.8-0	https://cran.r-project.org/web/packages/spatstat.utils/index.html ↗
spBayes	0.4-1	https://cran.r-project.org/web/packages/spBayes/index.html ↗
spData	0.2.8.3	https://cran.r-project.org/web/packages/spData/index.html ↗
spdep	0.7-4	https://cran.r-project.org/web/packages/spdep/index.html ↗
spikeslab	1.1.5	https://cran.r-project.org/web/packages/spikeslab/index.html ↗
splancs	2.01-40	https://cran.r-project.org/web/packages/splancs/index.html ↗
splines	3.4.4	https://cran.r-project.org/web/packages/splines/index.html ↗
spls	2.2-2	https://cran.r-project.org/web/packages/spls/index.html ↗
splus2R	1.2-2	https://cran.r-project.org/web/packages/splus2R/index.html ↗
spTimer	3.0-1	https://cran.r-project.org/web/packages/spTimer/index.html ↗
sqldf	0.4-11	https://cran.r-project.org/web/packages/sqldf/index.html ↗
SQUAREM	2017.10-1	https://cran.r-project.org/web/packages/SQUAREM/index.html ↗
sROC	0.1-2	https://cran.r-project.org/web/packages/sROC/index.html ↗
stabledist	0.7-1	https://cran.r-project.org/web/packages/stabledist/index.html ↗
stabs	0.6-3	https://cran.r-project.org/web/packages/stabs/index.html ↗
StanHeaders	2.17.2	https://cran.r-project.org/web/packages/StanHeaders/index.html ↗
statmod	1.4.30	https://cran.r-project.org/web/packages/statmod/index.html ↗

Package	Version	Link
statnet.common	4.0.0	https://cran.r-project.org/web/packages/statnet.common/index.html ↗
stats	3.4.4	NA
stats4	3.4.4	NA
stepPlr	0.93	https://cran.r-project.org/web/packages/stepPlr/index.html ↗
stinepack	1.4	https://cran.r-project.org/web/packages/stinepack/index.html ↗
stochvol	1.3.3	https://cran.r-project.org/web/packages/stochvol/index.html ↗
stopwords	0.9.0	https://cran.r-project.org/web/packages/stopwords/index.html ↗
stringdist	0.9.4.7	https://cran.r-project.org/web/packages/stringdist/index.html ↗
stringi	1.1.7	https://cran.r-project.org/web/packages/stringi/index.html ↗
stringr	1.3.0	https://cran.r-project.org/web/packages/stringr/index.html ↗
strucchange	1.5-1	https://cran.r-project.org/web/packages/strucchange/index.html ↗
stsm	1.9	https://cran.r-project.org/web/packages/stsm/index.html ↗
stsm.class	1.3	https://cran.r-project.org/web/packages/stsm.class/index.html ↗
sugrrants	0.2.4	https://cran.r-project.org/web/packages/sugrrants/index.html ↗
sunburstR	2.0.0	https://cran.r-project.org/web/packages/sunburstR/index.html ↗
SuppDists	1.1-9.4	https://cran.r-project.org/web/packages/SuppDists/index.html ↗
survey	3.33-2	https://cran.r-project.org/web/packages/survey/index.html ↗
survival	2.41-3	https://cran.r-project.org/web/packages/survival/index.html ↗
survminer	0.4.6	https://cran.r-project.org/web/packages/survminer/index.html ↗
survMisc	0.5.4	https://cran.r-project.org/web/packages/survMisc/index.html ↗
svglite	1.2.1	https://cran.r-project.org/web/packages/svglite/index.html ↗

Package	Version	Link
svmpath	0.955	https://cran.r-project.org/web/packages/svmpath/index.html ↗
svUnit	0.7-12	https://cran.r-project.org/web/packages/svUnit/index.html ↗
sweep	0.2.1	https://cran.r-project.org/web/packages/sweep/index.html ↗
sysfonts	0.7.2	https://cran.r-project.org/web/packages/sysfonts/index.html ↗
systemfit	1.1-20	https://cran.r-project.org/web/packages/systemfit/index.html ↗
syuzhet	1.0.4	https://cran.r-project.org/web/packages/syuzhet/index.html ↗
tau	0.0-20	https://cran.r-project.org/web/packages/tau/index.html ↗
tcltk	3.4.4	https://cran.r-project.org/web/packages/tcltk/index.html ↗
tcltk2	1.2-11	https://cran.r-project.org/web/packages/tcltk2/index.html ↗
TeachingDemos	2.10	https://cran.r-project.org/web/packages/TeachingDemos/index.html ↗
tensor	1.5	https://cran.r-project.org/web/packages/tensor/index.html ↗
tensorA	0.36	https://cran.r-project.org/web/packages/tensorA/index.html ↗
testthat	2.0.0	https://cran.r-project.org/web/packages/testthat/index.html ↗
text2vec	0.5.1	https://cran.r-project.org/web/packages/text2vec/index.html ↗
textcat	1.0-5	https://cran.r-project.org/web/packages/textcat/index.html ↗
textclean	0.9.3	https://cran.r-project.org/web/packages/textclean/index.html ↗
textir	2.0-5	https://cran.r-project.org/web/packages/textir/index.html ↗
textmineR	2.1.1	https://cran.r-project.org/web/packages/textmineR/index.html ↗
textshape	1.6.0	https://cran.r-project.org/web/packages/textshape/index.html ↗
tfplot	2015.12-1	https://cran.r-project.org/web/packages/tfplot/index.html ↗
tframe	2015.12-1	https://cran.r-project.org/web/packages/tframe/index.html ↗
tgp	2.4-14	https://cran.r-project.org/web/packages/tgp/index.html ↗
TH.data	1.0-8	https://cran.r-project.org/web/packages/TH.data/index.html ↗
threejs	0.3.1	https://cran.r-project.org/web/packages/threejs/index.html ↗

Package	Version	Link
tibble	2.1.1	https://cran.r-project.org/web/packages/tibble/index.html ↗
tibbletime	0.1.1	https://cran.r-project.org/web/packages/tibbletime/index.html ↗
tidycensus	0.4.1	https://cran.r-project.org/web/packages/tidycensus/index.html ↗
tidyrr	1.0.0	https://cran.r-project.org/web/packages/tidyrr/index.html ↗
tidyselect	0.2.5	https://cran.r-project.org/web/packages/tidyselect/index.html ↗
tidytext	0.1.8	https://cran.r-project.org/web/packages/tidytext/index.html ↗
tidyverse	1.2.1	https://cran.r-project.org/web/packages/tidyverse/index.html ↗
tiff	0.1-5	https://cran.r-project.org/web/packages/tiff/index.html ↗
tigris	0.6.2	https://cran.r-project.org/web/packages/tigris/index.html ↗
timeDate	3043.102	https://cran.r-project.org/web/packages/timeDate/index.html ↗
timelineS	0.1.1	https://cran.r-project.org/web/packages/timelineS/index.html ↗
timeSeries	3042.102	https://cran.r-project.org/web/packages/timeSeries/index.html ↗
timetk	0.1.0	https://cran.r-project.org/web/packages/timetk/index.html ↗
timevis	0.5	https://cran.r-project.org/web/packages/timevis/index.html ↗
tm	0.7-3	https://cran.r-project.org/web/packages/tm/index.html ↗
tmap	1.11-1	https://cran.r-project.org/web/packages/tmap/index.html ↗
tmaptools	1.2-3	https://cran.r-project.org/web/packages/tmaptools/index.html ↗
TMB	1.7.13	https://cran.r-project.org/web/packages/TMB/index.html ↗
tokenizers	0.2.1	https://cran.r-project.org/web/packages/tokenizers/index.html ↗
tools	3.4.4	NA
topicmodels	0.2-7	https://cran.r-project.org/web/packages/topicmodels/index.html ↗
TraMineR	2.0-8	https://cran.r-project.org/web/packages/TraMineR/index.html ↗

Package	Version	Link
translations	3.4.4	NA
tree	1.0-39	https://cran.r-project.org/web/packages/tree/index.html ↗
treemap	2.4-2	https://cran.r-project.org/web/packages/treemap/index.html ↗
trelliscopejs	0.1.18	https://cran.r-project.org/web/packages/trelliscopejs/index.html ↗
trimcluster	0.1-2	https://cran.r-project.org/web/packages/trimcluster/index.html ↗
truncnorm	1.0-8	https://cran.r-project.org/web/packages/truncnorm/index.html ↗
TSA	1.01	https://cran.r-project.org/web/packages/TSA/index.html ↗
tseries	0.10-43	https://cran.r-project.org/web/packages/tseries/index.html ↗
tsfa	2014.10-1	https://cran.r-project.org/web/packages/tsfa/index.html ↗
tsibble	0.8.5	https://cran.r-project.org/web/packages/tsibble/index.html ↗
tsintermittent	1.9	https://cran.r-project.org/web/packages/tsintermittent/index.html ↗
tsoutliers	0.6-6	https://cran.r-project.org/web/packages/tsoutliers/index.html ↗
TSP	1.1-5	https://cran.r-project.org/web/packages/TSP/index.html ↗
TSstudio	0.1.5	https://cran.r-project.org/web/packages/TSstudio/index.html ↗
TTR	0.23-3	https://cran.r-project.org/web/packages/TTR/index.html ↗
tweedie	2.3.2	https://cran.r-project.org/web/packages/tweedie/index.html ↗
tweenr	1.0.1	https://cran.r-project.org/web/packages/tweenr/index.html ↗
twitteR	1.1.9	https://cran.r-project.org/web/packages/twitteR/index.html ↗
udpipe	0.5	https://cran.r-project.org/web/packages/udpipe/index.html ↗
udunits2	0.13	https://cran.r-project.org/web/packages/udunits2/index.html ↗
units	0.6-2	https://cran.r-project.org/web/packages/units/index.html ↗
UpSetR	1.3.3	https://cran.r-project.org/web/packages/UpSetR/index.html ↗
urca	1.3-0	https://cran.r-project.org/web/packages/urca/index.html ↗
useful	1.2.3	https://cran.r-project.org/web/packages/useful/index.html ↗

Package	Version	Link
UsingR	2.0-5	https://cran.r-project.org/web/packages/UsingR/index.html ↗
usmap	0.2.1	https://cran.r-project.org/web/packages/usmap/index.html ↗
utf8	1.1.3	https://cran.r-project.org/web/packages/utf8/index.html ↗
utils	3.4.4	NA
uuid	0.1-2	https://cran.r-project.org/web/packages/uuid/index.html ↗
V8	2.2	https://cran.r-project.org/web/packages/V8/index.html ↗
vars	1.5-2	https://cran.r-project.org/web/packages/vars/index.html ↗
vcd	1.4-4	https://cran.r-project.org/web/packages/vcd/index.html ↗
vctrs	0.2.0	https://cran.r-project.org/web/packages/vctrs/index.html ↗
vdiffr	0.2.2	https://cran.r-project.org/web/packages/vdiffr/index.html ↗
vegan	2.4-6	https://cran.r-project.org/web/packages/vegan/index.html ↗
VennDiagram	1.6.20	https://cran.r-project.org/web/packages/VennDiagram/index.html ↗
VGAM	1.0-5	https://cran.r-project.org/web/packages/VGAM/index.html ↗
VIF	1.0	https://cran.r-project.org/web/packages/VIF/index.html ↗
VIM	4.7.0	https://cran.r-project.org/web/packages/VIM/index.html ↗
vioplot	0.2	https://cran.r-project.org/web/packages/vioplot/index.html ↗
viridis	0.5.1	https://cran.r-project.org/web/packages/viridis/index.html ↗
viridisLite	0.3.0	https://cran.r-project.org/web/packages/viridisLite/index.html ↗
visNetwork	2.0.3	https://cran.r-project.org/web/packages/visNetwork/index.html ↗
vistime	0.4.0	https://cran.r-project.org/web/packages/vistime/index.html ↗
waterfalls	0.1.2	https://cran.r-project.org/web/packages/waterfalls/index.html ↗
wavethresh	4.6.8	https://cran.r-project.org/web/packages/wavethresh/index.html ↗
webshot	0.5.0	https://cran.r-project.org/web/packages/webshot/index.html ↗

Package	Version	Link
webutils	0.6	https://cran.r-project.org/web/packages/webutils/index.html ↗
weco	1.1	https://cran.r-project.org/web/packages/weco/index.html ↗
WeibullR	1.0.10	https://cran.r-project.org/web/packages/WeibullR/index.html ↗
weights	0.85	https://cran.r-project.org/web/packages/weights/index.html ↗
whisker	0.3-2	https://cran.r-project.org/web/packages/whisker/index.html ↗
withr	2.1.2	https://cran.r-project.org/web/packages/withr/index.html ↗
wmtsa	2.0-3	https://cran.r-project.org/web/packages/wmtsa/index.html ↗
wordcloud	2.5	https://cran.r-project.org/web/packages/wordcloud/index.html ↗
wordcloud2	0.2.1	https://cran.r-project.org/web/packages/wordcloud2/index.html ↗
xesreadR	0.2.2	https://cran.r-project.org/web/packages/xesreadR/index.html ↗
xgboost	0.6.4.1	https://cran.r-project.org/web/packages/xgboost/index.html ↗
XML	3.98-1.10	https://cran.r-project.org/web/packages/XML/index.html ↗
xml2	1.2.0	https://cran.r-project.org/web/packages/xml2/index.html ↗
xplorerr	0.1.1	https://cran.r-project.org/web/packages/xplorerr/index.html ↗
xtable	1.8-2	https://cran.r-project.org/web/packages/xtable/index.html ↗
xts	0.10-2	https://cran.r-project.org/web/packages/xts/index.html ↗
yaml	2.1.18	https://cran.r-project.org/web/packages/yaml/index.html ↗
yarrr	0.1.5	https://cran.r-project.org/web/packages/yarrr/index.html ↗
YieldCurve	4.1	https://cran.r-project.org/web/packages/YieldCurve/index.html ↗
zeallot	0.1.0	https://cran.r-project.org/web/packages/zeallot/index.html ↗
zic	0.9.1	https://cran.r-project.org/web/packages/zic/index.html ↗
zipfR	0.6-10	https://cran.r-project.org/web/packages/zipfR/index.html ↗
zoo	1.8-1	https://cran.r-project.org/web/packages/zoo/index.html ↗

R scripts that aren't supported in Power BI

The following table shows which packages are not supported in the Power BI service.

Package	Request Date	Reason
RgoogleMaps	10/05/2016	Networking is blocked
mailR	10/03/2016	Networking is blocked
RevoScaleR	8/30/2016	Ships only with Microsoft R Server

Next steps

For more information about R in Power BI, take a look at the following articles:

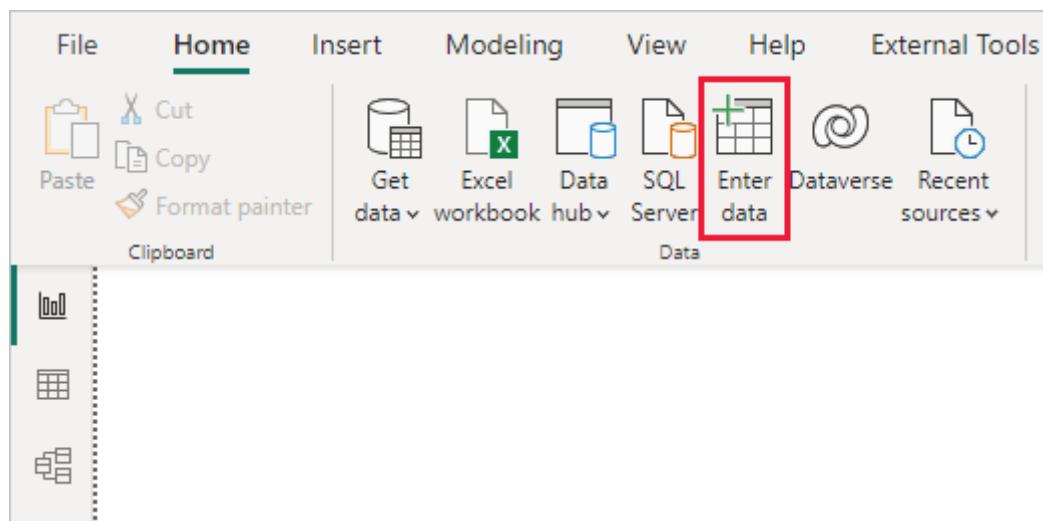
- [Creating R visuals in the Power BI service](#)
- [Create Power BI visuals using R](#)
- [Running R scripts in Power BI Desktop](#)
- [Using R in Power Query Editor](#)

Enter data directly into Power BI Desktop

Article • 03/20/2023

With Power BI Desktop, you can enter data directly and use that data in your reports and visualizations. For example, you can copy portions of a workbook or web page, then paste it into Power BI Desktop.

To enter data directly into Power BI Desktop in the form of a new table, select **Enter data** from the **Home** ribbon.

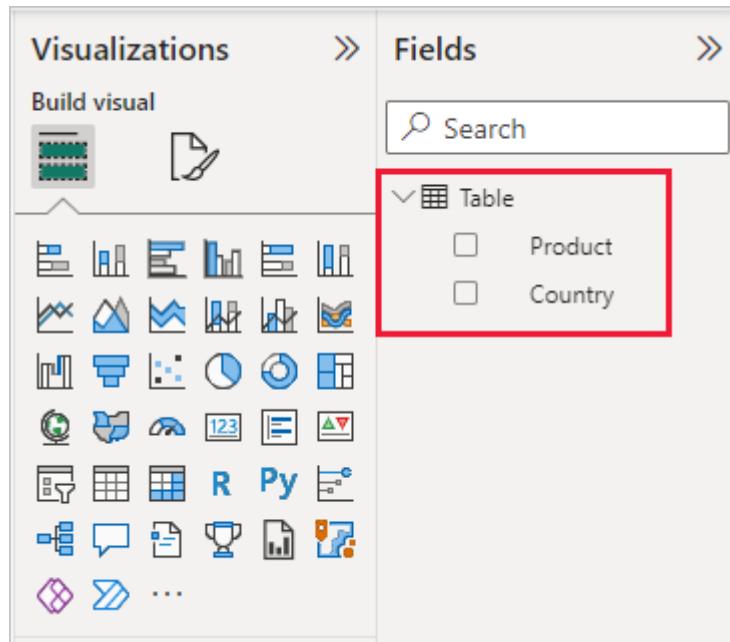


Power BI Desktop might attempt to make minor transformations on the data, if appropriate, just like it does when you load data from any source. For example, in the following case it promoted the first row of data to headers.

A screenshot of the 'Create Table' dialog box in Power BI Desktop. The title bar says 'Create Table'. A message bar at the top left says 'The first row of data that you pasted has been promoted to column headers.' with a 'Undo Headers' button. Below is a table preview showing 13 rows of data with columns 'Country' and 'Product'. At the bottom, there is a 'Name:' field containing 'Table' and buttons for 'Load', 'Edit', and 'Cancel'.

If you want to shape the data you entered or pasted, select **Edit** to open **Power Query Editor**. You can shape and transform the data before bringing it into Power BI Desktop. Select **Load** to import the data as it appears.

When you select **Load**, Power BI Desktop creates a new table from your data, and makes it available in the **Fields** pane. In the following image, Power BI Desktop shows your new table, called **Table**, and the two fields within that table that were created.



And that's it. It's that easy to enter data into Power BI Desktop.

You're now ready to use the data in Power BI Desktop. You can create visuals, reports, or interact with any other data you might want to connect with and import, such as Excel workbooks, databases, or any other data source.

ⓘ Note

To update, add, or delete data within items created by **Enter Data**, changes must be made in Power BI Desktop, and published. Data updates cannot be made directly from the Power BI service.

Next steps

There are all sorts of data you can connect to using Power BI Desktop. For more information on data sources, check out the following resources:

- [What is Power BI Desktop?](#)
- [Data sources in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)

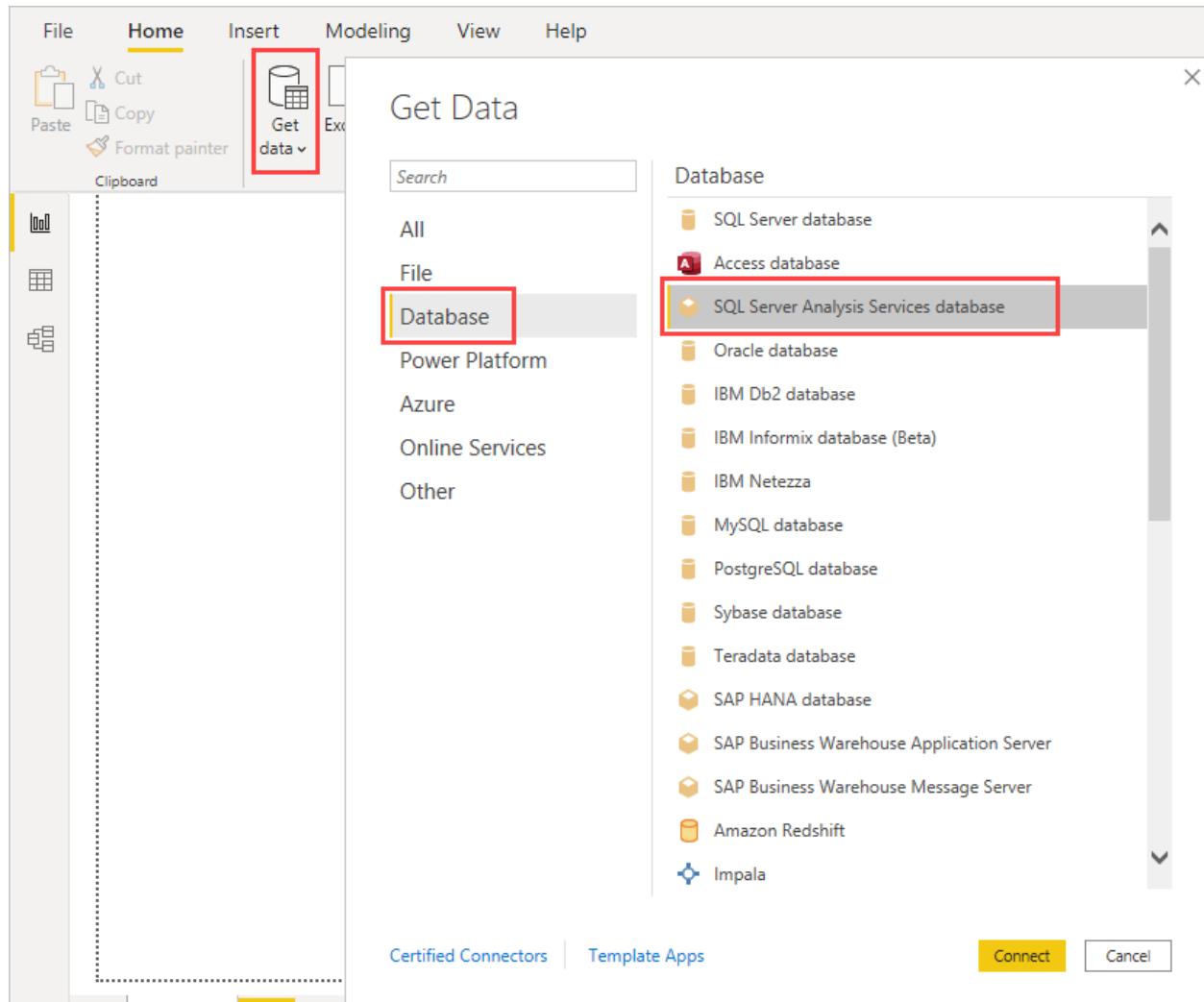
- Connect to Excel workbooks in Power BI Desktop
- Connect to CSV files in Power BI Desktop

Connect to SSAS multidimensional models in Power BI Desktop

Article • 01/23/2023

With Power BI Desktop, you can access SQL Server Analysis Services (SSAS) multidimensional models, commonly referred to as *SSAS MD*.

To connect to an SSAS MD database, select **Get data**, choose **Database > SQL Server Analysis Services database**, and then select **Connect**:



The Power BI service and Power BI Desktop both support SSAS multidimensional models in live connection mode. You can publish and upload reports that use **SSAS Multidimensional models** in live mode to the Power BI service.

Capabilities and features of SSAS MD

The following sections describe features and capabilities of Power BI and SSAS MD connections.

Tabular metadata of multidimensional models

The following table shows the correspondence between multidimensional objects and the tabular metadata that's returned to Power BI Desktop. Power BI queries the model for tabular metadata. Based on the returned metadata, Power BI Desktop runs appropriate DAX queries against SSAS when you create a visualization, such as a table, matrix, chart, or slicer.

BISM-Multidimensional object	Tabular Metadata
Cube	Model
Cube dimension	Table
Dimension attributes (keys), name	Columns
Measure group	Table
Measure	Measure
Measures without associated measure group	Within table called <i>Measures</i>
Measure group -> Cube dimension relationship	Relationship
Perspective	Perspective
KPI	KPI
User/parent-child hierarchies	Hierarchies

Measures, measure groups, and KPIs

Measure groups in a multidimensional cube are exposed as tables with a sigma (Σ) beside them in the **Fields** pane. Calculated measures without an associated measure group are grouped under a special table called *Measures* in the tabular metadata.

To help simplify complex models in a multidimensional model, you can define a set of measures or KPIs in a cube to be located within a *display folder*. Power BI recognizes display folders in tabular metadata, and it shows measures and KPIs within the display folders. KPIs in multidimensional databases support *Value*, *Goal*, *Status Graphic*, and *Trend Graphic*.

Dimension attribute type

Multidimensional models also support associating dimension attributes with specific dimension attribute types. For example, a **Geography** dimension where the *City*, *State*-

Province, *CountryRegion*, and *Postal Code* dimension attributes have appropriate geography types associated with them are exposed in the tabular metadata. Power BI recognizes the metadata, enabling you to create map visualizations. You can recognize these associations by the *map* icon next to the element in the **Field** pane in Power BI.

Power BI can also render images when you provide a field that contains uniform resource locators (URLs) of the images. You might specify these fields as *ImageURL* types in SQL Server Data Tools, or then in Power BI Desktop. Its type information is then provided to Power BI in the tabular metadata. Power BI can then retrieve those images from the URL and display them in visuals.

Parent-child hierarchies

Multidimensional models support parent-child hierarchies, which are presented as a *hierarchy* in the tabular metadata. Each level of the parent-child hierarchy is exposed as a hidden column in the tabular metadata. The key attribute of the parent-child dimension isn't exposed in the tabular metadata.

Dimension calculated members

Multidimensional models support creation of various types of *calculated members*. The two most common types of calculated members are:

- Calculated members on attribute hierarchies that aren't siblings of *All*
- Calculated members on user hierarchies

Multidimensional models expose *calculated members on attribute hierarchies* as values of a column. You have a few other options and constraints if you expose this type of calculated member:

- A dimension attribute can have an optional *UnknownMember*.
- An attribute containing calculated members can't be the key attribute of the dimension unless it's the only attribute of the dimension.
- An attribute containing calculated members can't be a parent-child attribute.

The calculated members of user hierarchies aren't exposed in Power BI. You can instead connect to a cube that contains calculated members on user hierarchies. However, you can't see calculated members if they don't meet the constraints that are mentioned in the previous bulleted list.

Security

Multidimensional models support dimension and cell level security by way of *roles*.

When you connect to a cube with Power BI, you're authenticated and evaluated for appropriate permissions. If a user has *dimension security* applied, the respective dimension members aren't seen by the user in Power BI. However, when a user has defined a *cell security* permission where certain cells are restricted, that user can't connect to the cube using Power BI.

Considerations and limitations

There are certain limitations to using SSAS MD:

- Only enterprise and BI editions of SQL Server 2014 support live connections. For the standard edition of SQL Server, SQL Server 2016 or later is required for live connections.
- *Actions* and *named sets* aren't exposed to Power BI. To create visuals and reports, you can still connect to cubes that also contain actions or named sets.
- When Power BI displays metadata for an SSAS model, occasionally you can't retrieve data from the model. This scenario can occur if you've installed the 32-bit version of the Microsoft Online Analytical Processing provider, but not the 64-bit version. Installing the 64-bit version might resolve the issue.
- You can't create *report level* measures when authoring a report that is connected live to an SSAS multidimensional model. The only measures that are available are measures defined in the MD model.

Supported features of SSAS MD in Power BI Desktop

Consumption of the following elements is supported in this release of SSAS MD. For more information about these features, see [Understanding power view for multidimensional models](#).

- Default members
- Dimension attributes
- Dimension attribute types
- Dimension calculated members, which:
 - must be a single real member when the dimension has more than one attribute;
 - can't be the key attribute of the dimension unless it's the only attribute; and
 - can't be a parent-child attribute.

- Dimension security
- Display folders
- Hierarchies
- ImageUrls
- KPIs
- KPI trends
- Measures (with or without measure groups)
- Measures as variant

Troubleshooting

The following list describes all known issues when connecting to SQL Server Analysis Services.

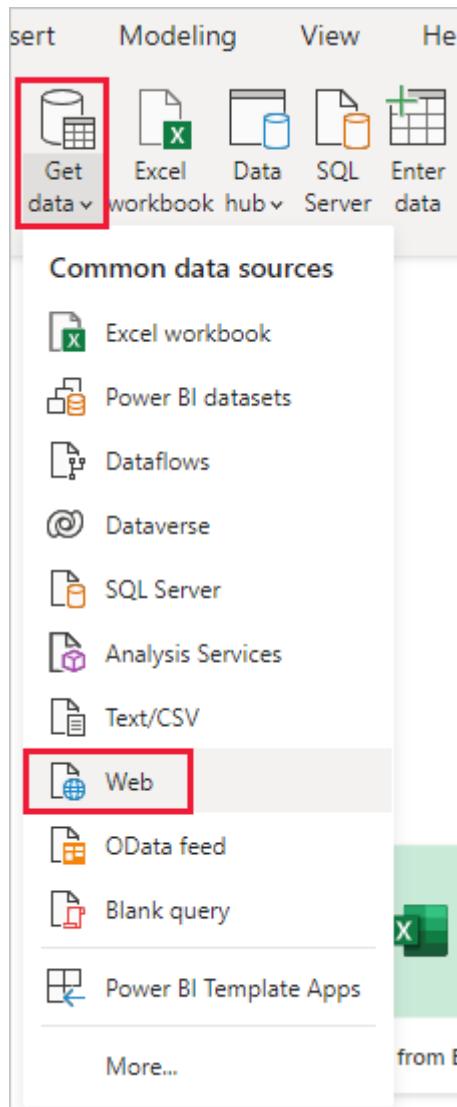
- **Error : Couldn't load model schema.** This error usually occurs when the user connecting to Analysis Services doesn't have access to database/cube.

Connect to webpages from Power BI Desktop

Article • 03/20/2023

You can connect to a webpage and import its data into Power BI Desktop, to use in your visuals and in your data models.

In Power BI Desktop, select **Get data > Web** from the **Home** ribbon.



A dialog appears, asking for the URL of the webpage from which you want to import data.

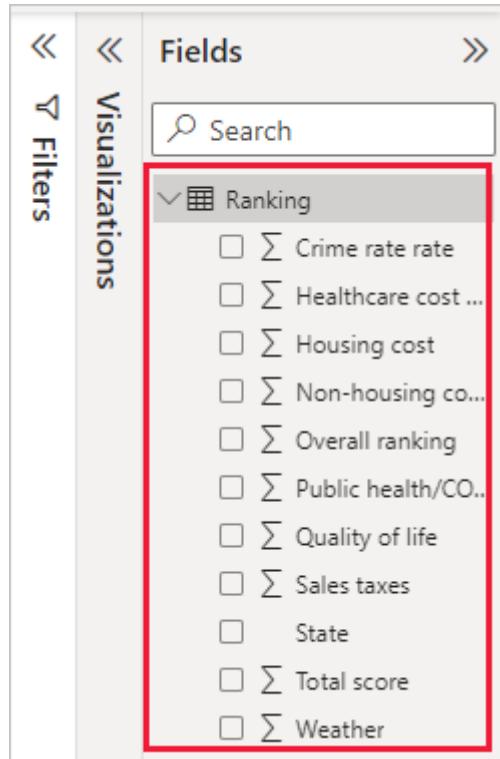


Once you've typed or pasted the URL, select **OK**.

Power BI Desktop connects to the webpage and then presents the page's available data in the **Navigator** window. When you select one of the available data elements, such as **Table 1**, the **Navigator** window displays a preview of that data on the right side of the window.

You can choose the **Transform Data** button, which launches **Power Query Editor**, where you can shape and transform the data on that webpage before importing it into Power BI Desktop. Or you can select the **Load** button, and import all of the data elements you selected in the left pane.

When you select **Load**, Power BI Desktop imports the selected items, and makes them available in the **Fields** pane, found on the right side of the Reports view in Power BI Desktop.



That's all there is to connecting to a webpage and bringing its data into Power BI Desktop.

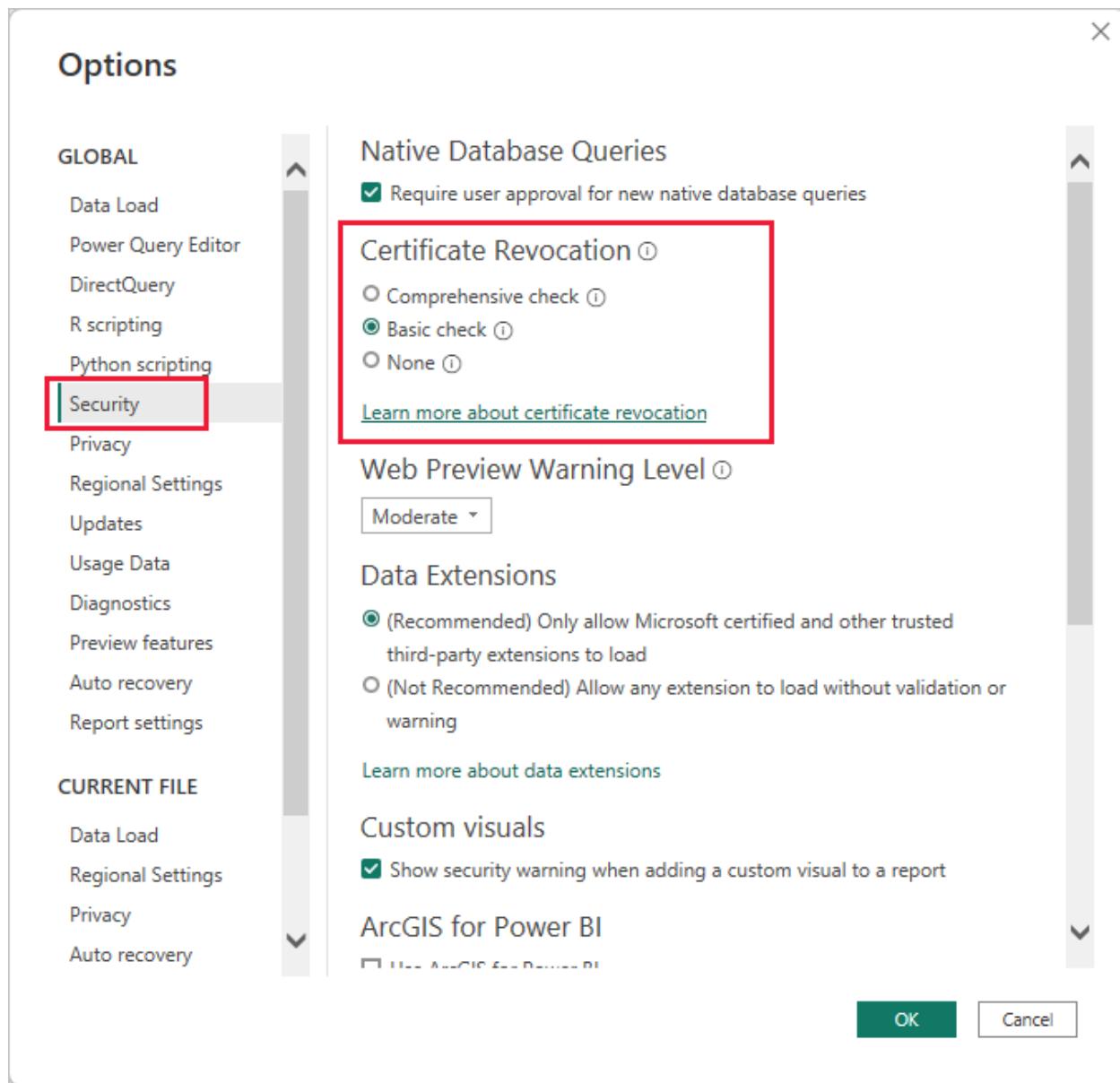
From there, you can drag those fields onto the Report canvas and create all the visualizations you want. You can also use the data from that webpage just like you would any other data. You can shape it, you can create relationships between it and other data sources in your model, and otherwise do what you like to create the Power BI report you want.

To see connecting to a webpage in more depth and action, take a look at the [Power BI Desktop Getting Started Guide](#).

Certificate revocation check

Power BI applies security for web connections to protect your data. In some scenarios, such as capturing web requests with Fiddler, web connections may not work properly. To enable such scenarios, you can modify the **Check if your certificates have been revoked** option in Power BI Desktop, then restart Power BI Desktop.

To change this option, select **File > Options and settings > Options**, then select **Security** in the left pane.



Next steps

There are all sorts of data you can connect to using Power BI Desktop. For more information on data sources, check out the following resources:

- [Data Sources in Power BI Desktop](#)
- [Shape and Combine Data with Power BI Desktop](#)
- [Connect to Excel workbooks in Power BI Desktop](#)
- [Connect to CSV files in Power BI Desktop](#)
- [Enter data directly into Power BI Desktop](#)

Connect to Snowflake in the Power BI service

Article • 08/09/2023

Connecting to Snowflake in the Power BI service differs from other connectors in only one way. Snowflake has a capability for Azure Active Directory (Azure AD), an option for SSO (single sign-on). Parts of the integration require different administrative roles across Snowflake, Power BI, and Azure. You can choose to enable Azure AD authentication without using SSO. Basic authentication works similarly to other connectors in the service.

To configure Azure AD integration and optionally enable SSO:

- If you're the Snowflake admin, see [Power BI SSO to Snowflake](#) in the Snowflake documentation.
- If you're a Power BI admin, go to the [Admin portal](#) section to enable SSO.
- If you're a Power BI dataset creator, go to the [Configure a dataset with Azure AD](#) section to enable SSO.

Power BI service configuration

Admin portal

To enable SSO, a global admin has to turn on the setting in the Power BI Admin portal. This setting approves sending Azure AD authentication tokens to Snowflake from within the Power BI service. This setting is set at an organizational level. Follow these steps to enable SSO:

1. [Sign in to Power BI](#) using global admin credentials.
2. Select **Settings** from the page header menu, then select **Admin portal**.
3. Select **Tenant settings**, then scroll to locate **Integration settings**.

The screenshot shows the 'Admin portal' interface. On the left, a sidebar lists various settings: Usage metrics, Users, Audit logs, Tenant settings (which is selected), Capacity settings, Refresh summary, Embed Codes, Organizational visuals, Dataflow settings, Workspaces, Custom branding, Protection metrics, and Featured content. The main content area is titled 'Integration settings' and contains several items: 'Use Analyze in Excel with on-premises datasets' (Enabled for the entire organization), 'Use ArcGIS Maps for Power BI' (Enabled for the entire organization), 'Use global search for Power BI (Preview)' (Enabled for the entire organization), and 'Snowflake SSO' (Unapplied changes). A note below states: 'Enable SSO capability for Snowflake. By enabling, user access token information, including name and email, will be sent to Snowflake for authentication.' A yellow 'Learn more' link is present. A red box highlights the 'Snowflake SSO' section, which includes a toggle switch set to 'Enabled', an 'Apply' button, a 'Cancel' button, and a note at the bottom stating '(i) This setting applies to the entire organization'.

4. Expand **Snowflake SSO**, toggle the setting to **Enabled**, then select **Apply**.

This step is required to consent to sending your Azure AD token to the Snowflake servers. After you enable the setting, it can take up to an hour for it to take effect.

After SSO is enabled, you can use reports with SSO.

Configure a dataset with Azure AD

After a report that's based on the Snowflake connector is published to the Power BI service, the dataset creator has to update settings for the appropriate workspace so it can use SSO.

For more information including steps for using Azure AD, SSO, and Snowflake, see [Data gateway support for single sign-on with Azure AD](#).

For information about how you can use the on-premises data gateway, see [What is an on-premises data gateway?](#)

If you aren't using the gateway, you're all set. When you have Snowflake credentials configured on your on-premises data gateway, but you're only using that data source in your model, switch the Dataset settings to off on the gateway for that data model.

▲ Gateway connection

You don't need a gateway for this dataset, because all of its data sources are in the cloud, but you can use a gateway for enhanced control over how you connect. [Learn more](#)

Use a data gateway



Apply

Discard

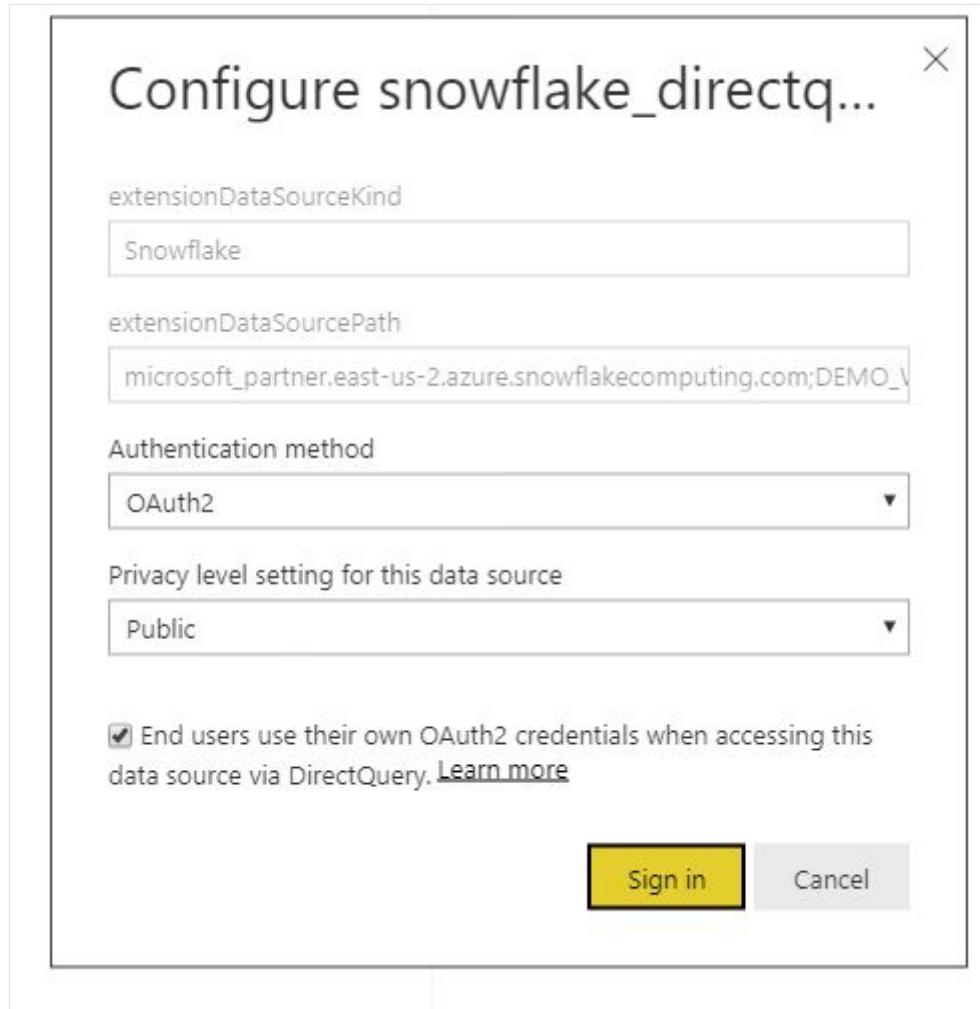
To turn on SSO for a dataset:

1. [Sign in to Power BI](#) using dataset creator credentials.
2. Select the appropriate workspace, then choose **Settings** from the more options menu that's located next to the dataset name.

The screenshot shows a list of datasets in a workspace. The 'Datasets + dataflows' tab is selected. A dataset named 'TeamsPower' is listed. To the right of the dataset name is a more options menu icon (three dots) which is highlighted with a red box. A dropdown menu is open from this icon, listing several options: 'Analyze in Excel', 'Create report', 'Delete', 'Get quick insights', 'Security', 'Rename', 'Settings' (which is also highlighted with a red box), 'Download .pbix', 'Manage permissions', and 'View lineage'. The 'Owner' column for the dataset shows 'Teams Power'.

All	Content	Datasets + dataflows						
		<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Owner</th></tr></thead><tbody><tr><td>TeamsPower</td><td>Dataset</td><td>Teams Power</td></tr></tbody></table>	Name	Type	Owner	TeamsPower	Dataset	Teams Power
Name	Type	Owner						
TeamsPower	Dataset	Teams Power						

3. Select **Data source credentials** and sign in. The dataset can be signed into Snowflake with Basic or OAuth2 (Azure AD) credentials. By using Azure AD, you can enable SSO in the next step.
4. Select the option **End users use their own OAuth2 credentials when accessing this data source via DirectQuery**. This setting will enable Azure AD SSO. The Azure AD credentials are sent for SSO.



After these steps are done, users should automatically use their Azure AD authentication to connect to data from that Snowflake dataset.

If you choose not to enable SSO, then users refreshing the report will use the credentials of the user who signed in, like most other Power BI reports.

Troubleshooting

If you run into any issues with the integration, see the Snowflake [troubleshooting guide](#).

Next steps

- [Data sources for the Power BI service](#)
- [Connect to datasets in the Power BI service from Power BI desktop](#)
- [Connect to Snowflake in Power BI Desktop](#)

Create visuals and reports with the Azure Cost Management connector in Power BI Desktop

Article • 08/10/2023

You can use the Azure Cost Management connector for Power BI Desktop to make powerful, customized visualizations and reports that help you better understand your Azure spend.

The Azure Cost Management connector currently supports customers with:

- A direct [Microsoft Customer Agreement](#)
- An Enterprise Agreement (EA)
- A [Microsoft Partner Agreement](#)

If you have an unsupported agreement, you can use Exports to save the cost data to a share and then connect to it using Power BI. For more information, see [Tutorial - Create and manage exported data from Azure Cost Management](#).

The Azure Cost Management connector uses OAuth 2.0 for authentication with Azure and identifies users who are going to use the connector. Tokens generated in this process are valid for a specific period. Power BI preserves the token for the next login. OAuth 2.0, is a standard for the process that goes on behind the scenes to ensure the secure handling of these permissions. To connect, you must use an [Enterprise Administrator](#) account for Enterprise Agreements, or have [appropriate permissions](#) at the billing account or billing profile levels for Microsoft Customer Agreements.

ⓘ Note

This connector replaces the previously available [Azure Consumption Insights](#) and [Azure Cost Management \(Beta\)](#) connectors. Any reports created with the previous connector must be recreated using this connector.

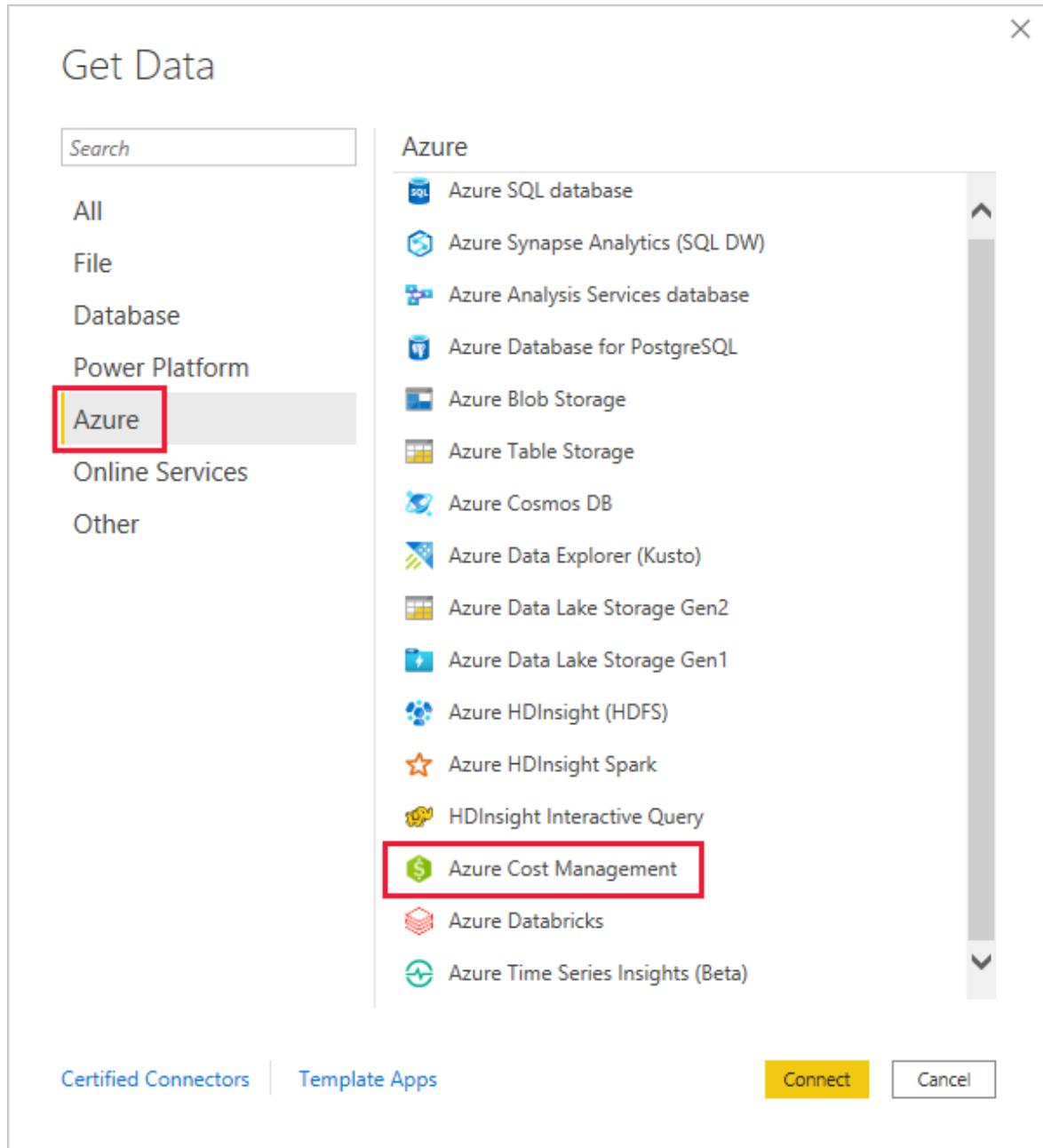
Connect using Azure Cost Management

To use the Azure Cost Management connector in Power BI Desktop, take the following steps:

1. In the Home ribbon, select **Get Data**.

2. Select Azure from the list of data categories.

3. Select Azure Cost Management.



4. In the dialog that appears, for the **Choose Scope** drop down, use **Manually Input Scope for Microsoft Customer Agreements**, or use **Enrollment Number** for Enterprise Agreements (EA).

Connect to a Microsoft Customer Agreement account

This section describes the steps necessary to connect to a Microsoft Customer Agreement account.

Connect to a billing account

To connect to a billing account, you need to retrieve your **Billing account ID** from the Azure portal:

1. In the [Azure portal](#), navigate to **Cost Management + Billing**.
2. Select your Billing profile.
3. Under **Settings** in the menu, select **Properties** in the sidebar.
4. Under **Billing profile**, copy the ID.

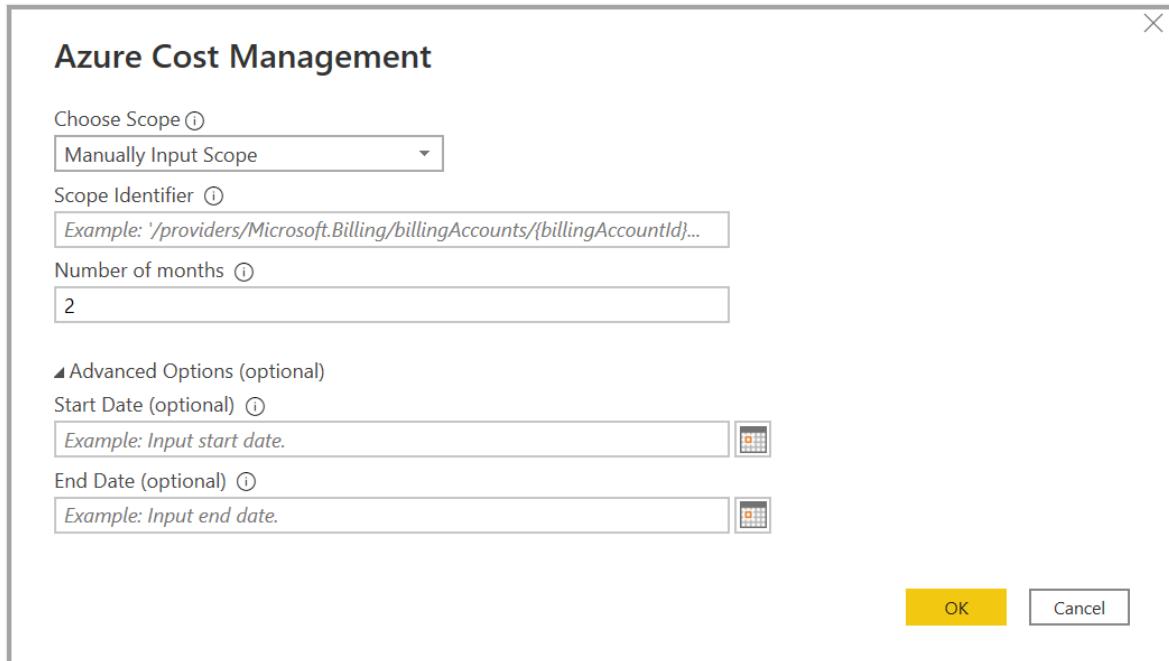
The screenshot shows the Azure portal interface for managing a billing account. The top navigation bar includes 'Home', 'Cost Management + Billing', and 'Test billing acct'. The main title is 'Test billing acct | Properties'. The left sidebar has sections for 'Overview', 'Access control (IAM)', 'Billing scopes', 'Cost management' (with sub-options for 'Cost analysis', 'Cost alerts', 'Budgets', and 'Advisor recommendations'), 'Billing' (with sub-options for 'Invoices', 'Payment methods', 'Reservation transactions', and 'Billing profiles'), 'Products + services' (with sub-options for 'Azure subscriptions' and 'Recurring charges'), 'Settings' (with sub-options for 'Properties' - which is highlighted with a grey background, 'Exports', and 'Cost allocation (preview)'), and 'Support + troubleshooting' (with sub-option for 'New support request'). The right pane displays the 'General' tab under 'Properties'. It shows the 'Name' as 'TestUser', 'Type' as 'Microsoft Customer Agreement', and 'Sold-to' information (Microsoft Redmond wa 98052 US). There is a red box highlighting the 'Copy' icon next to the 'Name' field. Below the 'General' tab, there are 'Agreements' and a '... more' button.

5. For **Choose Scope**, select **Manually Input Scope** and input the connection string as shown in the example below, replacing `{billingAccountId}` with the data copied from the previous steps.

```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}
```

Alternatively, for **Choose Scope**, select **Enrollment Number** and input the Billing Account ID string as copied from the previous steps.

6. Enter the number of months and select **OK**.



The screenshot shows the 'Demo Billing Profile | Properties' page in the Azure portal. On the left, there's a navigation menu with sections like Overview, Access control (IAM), Billing scopes, Cost management, Cost analysis, Cost alerts, Budgets, Advisor recommendations, Billing, Invoices, Payment methods, Payment history, Reservation transactions, Invoice sections, Products + services, Azure subscriptions, Recurring charges, Settings, and Properties. The 'Properties' item is currently selected. The main content area is titled 'Billing profile' and contains fields for ID, Name (Demo Billing Profile), PO number (Not available), Billing address (Redmond, WA, 98052, US), Enabled Azure plans (Microsoft Azure Plan), My role (Billing profile reader), Tags (No tags added), and a Billing account section with an ID field and Type (Microsoft Customer Agreement). The 'Billing profile' and 'Billing account' sections are highlighted with red boxes.

6. For **Choose Scope**, select **Manually Input Scope** and input the connection string as shown in the example below, replacing `{billingAccountId}` and `{billingProfileId}` with the data copied from the previous steps.

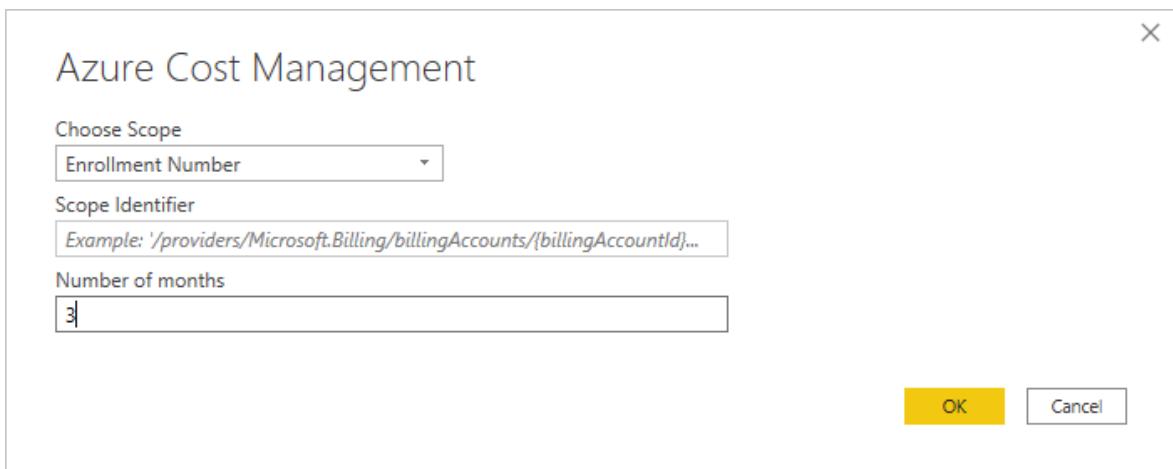
```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfile  
s/{billingProfileId}
```

7. Enter the number of months and select **OK**.
8. When prompted, sign in with your Azure user account and password. You must have access to the Billing profile to successfully access the billing profile data.

Connect to an Enterprise Agreement account

To connect with an Enterprise Agreement (EA) account, you can get your enrollment ID from the Azure portal:

1. In the [Azure portal](#), navigate to **Cost Management + Billing**.
2. Select your billing account.
3. On the **Overview** menu, copy the **Billing account ID**.
4. For **Choose Scope**, select **Enrollment Number** and paste the billing account ID from the previous step.
5. Enter the number of months and then select **OK**.



6. When prompted, sign in with your Azure user account and password. You must use an Enterprise Administrator account for Enterprise Agreements.

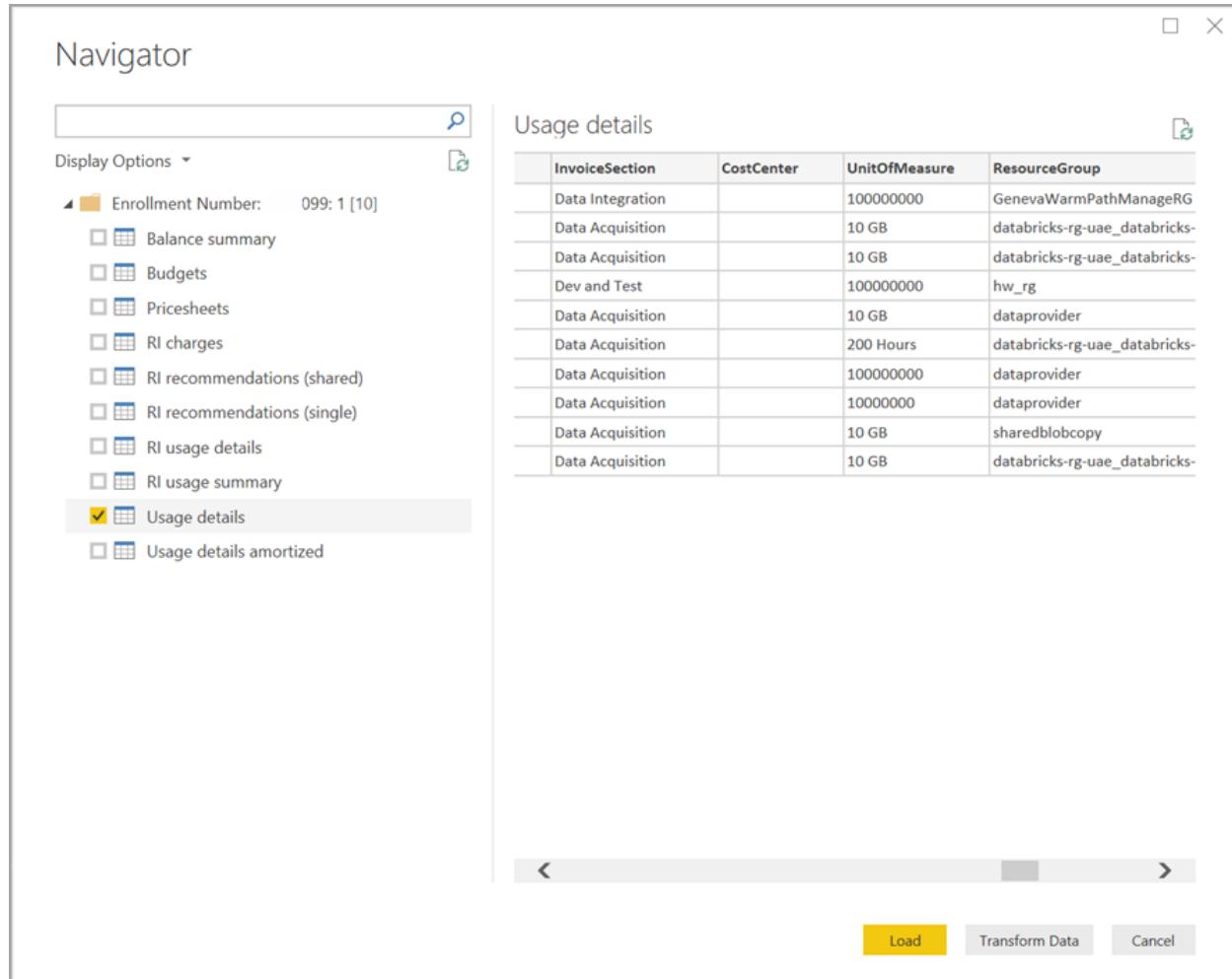
Data available through the connector

Once you successfully authenticate, a **Navigator** window appears with the following available data tables:

Table	Account Type	Supported Scopes	Description
Balance summary	EA only	EA Enrollment	Summary of the balance for the current billing month for Enterprise Agreements (EA).
Billing events	MCA only	Billing Profile	Event log of new invoices, credit purchases, etc. Microsoft Customer Agreement only.
Budgets	EA, MCA	EA Enrollment,MCA Billing Account,MCA	Budget details to view actual costs or usage against existing budget targets.

Table	Account Type	Supported Scopes	Description
		Billing Profile	
Charges	MCA only	MCA Billing Profile	A month-level summary of Azure usage, Marketplace charges, and charges billed separately. Microsoft Customer Agreement only.
Credit lots	MCA only	MCA Billing Profile	Azure credit lot purchase details for the provided billing profile. Microsoft Customer Agreement only.
Pricesheets	EA, MCA	EA Enrollment,MCA Billing Profile	Applicable meter rates for the provided billing profile or EA enrollment.
RI charges	EA, MCA	EA Enrollment,MCA Billing Profile	Charges associated to your Reserved Instances over the last 24 months. This table is in the process of being deprecated, please use RI transactions
RI recommendations (shared)	EA, MCA	EA Enrollment,MCA Billing Profile	Reserved Instance purchase recommendations based on all your subscription usage trends for the last 30 days.
RI recommendations (single)	EA, MCA	EA Enrollment,MCA Billing Profile	Reserved Instance purchase recommendations based on your single subscription usage trends for the last 30 days.
RI transactions	EA, MCA	EA Enrollment,MCA Billing Profile	List of transactions for reserved instances on billing account scope.
RI usage details	EA, MCA	EA Enrollment,MCA Billing Profile	Consumption details for your existing Reserved Instances over the last month.
RI usage summary	EA, MCA	EA Enrollment,MCA Billing Profile	Daily Azure reservation usage percentage.
Usage details	EA, MCA	EA Enrollment,MCA Billing Account,MCA Billing Profile	A breakdown of consumed quantities and estimated charges for the given billing profile on EA enrollment.
Usage details amortized	EA, MCA	EA Enrollment,MCA Billing Account,MCA Billing Profile	A breakdown of consumed quantities and estimated amortized charges for the given billing profile on EA enrollment.

You can select a table to see a preview dialog. You can select one or more tables by selecting the boxes beside their name and then select **Load**.



When you select **Load**, the data is loaded into Power BI Desktop.

When the data you selected is loaded, the data tables and fields are shown in the **Fields** pane.

Considerations and limitations

The following considerations and limitations apply to the Azure Cost Management data connector:

- Data row requests exceeding one million rows isn't supported by Power BI. Instead, you can try using the export feature described in [create and manage exported data in Azure Cost Management](#).
- The Azure Cost Management data connector doesn't work with Office 365 GCC customer accounts.
- **Data refresh:** The cost and usage data is typically updated and available in the Azure portal and supporting APIs within 8 to 24 hours, so we suggest you

constrain Power BI scheduled refreshes to once or twice a day.

- **Data source reuse:** If you have multiple reports that are pulling the same data, and don't need additional report-specific data transformations, you should reuse the same data source, which would reduce the amount of time required to pull the Usage Details data.

For more information on reusing data sources, see the following:

- [Introduction to datasets across workspaces](#)
- [Create reports based on datasets from different workspaces](#)

You might receive a *400 bad request* from the **RI usage details** when you try to refresh the data if you've chosen date parameter greater than three months. To mitigate the error, take the following steps:

1. In Power BI Desktop, select **Home > Transform data**.
2. In Power Query Editor, select the **RI usage details** dataset and select **Advanced Editor**.
3. Update the Power Query code as shown in the following paragraph(s), which will split the calls into three-month chunks. Make sure you note and retain your enrollment number, or billing account/billing profile ID.

For EA use the following code update:

```
let
    enrollmentNumber = "<<Enrollment Number>>",
    optionalParameters1 = [startBillingDataWindow = "-9",
    endBillingDataWindow = "-6"],
    source1 = AzureCostManagement.Tables("Enrollment Number",
enrollmentNumber, 5, optionalParameters1),
    riusagedetails1 = source1{[Key="riusagedetails"]}[Data],
    optionalParameters2 = [startBillingDataWindow = "-6",
endBillingDataWindow = "-3"],
    source2 = AzureCostManagement.Tables("Enrollment Number",
enrollmentNumber, 5, optionalParameters2),
    riusagedetails2 = source2{[Key="riusagedetails"]}[Data],
    riusagedetails = Table.Combine({riusagedetails1, riusagedetails2})
in
    riusagedetails
```

For Microsoft Customer Agreements use the following update:

```

let
    billingProfileId = "<>Billing Profile Id>>",
    optionalParameters1 = [startBillingDataWindow = "-9",
    endBillingDataWindow = "-6"],
    source1 = AzureCostManagement.Tables("Billing Profile Id",
    billingProfileId, 5, optionalParameters1),
    riusagedetails1 = source1{[Key="riusagedetails"]}[Data],
    optionalParameters2 = [startBillingDataWindow = "-6",
    endBillingDataWindow = "-3"],
    source2 = AzureCostManagement.Tables("Billing Profile Id",
    billingProfileId, 5, optionalParameters2),
    riusagedetails2 = source2{[Key="riusagedetails"]}[Data],
    riusagedetails = Table.Combine({riusagedetails1, riusagedetails2})
in
    riusagedetails

```

4. Once you've updated the code with the appropriate update from the previous step, select **Done** and then select **Close & Apply**.

You might run into a situation where tags aren't working in the usage details or the tags column can't be transformed to json. This issue stems from the current UCDD api returning the tags column by trimming the start and end brackets, which results in Power BI being unable to transform the column because it returns it as a string. To mitigate this situation, take the following steps.

1. Navigate to **Query Editor**.
2. Select the *Usage Details* table.
3. In the right pane, the **Properties** pane shows the **Applied Steps**. You need to add a custom column to the steps, after the **Navigation** step.
4. From the menu, select **Add column > Add custom column**
5. Name the column, for example you could name the column *TagsInJson* or whatever you prefer, and then enter the following text in the query:

DAX

```
```= {"& [Tags] & "}"
```

6. Completing the previous steps creates a new column of *tags* in the json format
7. You can now transfer and expand the column as you need to.

**Authentication issues encountered with Azure Active Directory guest accounts:** You may have the appropriate permissions to access the enrollment or billing account, but receive an authentication error similar to one of the following:

- *Access to the resource is forbidden*
- *We couldn't authenticate with the credentials provided. Please try again.*

These errors could be the result of having a user account in a different Azure Active Directory domain that has been added as a guest user.

For guest accounts: Use the following settings or options as you are prompted with the **authentication dialog** when connecting with the Cost Management Power BI connector:

1. Select **Sign-in**
2. Select the **Use another account** (bottom of the dialog)
3. Select **Sign-in options** (bottom of the dialog box)
4. Select **Sign into an organization**
5. For **Domain name**, provide the Fully Qualified Domain Name (FQDN) of the Azure Active Directory domain into which you've been added as a guest.
6. Then, for **Pick an account** select the user account that you've previously authenticated.

## Next steps

You can connect to many different data sources using Power BI Desktop. For more information, see the following articles:

- [What is Power BI Desktop?](#)
- [Data Sources in Power BI Desktop](#)
- [Shape and Combine Data with Power BI Desktop](#)
- [Connect to Excel workbooks in Power BI Desktop](#)
- [Enter data directly into Power BI Desktop](#)

# Edit SAP variables in Power BI

Article • 03/20/2023

Report authors who use SAP Business Warehouse or SAP HANA with DirectQuery can allow end users to edit SAP variables in Power BI Premium and shared workspaces. This article describes the requirements for editing SAP variables, how to enable this feature, and how to edit variables in Power BI Desktop and the Power BI service.

## Requirements and limitations

The following lists describe the requirements and limitations for editing SAP variables:

### Requirements

- **DirectQuery connection.** You must connect to the SAP data source by using DirectQuery. Import connections aren't supported.
- **Single sign-on (SSO) set up.** You must configure SSO for your gateway for this feature to work. For more information, see [Overview of single sign-on for on-premises data gateways in Power BI](#).
- **Latest gateway version.** Make sure to download the latest gateway or update your existing gateway. For more information, see [What is an on-premises data gateway?](#)

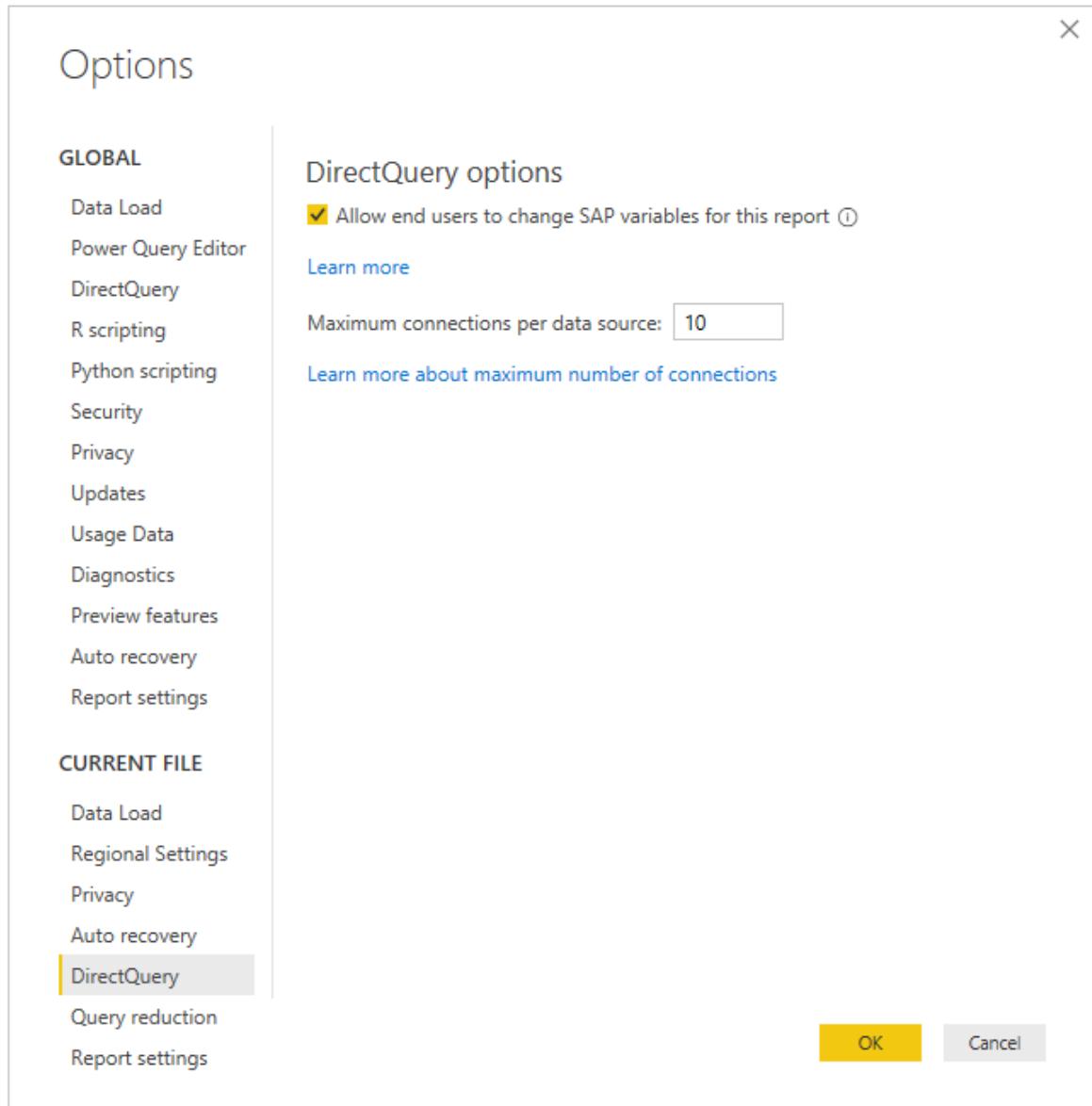
### Limitations

- **Multidimensional models only for SAP HANA.** For SAP HANA, the SAP edit variables feature works only with multidimensional models and doesn't work on relational sources.
- **No sovereign cloud support.** Power Query Online isn't available in sovereign clouds, so sovereign clouds don't support the edit SAP variables feature.
- **No mobile support.** You can't edit SAP variables in Power BI mobile apps.
- **Workspace restrictions.** Editing SAP variables doesn't work for reports in the **Shared with me** tab of **My Workspace**, or in apps created from V1 workspaces.

## Enable editing SAP variables

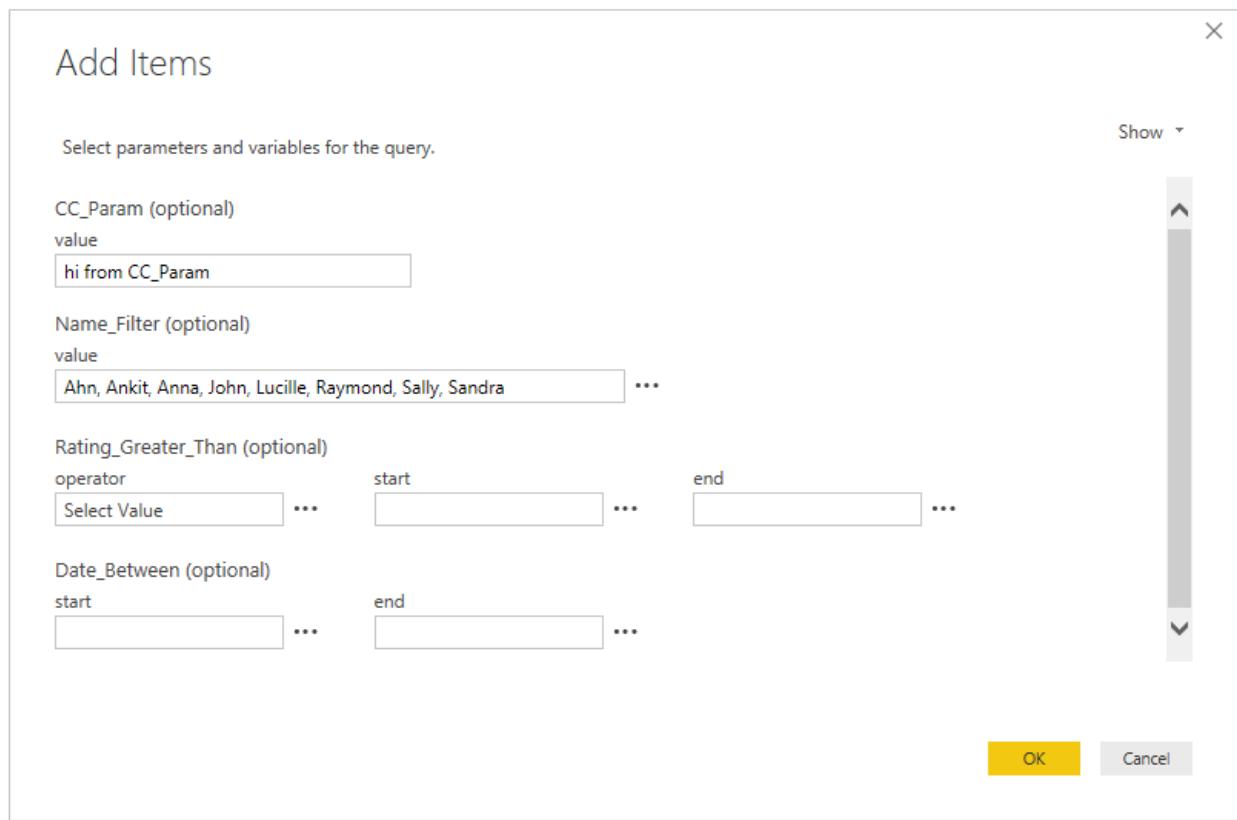
To enable report users to edit SAP variables:

1. In Power BI Desktop, connect to an SAP HANA or SAP BW data source with a DirectQuery connection.
2. Go to **File > Options and settings > Options**, and in the left pane, select **DirectQuery** under **Current File**.
3. Under **DirectQuery options** in the right pane, select the checkbox next to **Allow end users to change SAP variables for this report**.



## Edit SAP variables

In Power BI Desktop, you can edit variables by selecting **Transform data > Edit variables** in the ribbon. Report creators can add and select variables for the report by using the following dialog box:



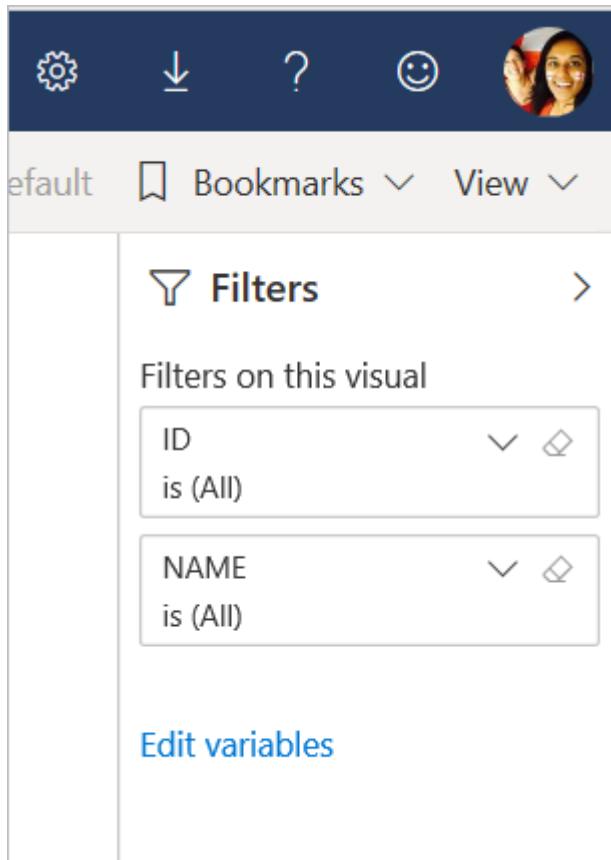
After you publish a report that enables editing SAP variables, the **Edit variables** link appears in the **Filter** pane for the report in the Power BI service. The first time you publish the report, it might take up to five minutes before the **Edit variables** link appears.

**① Note**

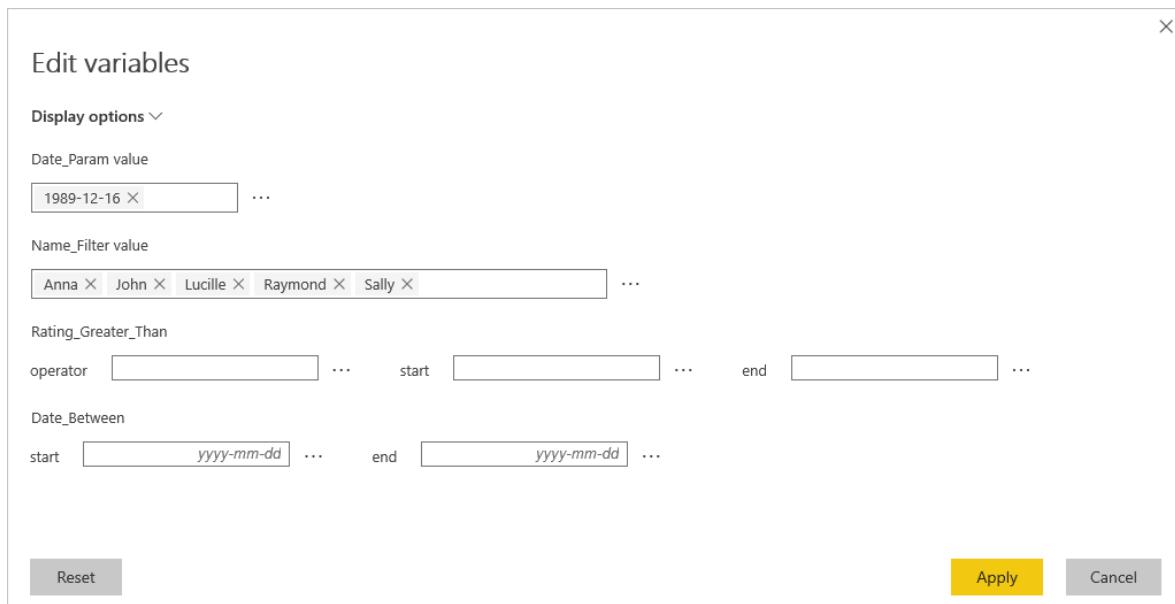
If the link doesn't appear, manually refresh the dataset by selecting it from the list in the **Datasets** tab of the workspace, and then selecting the **Refresh** icon.

To edit the variables in the Power BI service, report users can:

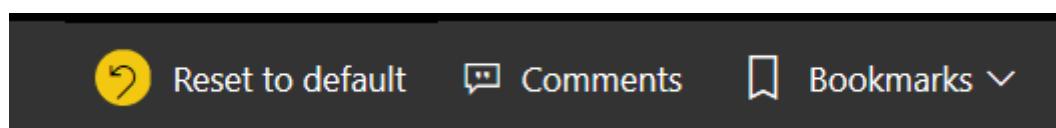
1. Select **Edit variables** in the **Filter** pane for the report.



2. In the **Edit variables** dialog box, edit and override the variable values, or select the **Reset** button to revert their changes.



Similar to other Power BI persistence behaviors, any changes users make in the **Edit variables** dialog box persist only for that user. Selecting **Reset to default** in the top menu bar resets the report to its original state, including the variables.



You can change the default variables for reports you own in the Power BI service. If you own a report that uses SAP HANA or SAP BW and enables editing variables, select **Edit variables** to change the variables. When you save the report, the changed variables become the new default settings for that report. Other users who access the report after you make the changes see the new settings as the defaults.

## Troubleshooting

If you get errors that Power BI can't load data or retrieve data for a visual, or that the data source connection failed, try the following actions to resolve the error:

- In the Power BI service, select **Edit variables**, set default values for the variables, and then save the report.
- In Power BI Desktop, if you no longer want users to be able to edit variables, uncheck the option at the report level.

## Next steps

- [Use SAP HANA in Power BI Desktop](#)
- [DirectQuery and SAP Business Warehouse \(BW\)](#)
- [DirectQuery and SAP HANA](#)
- [Use DirectQuery in Power BI](#)

# Connect to Analysis Services tabular data in Power BI Desktop

Article • 01/12/2023

With Power BI Desktop, there are two ways you can connect to and get data from your SQL Server Analysis Services tabular models:

- Explore by using a live connection
- Select items and import them into Power BI Desktop

**Explore by using a live connection:** When you use a live connection, items in your tabular model or perspective, like tables, columns, and measures, appear in your Power BI Desktop **Fields** pane list. You can use Power BI Desktop's advanced visualization and report tools to explore your tabular model in new, highly interactive ways.

When you connect live, no data from the tabular model is imported into Power BI Desktop. Each time you interact with a visualization, Power BI Desktop queries the tabular model and calculates the results that you see. You're always looking at the latest data that is available in the tabular model, either from the last processing time, or from DirectQuery tables available in the tabular model.

Keep in mind that tabular models are highly secure. Items that appear in Power BI Desktop depend on your permissions for the tabular model that you're connected to.

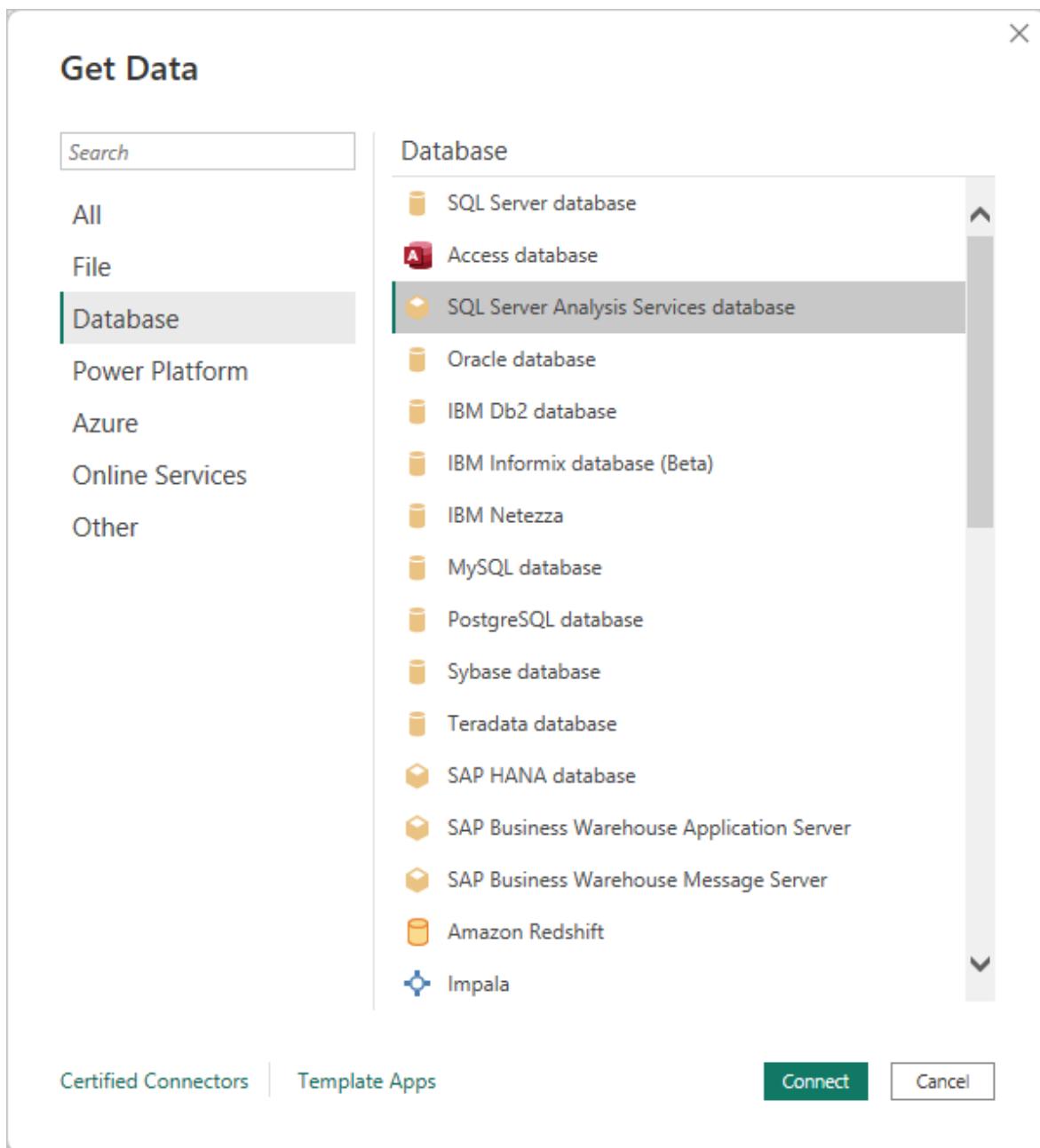
When you've created dynamic reports in Power BI Desktop, you can share them by publishing to your Power BI site. When you publish a Power BI Desktop file with a live connection to a tabular model to your Power BI site, an on-premises data gateway must be installed and configured by an administrator. for more information, see [On-premises data gateway](#).

**Select items and import into Power BI Desktop:** When you connect with this option, you can select items like tables, columns, and measures in your tabular model or perspective and load them into a Power BI Desktop model. Use Power BI Desktop's Power Query Editor to further shape what you want and its modeling features to further model the data. Because no live connection between Power BI Desktop and the tabular model is maintained, you can then explore your Power BI Desktop model offline or publish to your Power BI site.

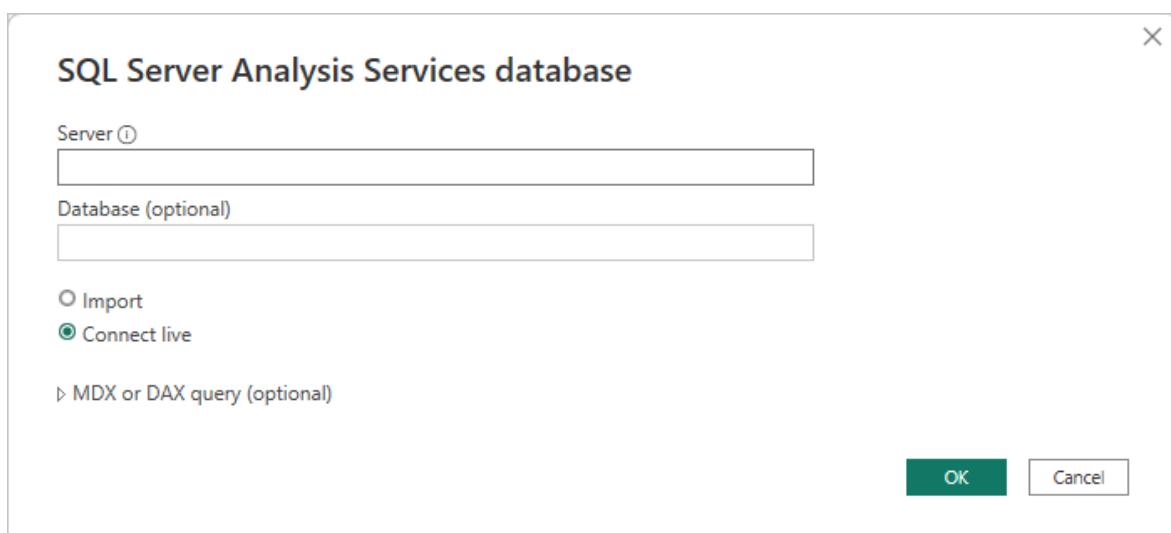
## To connect to a tabular model

1. In Power BI Desktop, on the **Home** ribbon, select **Get Data > More > Database**.

2. Select **SQL Server Analysis Services database**, and then select **Connect**.

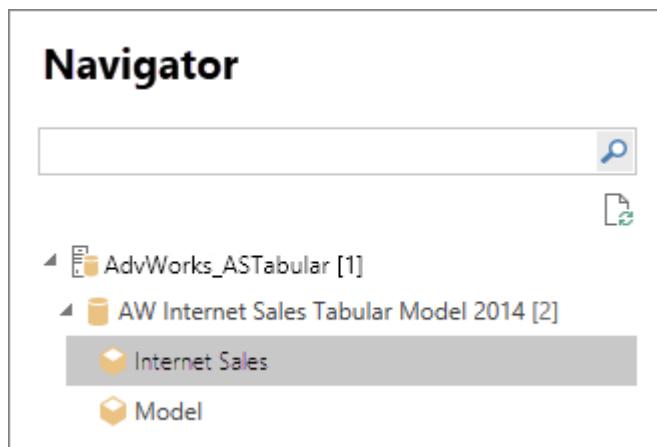


3. In the **SQL Server Analysis Services database** window, enter the **Server** name, choose a connection mode, and then select **OK**.

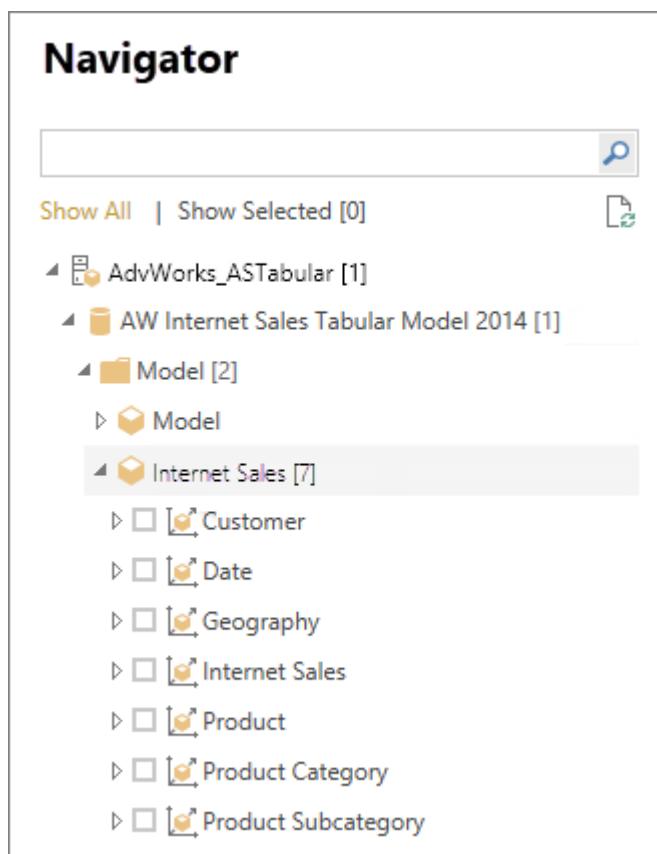


4. This step in the **Navigator** window depends on the connection mode you selected:

- If you're connecting live, select a tabular model or perspective.



- If you chose to select items and get data, select a tabular model or perspective, and then select a particular table or column to load. To shape your data before loading, select **Transform data** to open Power Query Editor. When you're ready, select **Load** to import the data into Power BI Desktop.



## Frequently asked questions

**Question:** Do I need an on-premises data gateway?

**Answer:** It depends. If you use Power BI Desktop to connect live to a tabular model, but have no intention to publish to your Power BI site, you don't need a gateway. On the other hand, if you do intend on publishing to your Power BI site, a data gateway is necessary to ensure secure communication between the Power BI service and your on-premises Analysis Services server. Be sure to talk to your Analysis Services server administrator before installing a data gateway.

If you choose to select items and get data, you import tabular model data directly into your Power BI Desktop file, so no gateway is necessary.

**Question:** What's the difference between connecting live to a tabular model from the Power BI service versus connecting live from Power BI Desktop?

**Answer:** When you connect live to a tabular model from your site in the Power BI service to an Analysis Services database on-premises in your organization, an on-premises data gateway is required to secure communications between them. When you connect live to a tabular model from Power BI Desktop, a gateway isn't required because the Power BI Desktop and the Analysis Services server you're connecting to are both running on-premises in your organization. However, if you publish your Power BI Desktop file to your Power BI site, a gateway is required.

**Question:** If I created a live connection, can I connect to another data source in the same Power BI Desktop file?

**Answer:** No. You can't explore live data and connect to another type of data source in the same file. If you've already imported data or connected to a different data source in a Power BI Desktop file, you need to create a new file to explore live.

**Question:** If I created a live connection, can I edit the model or query in Power BI Desktop?

**Answer:** You can create report level measures in the Power BI Desktop, but all other query and modeling features are disabled when exploring live data.

**Question:** If I created a live connection, is it secure?

**Answer:** Yes. Your current Windows credentials are used to connect to the Analysis Services server. You can't use basic or stored credentials in either the Power BI service or Power BI Desktop when exploring live.

**Question:** In Navigator, I see a model and a perspective. What's the difference?

**Answer:** A perspective is a particular view of a tabular model. It might include only particular tables, columns, or measures depending on a unique data analysis need. A tabular model always contains at least one perspective, which could include everything

in the model. If you're unsure which perspective you should select, check with your administrator.

**Question:** Are there any features of Analysis Services that change the way Power BI behaves?

**Answer:** Yes. Depending on the features your tabular model uses, the experience in Power BI Desktop might change. Some examples include:

- You may see measures in the model grouped together at the top of the **Fields** pane list rather than in tables alongside columns. Don't worry, you can still use them as normal, it's just easier to find them this way.
- If the tabular model has calculation groups defined, you can use them only with model measures and not with implicit measures you create by adding numeric fields to a visual. The model might also have had the **DiscourageImplicitMeasures** flag set manually, which has the same effect. For more information, see [Calculation groups in Analysis Services](#).

## To change the server name after initial connection

After you create a Power BI Desktop file with an explore live connection, there might be some cases where you want to switch the connection to a different server. For example, if you created your Power BI Desktop file when connecting to a development server, and before publishing to the Power BI service, you want to switch the connection to production server.

To change the server name:

1. Select **Transform data > Data source settings** from the **Home** tab.
2. In the **SQL Server Analysis Services database** window, enter the new **Server** name, and then select **OK**.

## Troubleshooting

The following list describes all known issues when connecting to SQL Server Analysis Services (SSAS) or Azure Analysis Services:

- **Error: Couldn't load model schema:** This error usually occurs when the user connecting to Analysis Services doesn't have access to the database/model.

# Use DirectQuery in Power BI Desktop

Article • 08/21/2023

When you connect to any data source with Power BI Desktop, you can import a copy of the data. For some data sources, you can also connect directly to the data source without importing data by using DirectQuery.

To determine whether a data source supports DirectQuery, view the full listing of available data sources found in the article [Connectors in Power Query](#) which also applies to Power BI, select the article that describes the data source you're interested in from the list of supported connectors, then see the section in that connector's article titled **Capabilities supported**. If DirectQuery isn't listed in that section for the data source's article, DirectQuery isn't supported for that data connector.

Here are the differences between using import and DirectQuery connectivity modes:

- **Import:** A copy of the data from the selected tables and columns imports into Power BI Desktop. As you create or interact with visualizations, Power BI Desktop uses the imported data. To see underlying data changes after the initial import or the most recent refresh, you must import the full dataset again to refresh the data.
- **DirectQuery:** No data imports into Power BI Desktop. For relational sources, you can select tables and columns to appear in the Power BI Desktop **Fields** list. For multidimensional sources like SAP Business Warehouse (SAP BW), the dimensions and measures of the selected cube appear in the **Fields** list. As you create or interact with visualizations, Power BI Desktop queries the underlying data source, so you're always viewing current data.

With DirectQuery, when you create or interact with a visualization, you must query the underlying source. The time that's needed to refresh the visualization depends on the performance of the underlying data source. If the data needed to service the request was recently requested, Power BI Desktop uses the recent data to reduce the time required to show the visualization. Selecting **Refresh** from the **Home** ribbon refreshes all visualizations with current data.

Many data modeling and data transformations are available when using DirectQuery, although with some performance-based limitations. For more information about DirectQuery benefits, limitations, and recommendations, see [DirectQuery in Power BI](#).

## DirectQuery benefits

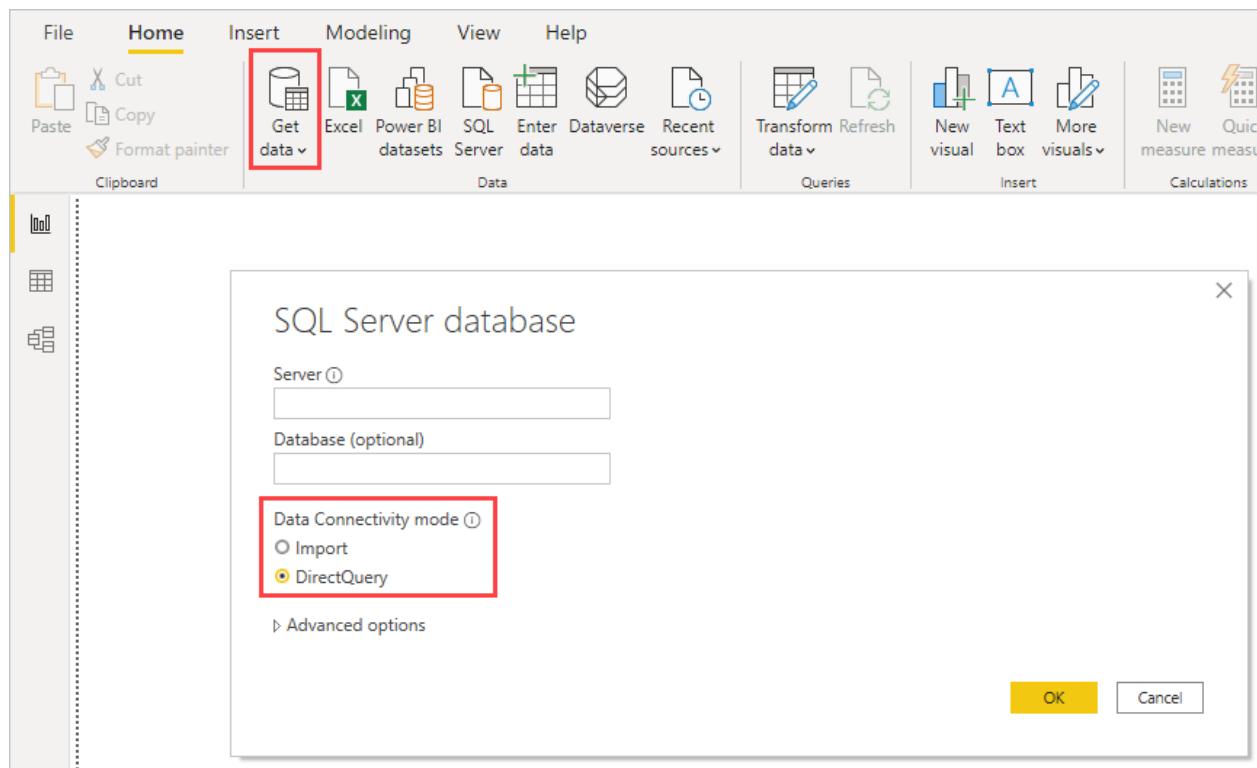
Some benefits of using DirectQuery include:

- DirectQuery lets you build visualizations over very large datasets, where it would be unfeasible to import all the data with pre-aggregation.
- DirectQuery reports always use current data. Seeing underlying data changes requires you to refresh the data, and reimporting large datasets to refresh data could be unfeasible.
- The 1-GB dataset limitation doesn't apply with DirectQuery.

## Connect using DirectQuery

To connect to a data source with DirectQuery:

1. In the **Home** group of the Power BI Desktop ribbon, select **Get data**, and then select a data source that DirectQuery supports, such as **SQL Server**.
2. In the dialog box for the connection, under **Data connectivity mode**, select **DirectQuery**.



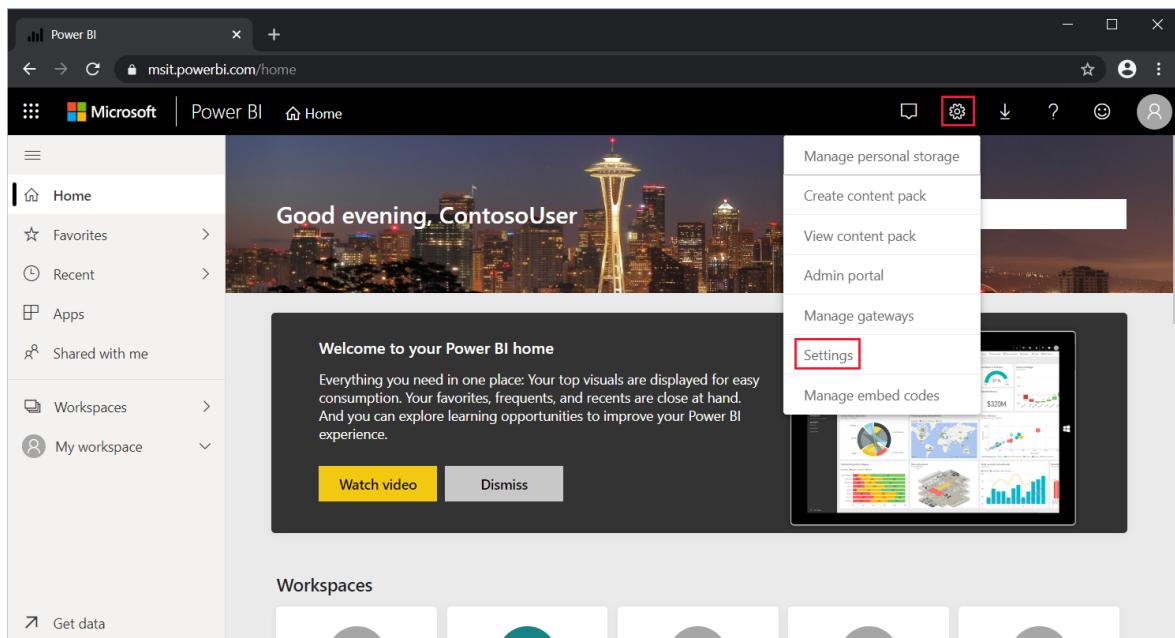
## Publish to the Power BI service

You can publish DirectQuery reports to the Power BI service, but you need to take extra steps for the Power BI service to open the reports.

- To connect the Power BI service to DirectQuery data sources other than Azure SQL Database, Azure Synapse Analytics (formerly SQL Data Warehouse), Amazon Redshift, and Snowflake Data Warehouse, [install an on-premises data gateway](#) and register the data source.
- If you used DirectQuery with cloud sources like Azure SQL Database, Azure Synapse, Amazon Redshift, or Snowflake Data Warehouse, you don't need an on-premises data gateway. You still must provide credentials for the Power BI service to open the published report. Without credentials, an error occurs when you try to open a published report or explore a dataset created with a DirectQuery connection.

To provide credentials for opening the report and refreshing the data:

1. In the Power BI service, select the gear icon at upper-right and choose **Settings**.



2. On the **Settings** page, select the **Datasets** tab, and choose the dataset that uses DirectQuery.
3. Under **Data source connection**, provide the credentials to connect to the data source.

#### ! Note

If you used DirectQuery with an Azure SQL Database that has a private IP address, you need to use an on-premises gateway.

## Considerations and limitations

Some Power BI Desktop features aren't supported in DirectQuery mode, or they have limitations. Some capabilities in the Power BI service, such as quick insights, also aren't available for datasets that use DirectQuery. When you decide whether to use DirectQuery, consider these feature limitations. Also consider the following factors:

## Performance and load considerations

DirectQuery sends all requests to the source database, so the required refresh time for visuals depends on how long the underlying source takes to return results. Five seconds or less is the recommended response time for receiving requested data for visuals. Refresh times over 30 seconds produce an unacceptably poor experience for users consuming the report. A query that takes longer than four minutes times out in the Power BI service, and the user receives an error.

Load on the source database also depends on the number of Power BI users who consume the published report, especially if the report uses row-level security (RLS). The refresh of a non-RLS dashboard tile shared by multiple users sends a single query to the database, but refreshing a dashboard tile that uses RLS requires one query per user. The increased queries significantly increase load and potentially affect performance.

## One-million row limit

DirectQuery defines a one-million row limit for data returned from cloud data sources, which are any data sources that aren't on-premises. On-premises sources are limited to a defined payload of about 4 MB per row, depending on proprietary compression algorithm, or 16 MB for the entire visual. Premium capacities can set different maximum row limits, as described in the blog post [Power BI Premium new capacity settings](#).

Power BI creates queries that are as efficient as possible, but some generated queries might retrieve too many rows from the underlying data source. For example, this situation can occur with a simple chart that includes a high cardinality column with the aggregation option set to **Don't Summarize**. The visual must have only columns with a cardinality below 1 million, or must apply the appropriate filters.

The row limit doesn't apply to aggregations or calculations used to select the dataset DirectQuery returns, only to the rows returned. For example, the query that runs on the data source can aggregate 10 million rows. As long as the data returned to Power BI is less than 1 million rows, the query can accurately return the results. If the data is over 1 million rows, Power BI shows an error, except in Premium capacity with different admin-set limits. The error states: **The resultset of a query to external data source has exceeded the maximum allowed size of '1000000' rows.**

## Security considerations

By default, all users who consume a published report in the Power BI service connect to the underlying data source by using the credentials entered after publication. This situation is the same as for imported data. All users see the same data, regardless of any security rules that the underlying source defines.

If you need per-user security implemented with DirectQuery sources, either use RLS or configure Kerberos-constrained authentication against the source. Kerberos isn't available for all sources. For more information, see [Row-level security \(RLS\) with Power BI](#) and [Configure Kerberos-based SSO from Power BI service to on-premises data sources](#).

## Other DirectQuery limitations

Some other limitations of using DirectQuery include:

- If the Power Query Editor query is overly complex, an error occurs. To fix the error, you must either delete the problematic step in Power Query Editor, or switch to import mode. Multidimensional sources like SAP BW can't use the Power Query Editor.
- Automatic date/time hierarchy is unavailable in DirectQuery. DirectQuery mode doesn't support date column drilldown by year, quarter, month, or day.
- For table or matrix visualizations, there's a 125-column limit for results that return more than 500 rows from DirectQuery sources. These results display a scroll bar in the table or matrix that lets you fetch more data. In that situation, the maximum number of columns in the table or matrix is 125. If you must include more than 125 columns in a single table or matrix, consider creating measures that use `MIN`, `MAX`, `FIRST`, or `LAST`, because they don't count against this maximum.
- You can't change from import to DirectQuery mode. You can switch from DirectQuery mode to import mode if you import all the necessary data. It's not possible to switch back, mostly because of the feature set that DirectQuery doesn't support. DirectQuery models over multidimensional sources, like SAP BW, can't be switched from DirectQuery to import mode either, because of the different treatment of external measures.
- Calculated tables and calculated columns that reference a DirectQuery table from a data source with single sign-on (SSO) authentication aren't supported in the Power BI service.

# Next steps

- DirectQuery in Power BI
- Data sources supported by DirectQuery
- DirectQuery and SAP Business Warehouse (BW)
- DirectQuery and SAP HANA
- What is an on-premises data gateway?
- Using DirectQuery for Power BI datasets and Azure Analysis Services (preview)

# Connect to SAP Business Warehouse by using DirectQuery in Power BI

Article • 03/09/2023

You can connect to *SAP Business Warehouse* (SAP BW) data sources directly using *DirectQuery*. Given the OLAP/multidimensional nature of SAP BW, there are many important differences between DirectQuery over SAP BW versus relational sources like SQL Server. These differences are summarized as follows:

- In DirectQuery over relational sources, there's a set of queries, as defined in the **Get Data or Power Query Editor** dialog, that logically defines the data that is available in the field list. This configuration is *not* the case when connecting to an OLAP source such as SAP BW. Instead, when connecting to the SAP server using **Get Data**, just the InfoCube or BEx Query is selected. Then all the Key Figures and dimensions of the selected InfoCube/BEx Query are available in the field list.
- Similarly, there's no **Power Query Editor** when connecting to SAP BW. The data source settings, for example, server name, can be changed by selecting **Transform data > Data source settings**. The settings for any parameters can be changed by selecting **Transform data > Edit parameters**.
- Given the unique nature of OLAP sources, there are other restrictions for both modeling and visualizations that apply, in addition to the normal restrictions imposed for DirectQuery. These restrictions are described later in this article.

In addition, it's *extremely important* to understand that there are many features of SAP BW that aren't supported in Power BI, and that because of the nature of the public interface to SAP BW, there are important cases where the results seen through Power BI don't match the ones seen when using an SAP tool. These limitations are described later in this article. These limitations and behavior differences should be carefully reviewed to ensure that the results seen through Power BI, as returned by the SAP public interface, are interpreted correctly.

## ⓘ Note

The ability to use DirectQuery over SAP BW was in preview until the March 2018 update to Power BI Desktop. During the preview, feedback and suggested improvements prompted a change that impacts reports that were created using that preview version. Now that General Availability (GA) of DirectQuery over SAP BW has released, you *must* discard any existing (preview-based) reports using DirectQuery over SAP BW that were created with the pre-GA version.

In reports created with the pre-GA version of DirectQuery over SAP BW, errors occur with those pre-GA reports upon invoking Refresh, as a result of attempting to refresh the metadata with any changes to the underlying SAP BW cube. Please re-create those reports from a blank report, using the GA version of DirectQuery over SAP BW.

## Additional modeling restrictions

The other primary modeling restrictions when connecting to SAP BW using DirectQuery in Power BI are:

- **No support for calculated columns:** The ability to create calculated columns is disabled. This fact also means that grouping and clustering, which create calculated columns, aren't available.
- **Additional limitations for measures:** There are other limitations imposed on the DAX expressions that can be used in measures to reflect the level of support offered by SAP BW.
- **No support for defining relationships:** The relationships are inherent in the external SAP source. Other relationships can't be defined in the model.
- **No Data View:** The data view normally displays the detail level data in the tables. Given the nature of OLAP sources like SAP BW, this view isn't available over SAP BW.
- **Column and measure details are fixed:** The list of columns and measures seen in the field list are fixed by the underlying source, and can't be modified. For example, it's not possible to delete a column or change its datatype. It can, however, be renamed.
- **Additional limitations in DAX:** There are more limitations on the DAX that can be used in measure definitions to reflect limitations in the source. For example, it's not possible to use an aggregate function over a table.

## Additional visualization restrictions

The other primary restrictions in visualizations when connecting to SAP BW using DirectQuery in Power BI are:

- **No aggregation of columns:** It's not possible to change the aggregation for a column on a visual. It's always *Do Not Summarize*
- **Measure filtering is disabled:** Measure filtering is disabled to reflect the support offered by SAP BW.

- Multi-select and include/exclude:** The ability to multi-select data points on a visual is disabled if the points represent values from more than one column. For example, given a bar chart showing Sales by Country/Region, with Category on the Legend, it wouldn't be possible to select the point for (USA, Bikes) and (France, Clothes). Similarly, it wouldn't be possible to select the point for (USA, Bikes) and exclude it from the visual. Both limitations are imposed to reflect the support offered by SAP BW.

## Support for SAP BW features

The following table lists all SAP BW features that aren't fully supported, or behave differently when using Power BI.

Feature	Description
Local calculations	<p>Local calculations defined in a BEx Query change the numbers as displayed through tools like BEx Analyzer. However, they aren't reflected in the numbers returned from SAP, through the public MDX interface.</p> <p><b>As such, the numbers seen in a Power BI visual don't necessarily match those for a corresponding visual in an SAP tool.</b></p>
	<p>For example, when connecting to a query cube from a BEx query that sets the aggregation to be <i>Cumulated</i>, or running sum, Power BI would get back the base numbers, ignoring that setting. An analyst could certainly then apply a running sum calculation locally in Power BI, but would need to exercise caution in how the numbers are interpreted if this action isn't done.</p>
Aggregations	<p>In some cases, particularly when dealing with multiple currencies, the aggregate numbers returned by the SAP public interface don't match the results shown by SAP tools.</p> <p><b>As such, the numbers seen in a Power BI visual don't necessarily match those for a corresponding visual in an SAP tool.</b></p>
	<p>For example, totals over different currencies would show as "*" in BEx Analyzer, but the total would get returned by the SAP public interface, without any information that such an aggregate number is meaningless. Thus the number aggregating, say, \$, EUR, and AUD, would get displayed by Power BI.</p>
Currency formatting	Any currency formatting, for example, \$2,300 or 4000 AUD, isn't reflected in Power BI.
Units of measure	Units of measure, for example, 230 KG, aren't reflected in Power BI.

Feature	Description
<b>Key versus text</b> (short, medium, long)	For an SAP BW characteristic like <code>CostCenter</code> , the field list shows a single column <i>Cost Center</i> . Using that column displays the default text. By showing hidden fields, it's also possible to see the unique name column that returns the unique name assigned by SAP BW, and is the basis of uniqueness.
	The key and other text fields aren't available.
<b>Multiple hierarchies of a characteristic</b>	In SAP, a characteristic can have multiple hierarchies. Then in tools like BEx Analyzer, when a characteristic is included in a query, the user can select the hierarchy to use.
	In Power BI, the various hierarchies can be seen in the field list as different hierarchies on the same dimension. However, selecting multiple levels from two different hierarchies on the same dimension results in empty data being returned by SAP.
<b>Treatment of ragged hierarchies</b>	<p>SAP BW supports ragged hierarchies, where levels can be missed e.g.</p> <ul style="list-style-type: none"> <li>Continent</li> <li>Americas</li> <li>    Canada</li> <li>    USA</li> <li>Not Assigned</li> <li>Australia</li> </ul>
	In Power BI, this appears with (Blank) at the missing level
	<ul style="list-style-type: none"> <li>Continent</li> <li>Americas</li> <li>    Canada</li> <li>    USA</li> <li>Not Assigned</li> <li>(Blank)</li> <li>Australia</li> </ul>
<b>Scaling factor/reverse sign</b>	In SAP, a key figure can have a scaling factor, for example, <i>1000</i> , defined as a formatting option, meaning that all display is scaled by that factor.
	It can similarly have a property set that reverses the sign. Use of such a key figure in Power BI in a visual, or as part of a calculation results in the unscaled number being used. The sign isn't reversed. The underlying scaling factor isn't available. In Power BI visuals, the scale units shown on the axis (K,M,B) can be controlled as part of the visual formatting.
<b>Hierarchies where levels appear/disappear dynamically</b>	Initially when connecting to SAP BW, the information on the levels of a hierarchy are retrieved, resulting in a set of fields in the field list. This information is cached, and if the set of levels changes, then the set of fields don't change until <i>Refresh</i> is invoked.

Feature	Description
	This situation is only possible in <b>Power BI Desktop</b> . Such refresh to reflect changes to the levels can't be invoked in the Power BI service after publish.
<b>Default filter</b>	A BEx query can include default filters, which are applied automatically by SAP BEx Analyzer. These filters aren't exposed, and hence the equivalent usage in Power BI doesn't apply the same filters by default.
<b>Hidden Key figures</b>	A BEx query can control visibility of key figures, and those key figures that are hidden don't appear in SAP BEx Analyzer. This fact isn't reflected through the public API, and hence such hidden key figures still appear in the field list. However, they can then be hidden within Power BI.
<b>Numeric formatting</b>	Any numeric formatting, such as number of decimal positions and decimal point, isn't automatically reflected in Power BI. However, it's possible to then control such formatting within Power BI.
<b>Hierarchy versioning</b>	SAP BW allows different versions of a hierarchy to be maintained, for example, the cost center hierarchy in 2007 versus 2008. Only the latest version is available in Power BI, as information on versions isn't exposed by the public API.
<b>Time dependent hierarchies</b>	When using Power BI, time dependent hierarchies are evaluated at the current date.
<b>Currency conversion</b>	SAP BW supports currency conversion, based on rates held in the cube. Such capabilities aren't exposed by the public API, and are therefore not available in Power BI.
<b>Sort Order</b>	The sort order, such as <i>by Text</i> or <i>by Key</i> , for a characteristic can be defined in SAP. This sort order isn't reflected in Power BI. For example, months might appear as "April", "Aug", and so on.
	It's not possible to change this sort order in Power BI.
<b>Technical names</b>	In <b>Get Data</b> , the characteristic/measure names (descriptions) and technical names can both be seen. The field list contains just the characteristic/measure names (descriptions).
<b>Attributes</b>	It's not possible to access the attributes of a characteristic within Power BI.
<b>End user language setting</b>	The locale used to connect to SAP BW is set as part of the connection details, and doesn't reflect the locale of the final report consumer.
<b>Text variables</b>	SAP BW allows field names to contain placeholders for variables, for example, <code>\$YEAR\$ Actuals</code> , that would then get replaced by the selected value. For example, the field appears as <i>2016 Actuals</i> in BEx tools, if the year 2016 were selected for the variable.

Feature	Description
	The column name in Power BI isn't changed depending on the variable value, and therefore would appear as <code>\$YEAR\$ Actuals</code> . However, the column name can then be changed in Power BI.
Customer exit variables	Customer exit variables aren't exposed by the public API, and are therefore not supported by Power BI.
Characteristic structures	Any characteristic structures in the underlying SAP BW source results in an explosion of measures being exposed in Power BI. For example, with two measures <code>Sales</code> and <code>Costs</code> , and a characteristic structure containing Budget and Actual, four measures are exposed: <code>Sales.Budget</code> , <code>Sales.Actual</code> , <code>Costs.Budget</code> , <code>Costs.Actual</code> .

## Next steps

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

- [DirectQuery in Power BI](#)
- [Data Sources supported by DirectQuery](#)
- [DirectQuery and SAP HANA](#)

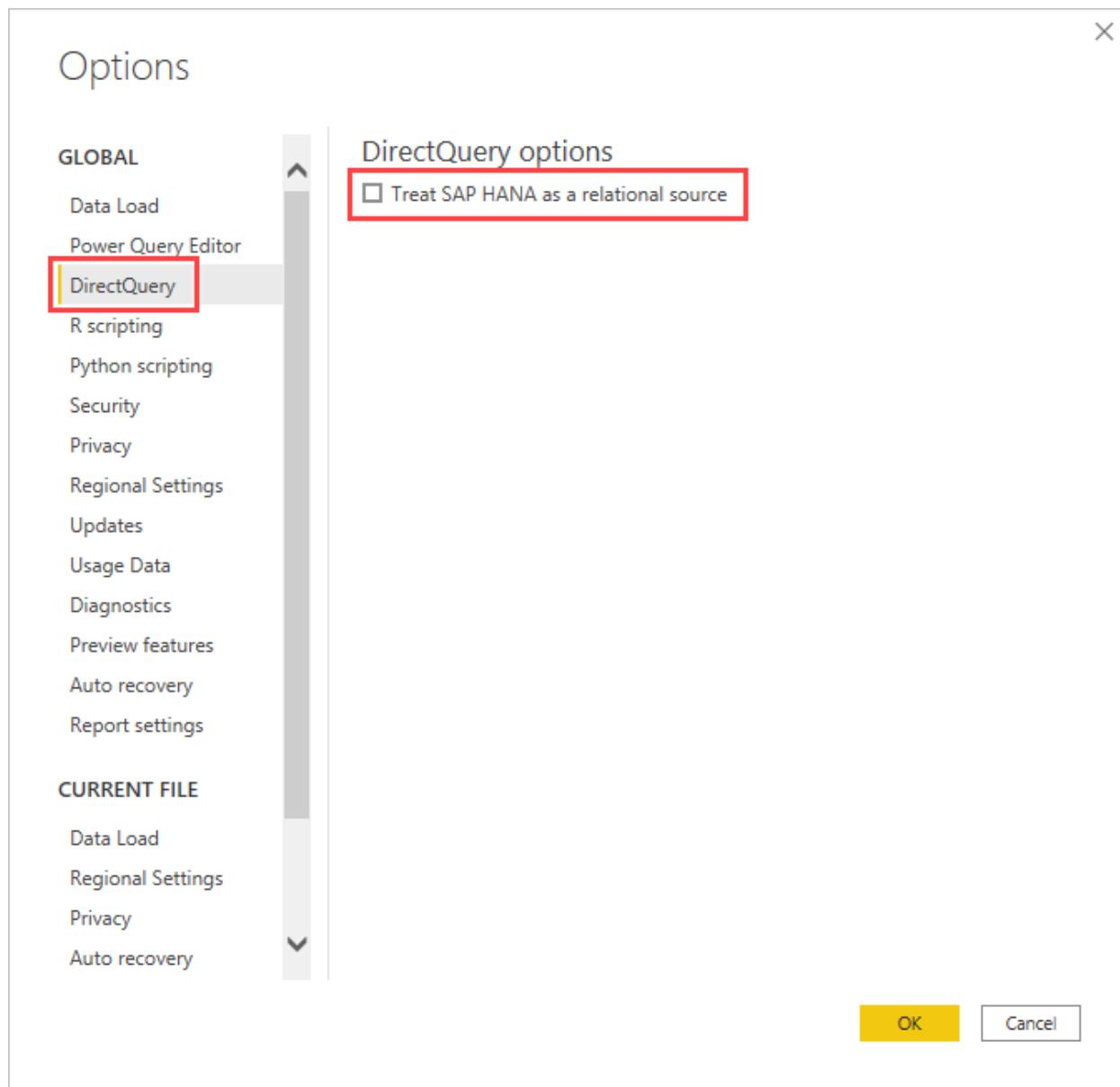
# Connect to SAP HANA data sources by using DirectQuery in Power BI

Article • 07/05/2023

You can connect to SAP HANA data sources directly using DirectQuery. There are two options when connecting to SAP HANA:

- **Treat SAP HANA as a multi-dimensional source (default):** In this case, the behavior is similar to when Power BI connects to other multi-dimensional sources like SAP Business Warehouse, or Analysis Services. When you connect to SAP HANA using this setting, a single analytic or calculation view is selected and all the measures, hierarchies and attributes of that view are available in the field list. As visuals are created, the aggregate data is always retrieved from SAP HANA. This technique is the recommended approach, and is the default for new DirectQuery reports over SAP HANA.
- **Treat SAP HANA as a relational source:** In this case, Power BI treats SAP HANA as a relational source. This approach offers greater flexibility. Care must be taken with this approach to ensure that measures are aggregated as expected, and to avoid performance issues.

The connection approach is determined by a global tool option, which is set by selecting **File > Options and settings** and then **Options > DirectQuery**, then selecting the option **Treat SAP HANA as a relational source**, as shown in the following image.



The option to treat SAP HANA as a relational source controls the approach used for any *new* report using DirectQuery over SAP HANA. It has no effect on any existing SAP HANA connections in the current report, nor on connections in any other reports that are opened. So if the option is currently unchecked, then upon adding a new connection to SAP HANA using **Get Data**, that connection is made treating SAP HANA as a multi-dimensional source. However, if a different report is opened that also connects to SAP HANA, then that report continues to behave according to the option that was set *at the time it was created*. This fact means that any reports connecting to SAP HANA that were created prior to February 2018 continue to treat SAP HANA as a relational source.

The two approaches constitute different behavior, and it's not possible to switch an existing report from one approach to the other.

## Treat SAP HANA as a multi-dimensional source (default)

All new connections to SAP HANA use this connection method by default, treating SAP HANA as a multi-dimensional source. In order to treat a connection to SAP HANA as a relational source, you must select **File > Options and settings > Options**, then check the box under **Direct Query > Treat SAP HANA as a relational source**.

When connecting to SAP HANA as a multi-dimensional source, the following considerations apply:

- In the **Get Data Navigator**, a single SAP HANA view can be selected. It isn't possible to select individual measures or attributes. There's no query defined at the time of connecting, which is different from importing data or when using DirectQuery while treating SAP HANA as a relational source. This consideration also means that it's not possible to directly use an SAP HANA SQL query when selecting this connection method.
- All the measures, hierarchies, and attributes of the selected view are displayed in the field list.
- As a measure is used in a visual, SAP HANA is queried to retrieve the measure value at the level of aggregation necessary for the visual. When dealing with non-additive measures, such as counters and ratios, all aggregations are performed by SAP HANA, and no further aggregation is performed by Power BI.
- To ensure the correct aggregate values can always be obtained from SAP HANA, certain restrictions must be imposed. For example, it's not possible to add calculated columns, or to combine data from multiple SAP HANA views within the same report.

Treating SAP HANA as a multi-dimensional source doesn't offer the greater flexibility provided by the alternative *relational* approach, but it's simpler. The approach also ensures correct aggregate values when dealing with more complex SAP HANA measures, and generally results in higher performance.

The **Field** list includes all measures, attributes, and hierarchies from the SAP HANA view. Note the following behaviors that apply when using this connection method:

- Any attribute that is included in at least one hierarchy is hidden by default. However, they can be seen if required by selecting **View hidden** from the context menu on the field list. From the same context menu they can be made visible, if necessary.
- In SAP HANA, an attribute can be defined to use another attribute as its label. For example, **Product**, with values **1**, **2**, **3**, and so on, could use **ProductName**, with values **Bike**, **Shirt**, **Gloves**, and so on, as its label. In this case, a single field

**Product** is shown in the field list, whose values are the labels `Bike`, `Shirt`, `Gloves`, and so on, but which is sorted by, and with uniqueness determined by, the key values `1`, `2`, `3`. A hidden column **Product.Key** is also created, allowing access to the underlying key values if necessary.

Any variables defined in the underlying SAP HANA view are displayed at the time of connecting, and the necessary values can be entered. Those values can later be changed by selecting **Transform data** from the ribbon, and then **Edit parameters** from the dropdown menu displayed.

The modeling operations allowed are more restrictive than in the general case when using DirectQuery, given the need to ensure that correct aggregate data can always be obtained from SAP HANA. However, it's still possible to make many additions and changes, including defining measures, renaming and hiding fields, and defining display formats. All such changes are preserved on refresh, and any non-conflicting changes made to the SAP HANA view are applied.

## Additional modeling restrictions

The other primary modeling restrictions when connecting to SAP HANA using DirectQuery (treat as multi-dimensional source) are the following restrictions:

- **No support for calculated columns:** The ability to create calculated columns is disabled. This fact also means that Grouping and Clustering, which create calculated columns, aren't available.
- **Additional limitations for measures:** There are other limitations imposed on the DAX expressions that can be used in measures, to reflect the level of support offered by SAP HANA.
- **No support for defining relationships:** Only a single view can be queried within a report, and as such, there's no support for defining relationships.
- **No Data View:** The **Data View** normally displays the detail level data in the tables. Given the nature of OLAP sources such as SAP HANA, this view isn't available over SAP HANA.
- **Column and measure details are fixed:** The list of columns and measures seen in the field list are fixed by the underlying source, and can't be modified. For example, it's not possible to delete a column, nor change its datatype. It can, however, be renamed.
- **Additional limitations in DAX:** There are other limitations on the DAX that can be used in measure definitions, to reflect limitations in the source. For example, it's not possible to use an aggregate function over a table.

## Additional visualization restrictions

There are restrictions in visuals when connecting to SAP HANA using DirectQuery (treat as multi-dimensional source):

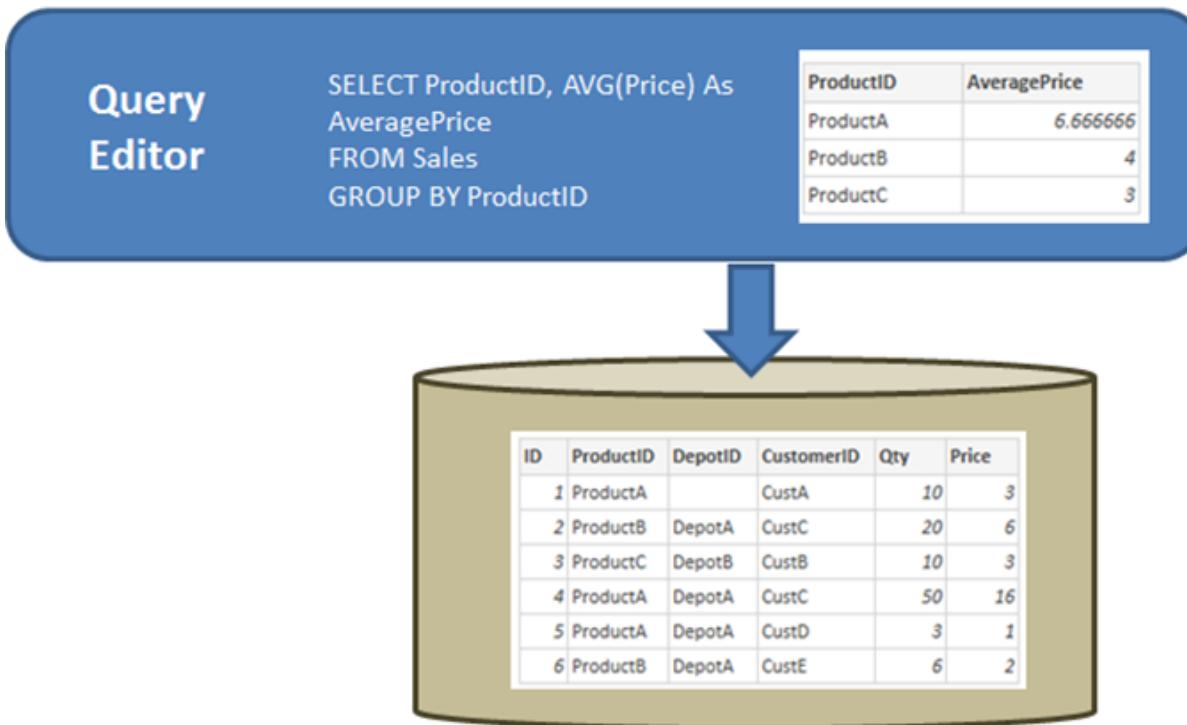
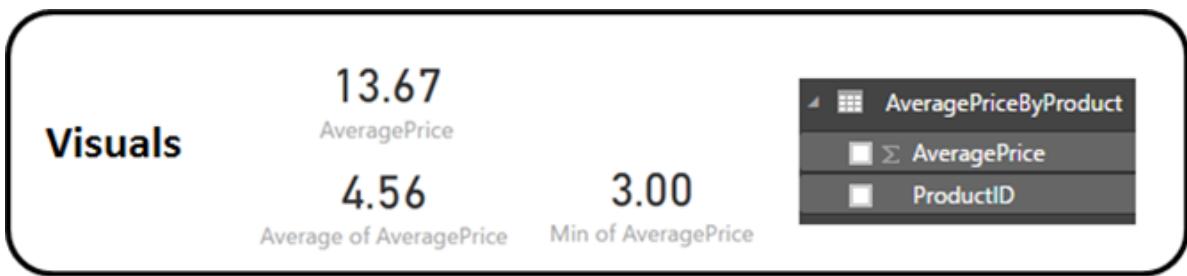
- **No aggregation of columns:** It's not possible to change the aggregation for a column on a visual, and it's always *Do Not Summarize*.

## Treat SAP HANA as a relational source

When choosing to connect to SAP HANA as a relational source, some extra flexibility becomes available. For example, you can create calculated columns, include data from multiple SAP HANA views, and create relationships between the resulting tables.

However, there are differences from the behavior when treating SAP HANA as a multidimensional source, particularly when the SAP HANA view contains non-additive measures, for example, distinct counts, or averages, rather than simple sums, and related to the efficiency of the queries that are run against SAP HANA.

It's useful to start by clarifying the behavior of a relational source such as SQL Server, when the query defined in **Get Data** or Power Query Editor performs an aggregation. In the example that follows, a query defined in Power Query Editor returns the average price by *ProductID*.



If the data is being imported into Power BI versus using DirectQuery, the following situation would result:

- The data is imported at the level of aggregation defined by the query created in Power Query Editor. For example, average price by product. This fact results in a table with the two columns *ProductID* and *AveragePrice* that can be used in visuals.
- In a visual, any subsequent aggregation, such as *Sum*, *Average*, *Min*, and others, is performed over that imported data. For example, including *AveragePrice* on a visual uses the *Sum* aggregate by default, and would return the sum over the *AveragePrice* for each *ProductID*, in this example, 13.67. The same applies to any alternative aggregate function, such as *Min* or *Average*, used on the visual. For example, *Average of AveragePrice* returns the average of 6.66, 4 and 3, which equates to 4.56, and not the average of *Price* on the six records in the underlying table, which is 5.17.

If DirectQuery over that same relational source is being used instead of Import, the same semantics apply and the results would be exactly the same:

- Given the same query, logically exactly the same data is presented to the reporting layer – even though the data isn't actually imported.

- In a visual, any subsequent aggregation, such as *Sum*, *Average*, and *Min*, is again performed over that logical table from the query. And again, a visual containing *Average of AveragePrice* returns the same 4.56.

Consider SAP HANA when the connection is treated as a relational source. Power BI can work with both *Analytic Views* and *Calculation Views* in SAP HANA, both of which can contain measures. Yet today the approach for SAP HANA follows the same principles as described previously in this section: the query defined in **Get Data** or **Power Query Editor** determines the data available, and then any subsequent aggregation in a visual is over that data, and the same applies for both Import and DirectQuery. However, given the nature of SAP HANA, the query defined in the initial **Get Data** dialog or **Power Query Editor** is always an aggregate query, and generally includes measures where the actual aggregation that are used is defined by the SAP HANA view.

The equivalent of the previous SQL Server example is that there's an SAP HANA view containing *ID*, *ProductID*, *DepotID*, and measures including *AveragePrice*, defined in the view as *Average of Price*.

If in the **Get Data** experience, the selections made were for **ProductID** and the **AveragePrice** measure, then that is defining a query over the view, requesting that aggregate data. In the earlier example, for simplicity pseudo-SQL is used that doesn't match the exact syntax of SAP HANA SQL. Then any further aggregations defined in a visual are further aggregating the results of such a query. Again, as described previously for SQL Server, this result applies both for the Import and DirectQuery case. In the DirectQuery case, the query from **Get Data** or **Power Query Editor** are used in a subselect within a single query sent to SAP HANA, and thus it isn't actually the case that all the data would be read in, prior to aggregating further.

All of these considerations and behaviors necessitate the following important considerations when using DirectQuery over SAP HANA:

- Attention must be paid to any further aggregation performed in visuals, whenever the measure in SAP HANA is non-additive, for example, not a simple *Sum*, *Min*, or *Max*.
- In **Get Data** or **Power Query Editor**, only the required columns should be included to retrieve the necessary data, reflecting the fact that the result is a query that must be a reasonable query that can be sent to SAP HANA. For example, if dozens of columns were selected, with the thought that they might be needed on subsequent visuals, then even for DirectQuery a simple visual means the aggregate query used in the subselect contains those dozens of columns, which generally perform poorly.

In the following example, selecting five columns (**CalendarQuarter**, **Color**, **LastName**, **ProductLine**, **SalesOrderNumber**) in the **Get Data** dialog, along with the measure **OrderQuantity**, means that later creating a simple visual containing the **Min OrderQuantity** results in the following SQL query to SAP HANA. The shaded is the subselect, containing the query from **Get Data** / Power Query Editor. If this subselect gives a high cardinality result, then the resulting SAP HANA performance is likely to be poor.

```
SELECT MIN("t0"."OrderQuantity")
AS "a0"
FROM
(
(select "CalendarQuarter",
"Color",
"LastName",
"ProductLine",
"SalesOrderNumber",
sum(cast("OrderQuantity" as DOUBLE)) as "OrderQuantity"
from "_SYS_BIC"."ADVENTUREWORKS/AN_RESELLERSALES"
group by "CalendarQuarter",
"Color",
"LastName",
"ProductLine",
"SalesOrderNumber")
)
AS "t0"
```

Because of this behavior, we recommend the items selected in **Get Data** or Power Query Editor be limited to those items that are needed, while still resulting in a reasonable query for SAP HANA.

## Best practices

For both approaches to connecting to SAP HANA, recommendations for using DirectQuery also apply to SAP HANA, particularly recommendations related to ensuring good performance. For more information, see [using DirectQuery in Power BI](#).

## Considerations and limitations

The following list describes all SAP HANA features that aren't fully supported, or features that behave differently when using Power BI.

- **Parent Child Hierarchies:** Parent child hierarchies aren't visible in Power BI. This fact is because Power BI accesses SAP HANA using the SQL interface, and parent child hierarchies can't be fully accessed by using SQL.
- **Other hierarchy metadata:** The basic structure of hierarchies is displayed in Power BI, however some hierarchy metadata, such as controlling the behavior of ragged hierarchies, have no effect. Again, this fact is due to the limitations imposed by the SQL interface.
- **Connection using SSL:** You can connect using Import and multi-dimensional with TLS, but can't connect to SAP HANA instances configured to use TLS for the relational connector.
- **Support for Attribute views:** Power BI can connect to Analytic and Calculation views, but can't connect directly to Attribute views.
- **Support for Catalog objects:** Power BI can't connect to Catalog objects.
- **Change to Variables after publish:** You can't change the values for any SAP HANA variables directly in the Power BI service, after the report is published.

## Known issues

The following list describes all known issues when connecting to SAP HANA (DirectQuery) using Power BI.

- **SAP HANA issue when query for Counters, and other measures:** Incorrect data is returned from SAP HANA if connecting to an Analytical View, and a Counter measure and some other ratio measure, are included in the same visual. This issue is covered by [SAP Note 2128928 \(Unexpected results when query a Calculated Column and a Counter\)](#). The ratio measure is incorrect in this case.
- **Multiple Power BI columns from single SAP HANA column:** For some calculation views, where an SAP HANA column is used in more than one hierarchy, SAP HANA exposes the column as two separate attributes. This approach results in two columns being created in Power BI. Those columns are hidden by default, however, and all queries involving the hierarchies, or the columns directly, behave correctly.

## Next steps

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

- [DirectQuery in Power BI](#)

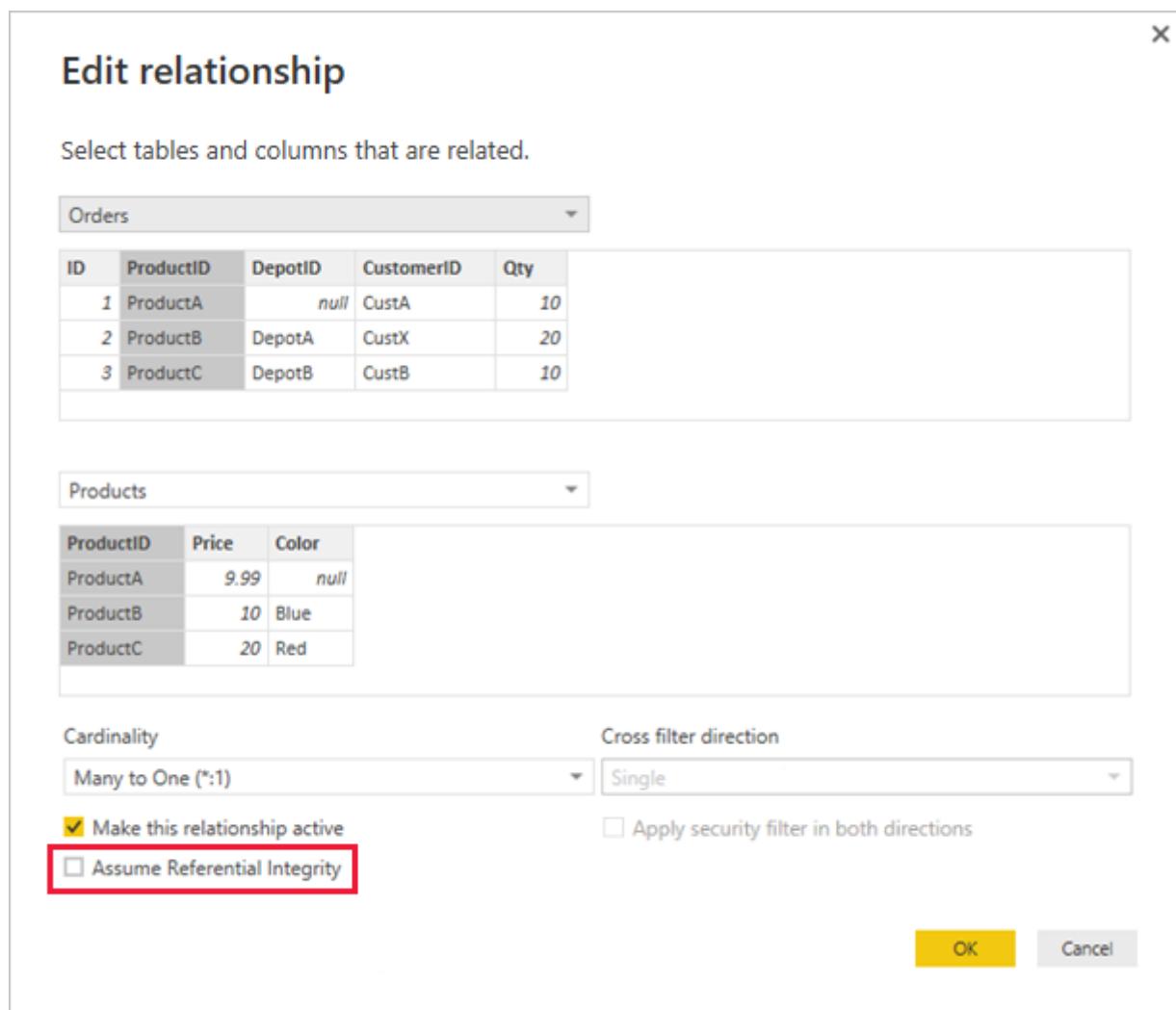
- Data sources supported by DirectQuery
- DirectQuery and SAP BW
- On-premises data gateway

# Apply the Assume Referential Integrity setting in Power BI Desktop

Article • 01/18/2023

When connecting to a data source using DirectQuery, you can use the **Assume Referential Integrity** selection to enable running more efficient queries against your data source. This feature has a few requirements of the underlying data, and it's only available when using DirectQuery.

Setting **Assume Referential Integrity** enables queries on the data source to use *INNER JOIN* statements rather than *OUTER JOIN*, which improves query efficiency.



## Requirements for using Assume Referential Integrity

This setting is an advanced setting, and is only enabled when connecting to data using DirectQuery. The following requirements are necessary for **Assume Referential Integrity**:

to work properly:

- Data in the **From** column in the relationship is never *Null* or *blank*
- For each value in the **From** column, there's a corresponding value in the **To** column

In this context, the **From** column is the *Many* in a *One-to-Many* relationship, or it's the column in the first table in a *One-to-One* relationship.

## Example of using Assume Referential Integrity

The following example demonstrates how **Assume Referential Integrity** behaves when used in data connections. The example connects to a data source that includes an **Orders** table, a **Products** table, and a **Depots** table.

- In the following image that shows the **Orders** table and the **Products** table, referential integrity exists between **Orders[ProductID]** and **Products[ProductID]**. The **[ProductID]** column in the **Orders** table is never *Null*, and every value also appears in the **Products** table. As such, **Assume Referential Integrity** should be set to get more efficient queries. Using this setting doesn't change the values shown in visuals.

Orders table				
ID	ProductID	DepotID	CustomerID	Qty
1	ProductA	<i>null</i>	CustA	10
2	ProductB	DepotA	CustX	20
3	ProductC	DepotB	CustB	10

Products table		
ProductID	Price	Color
ProductA	9.99	<i>null</i>
ProductB	10	Blue
ProductC	20	Red

- In the next image, notice that no referential integrity exists between **Orders[DepotID]** and **Depots[DepotID]**, because the **DepotID** is *Null* for some **Orders**. As such, **Assume Referential Integrity** should *not* be set.

**Orders  
table**

ID	ProductID	DepotID	CustomerID	Qty
1	ProductA	null	CustA	10
2	ProductB	DepotA	CustX	20
3	ProductC	DepotB	CustB	10

**Depots  
table**

DepotID	City
DepotA	Seattle
DepotB	New York

- Finally, no referential integrity exists between **Orders[CustomerID]** and **Customers[CustID]** in the following tables. The **CustomerID** contains some values, in this case, *CustX*, that don't exist in the *Customers* table. As such, **Assume Referential Integrity** should *not* be set.

**Orders  
table**

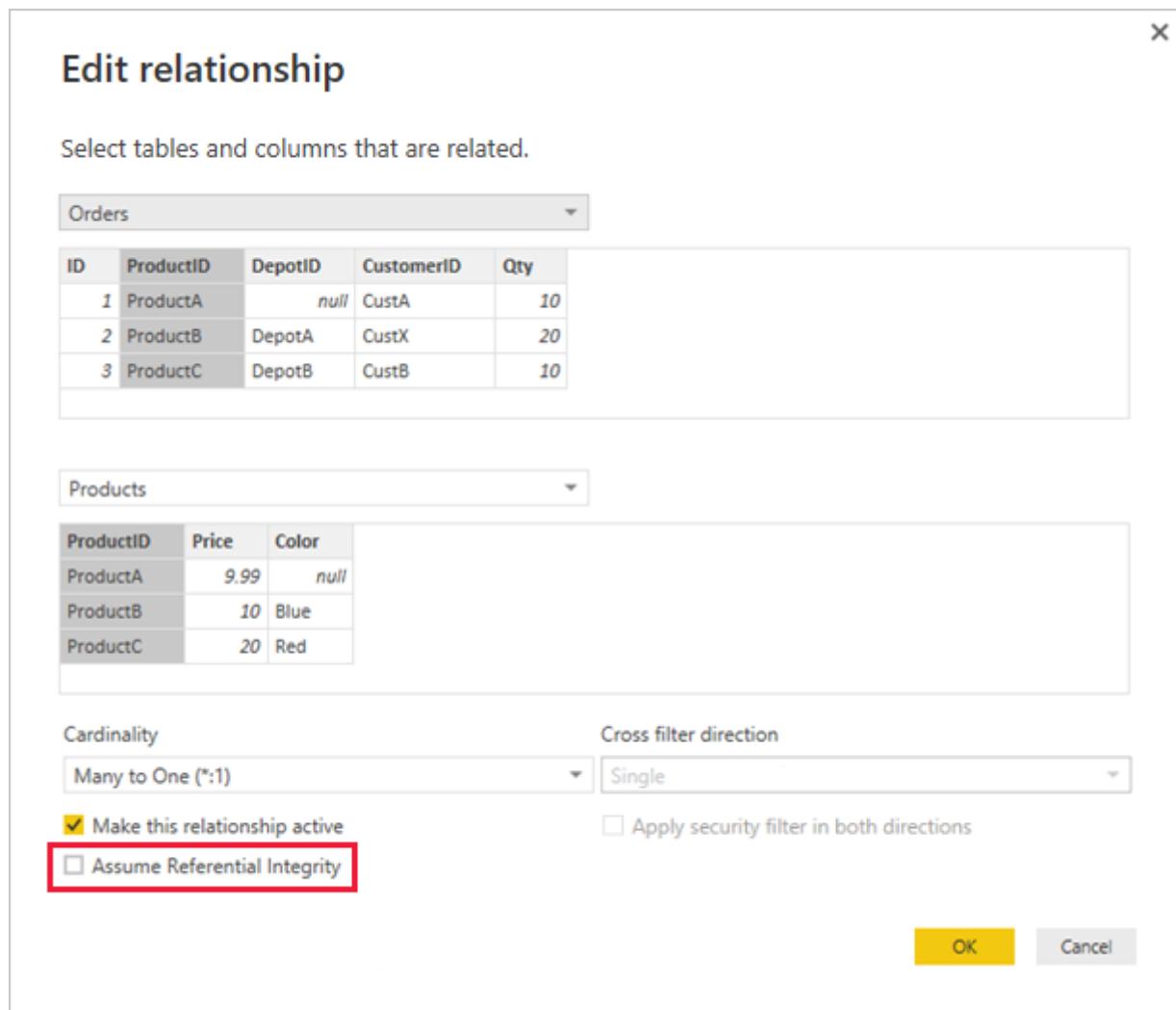
ID	ProductID	DepotID	CustomerID	Qty
1	ProductA	null	CustA	10
2	ProductB	DepotA	CustX	20
3	ProductC	DepotB	CustB	10

**Customers  
table**

CustID	Name
CustA	John Doe
CustB	Jack Smith

## Setting Assume Referential Integrity

To enable this feature, select **Assume Referential Integrity** as shown in the following image.



When selected, the setting is validated against the data to ensure there are no *Null* or mismatched rows. However, for cases with a very large number of values, the validation isn't a guarantee that there are no referential integrity issues.

In addition, the validation occurs at the time of editing the relationship, and does *not* reflect any subsequent changes to the data.

## What happens if you incorrectly set Assume Referential Integrity?

If you set **Assume Referential Integrity** when there are referential integrity issues in the data, that setting doesn't result in errors. However, it does result in apparent inconsistencies in the data. For example, for the relationship to the **Depots** table described here, it would result in the following:

- A visual showing the total *Order Qty* would show a value of 40
- A visual showing the total *Order Qty by Depot City* would show a total value of only 30, because it wouldn't include Order ID 1, where **DepotID** is *Null*.

## Next steps

- Learn more about [DirectQuery](#).
- Get more information about [Relationships in Power BI](#).
- Learn more about [Relationship View in Power BI Desktop](#).

# Use the SAP Business Warehouse connector in Power BI Desktop

Article • 01/24/2023

You can use Power BI Desktop to access SAP Business Warehouse (SAP BW) data. The SAP BW Connector Implementation 2.0 has significant improvements in performance and capabilities from version 1.0.

For information about how SAP customers can benefit from connecting Power BI to their SAP BW systems, see the [Power BI and SAP BW whitepaper](#). For details about using DirectQuery with SAP BW, see [DirectQuery and SAP Business Warehouse \(BW\)](#).

## Important

Version 1.0 of the SAP BW connector is deprecated. New connections use Implementation 2.0 of the SAP BW connector. All support for version 1.0 will be removed from the connector in the near future. Use the information in this article to update existing version 1.0 reports to use Implementation 2.0 of the connector.

## Use the SAP BW Connector

Follow these steps to install and connect to data with the SAP BW Connector.

### Prerequisite

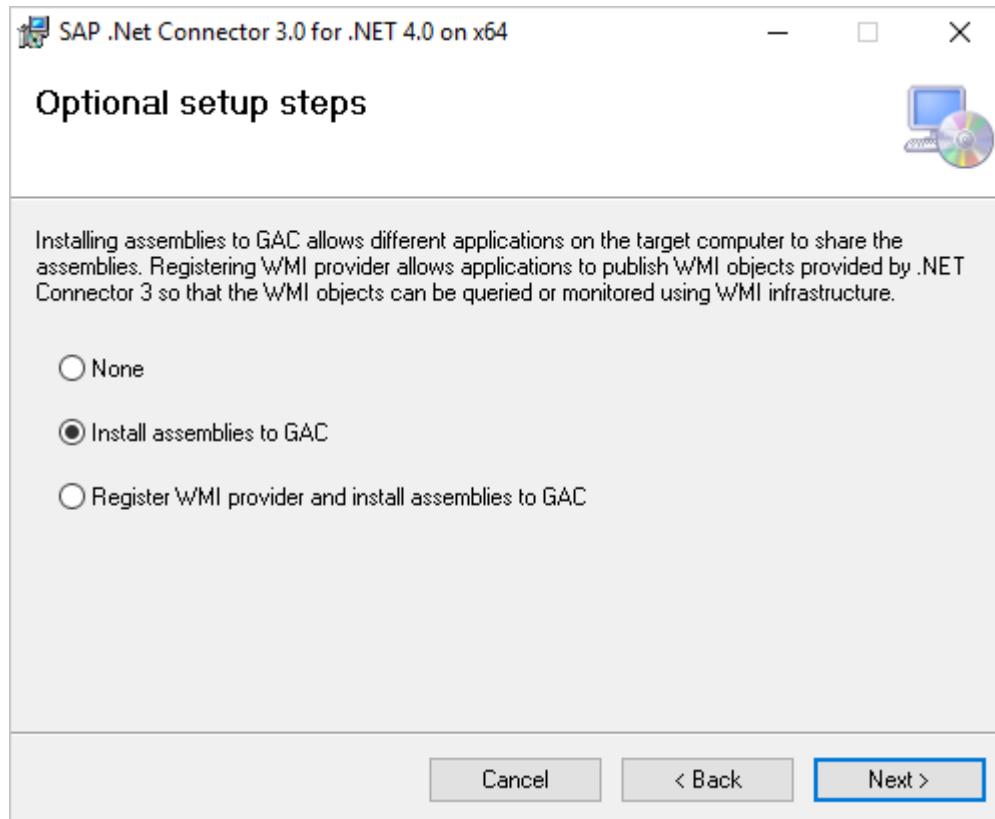
Implementation 2.0 of the SAP Connector requires the SAP .NET Connector 3.0. You can download the [SAP .NET Connector 3.0](#) from SAP. Access to the download requires a valid S-user sign-in.

## Important

Be sure to use SAP .NET Connector 3.0. The SAP BW Application Server connector doesn't currently support SAP .NET Connector 3.1.

The .NET 4.0 framework connector comes in 32-bit and 64-bit versions. Choose the version that matches your Power BI Desktop installation version.

When you install, in **Optional setup steps**, make sure you select **Install assemblies to GAC**.



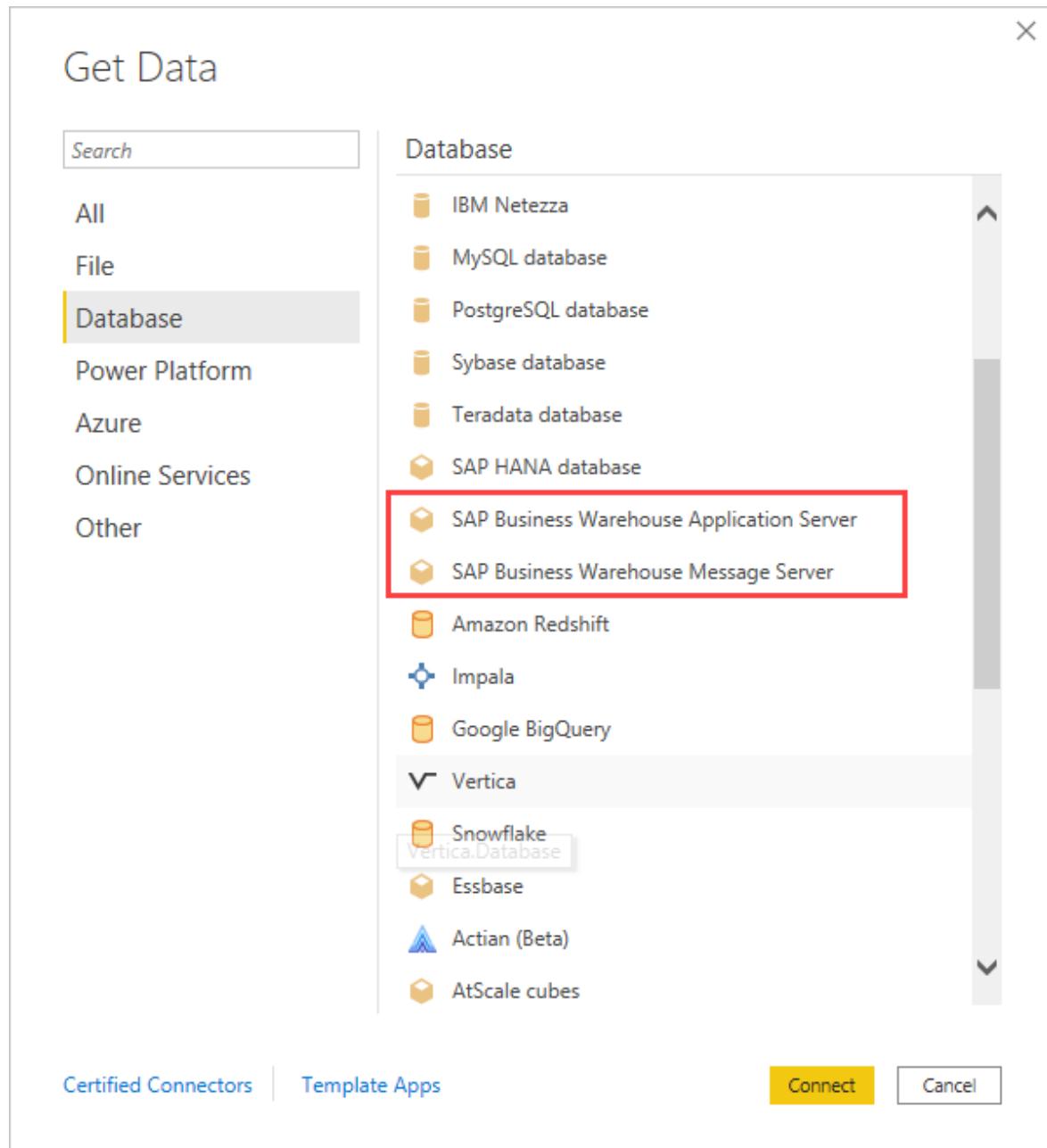
#### **ⓘ Note**

The first version of the SAP BW Connector required the NetWeaver DLLs. The current version doesn't require NetWeaver DLLs.

## Connect to SAP BW data in Power BI Desktop

To connect to SAP BW data by using the SAP BW Connector, follow these steps:

1. In Power BI Desktop, select **Get data**.
2. On the **Get Data** screen, select **Database**, and then select either **SAP Business Warehouse Application Server** or **SAP Business Warehouse Message Server**.



3. Select **Connect**.

4. On the next screen, enter server, system, and client information, and whether to use **Import** or **DirectQuery** connectivity method. For detailed instructions, see:
- [Connect to an SAP BW Application Server from Power Query Desktop](#)
  - [Connect to an SAP BW Message Server from Power Query Desktop](#)

**(!) Note**

You can use the SAP BW Connector to import data from your SAP BW Server cubes, which is the default, or you can use DirectQuery to connect to the data. For more information about using the SAP BW Connector with DirectQuery, see [DirectQuery and SAP Business Warehouse \(BW\)](#).

You can also select **Advanced options**, and select a **Language code**, a custom **MDX statement** to run against the specified server, and other options. For more information, see [Use advanced options](#).

5. Select **OK** to establish the connection.
6. Provide any necessary authentication data and select **Connect**. For more information about authentication, see [Authentication with a data source](#).
7. If you didn't specify a custom MDX statement, the **Navigator** screen shows a list of all cubes available on the server. You can drill down and select items from the available cubes, including dimensions and measures. Power BI shows queries and cubes that the [Open Analysis Interfaces](#) expose.

When you select one or more items from the server, the **Navigator** shows a preview of the output table.

The screenshot shows the 'Navigator' dialog box. On the left, there is a tree view under 'Display Options' with several items expanded, including 'Purchasing Data [23]' which has 'Invoice Amount' checked. To the right is a preview table titled 'Purchasing Data' with columns 'Vendor.Vendor Level 01', 'Invoice Amount', and 'No. of Deliveries'. The table lists various companies with their respective invoice amounts and delivery counts. At the bottom of the dialog are buttons for 'Extract Table Using Examples', 'Load' (highlighted in yellow), 'Transform Data', and 'Cancel'.

Vendor.Vendor Level 01	Invoice Amount	No. of Deliveries
Sunny Electronics Inc.	5588152	184
PAQ Germany GmbH	9132897	189
Jotachi Deutschland AG	4309632	138
Computer 3000 Inc.	4420784	90
CompSmart Inc.	13285140	207
Technik und Systeme GmbH	3331875	72
Phoenix Supplies	2604640	83
Pyramid Systems	5162710	205
HiTech Corp	2537990	41
Omnium	5929952	174
Asia Technologies	460502	9
Computer Competence Center AG	5066768	116
PC Warehouse	2346024	52
CompuMax Corp	10730400	164
C.E.B Paris	2021691	66
Becker Components AG	16654265	255
Générale Electronique	7725896	63
Logo Systems	4177492	128
SAPSOTA Corp	341640	7
ABC Technology	29656584	205
Superminus	3400195	66
Hatsuhiba Co. Ltd	14842830	193
Marwick Inc.	7276500	122
#	0	0

The **Navigator** dialog also provides the following display options:

- **Only selected items.** By default, **Navigator** displays all items. This option is useful to verify the final set of items you select. Alternatively, you can select the column names in the preview area to view the selected items.
- **Enable data previews.** This value is the default, and displays data previews. Deselect this option to reduce the number of server calls by no longer

requesting preview data.

- **Technical names.** SAP BW supports user-defined *technical names* for objects within a cube. Cube owners can expose these friendly names for cube objects, instead of exposing only the physical names for the objects.

The screenshot shows the SAP BW Navigator interface. On the left, there is a tree view of objects under the 'Purch' category, with several checkboxes checked. A context menu is open over one of the checked items, showing options: 'Only selected items', 'Enable data previews' (which is checked), and 'Technical names'. To the right of the tree view is a table titled 'Purchasing Data' with three columns: 'Vendor.Vendor Level 01', 'Invoice Amount', and 'No. of Deliveries'. The table lists various companies with their respective invoice amounts and delivery counts. At the bottom of the interface are buttons for 'Extract Table Using Examples', 'Load' (highlighted in yellow), 'Transform Data', and 'Cancel'.

Vendor.Vendor Level 01	Invoice Amount	No. of Deliveries
Sunny Electronics Inc.	5588152	184
PAQ Germany GmbH	9132897	189
Jotachi Deutschland AG	4309632	138
Computer 3000 Inc.	4420784	90
CompSmart Inc.	13285140	207
Technik und Systeme GmbH	3331875	72
Phoenix Supplies	2604640	83
Pyramid Systems	5162710	205
HiTech Corp	2537990	41
Omnium	5929962	174
Asia Technologies	460502	9
Computer Competence Center AG	5066768	116
PC Warehouse	2346024	52
CompuMax Corp	10730400	164
C.E.B Paris	2021691	66
Becker Components AG	16654265	255
Générale Electronique	7725896	63
Logo Systems	4177492	128
SAPSOTA Corp	341640	7
ABC Technology	29656584	205
Superminus	3400195	66
Hatushiba Co. Ltd	14842830	193
Marwick Inc.	7276500	122
	0	0

8. After you select all the objects you want, choose one of the following options:

- **Load** to load the entire set of rows for the output table into the Power BI Desktop data model. The **Report** view opens. You can begin visualizing the data, or make further modifications by using the **Data** or **Model** views.
- **Transform Data** to open **Power Query Editor** with the data. You can specify more data transformation and filtering steps before you bring the entire set of rows into the Power BI Desktop data model.

Along with data from SAP BW cubes, you can also import data from a wide range of other data sources in Power BI Desktop, and combine them into a single report. This ability presents many interesting scenarios for reporting and analytics on top of SAP BW data.

## New options in SAP BW Implementation 2.0

This section lists some SAP BW Connector Implementation 2.0 features and improvements. For more information, see [Implementation details](#).

## Advanced options

You can set the following options under **Advanced options** on the SAP BW connection screen:

- **Execution mode** specifies how the MDX interface executes queries on the server. The following options are valid:
  - **BasXml**
  - **BasXmlGzip**
  - **DataStream**The default value is **BasXmlGzip**. This mode can improve performance for low latency or high volume queries.
- **Batch size** specifies the maximum number of rows to retrieve at a time when executing an MDX statement. A small number means more calls to the server while retrieving a large dataset. A large value might improve performance, but could cause memory issues on the SAP BW server. The default value is **50000**.
- **Enable characteristic structures** changes the way the **Navigator** displays characteristic structures. The default value for this option is false, or unchecked. This option affects the list of objects available for selection, and isn't supported in native query mode.

## Other improvements

The following list describes other Implementation 2.0 improvements:

- Better performance.
- Ability to retrieve several million rows of data, and fine-tuning through the batch size parameter.
- Ability to switch execution modes.
- Support for compressed mode, especially beneficial for high-latency connections or large datasets.
- Improved detection of **Date** variables.
- **Date** (ABAP type **DATS**) and **Time** (ABAP type **TIMS**) dimensions exposed as dates and times, instead of text values. For more information, see [Support for typed dates in SAP BW](#).
- Better exception handling. Errors that occur in BAPI calls are now surfaced.

- Column folding in `BasXml` and `BasXmlGzip` modes. For example, if the generated MDX query retrieves 40 columns but the current selection only needs 10, this request passes on to the server to retrieve a smaller dataset.

## Update existing Implementation 1.0 reports

You can change existing reports to use Implementation 2.0 only in Import mode.

1. From the existing report in Power BI Desktop, select **Transform data** in the ribbon, and then select the SAP Business Warehouse query to update.
2. Right-click the query and select **Advanced Editor**.
3. In the **Advanced Editor**, change the `SapBusinessWarehouse.Cubes` calls as follows:
4. Determine whether the query already contains an option record, such as the following examples:

```
Source = SapBusinessWarehouse.Cubes("someserver", "00", "900",
[ScaleMeasures=false])

Source = SapBusinessWarehouse.Cubes("sapbw73", "00", "900", [Query="SELECT {
[Measures].[006EI86RZI4D2RTX6HQP12JIY] }ON 0, NON EMPTY CROSSJOIN ... ON 1 FROM
[0D_SD_C03/REP_20180117030589]"])
```

If so, add the `[Implementation=2.0]` option, and remove any `ScaleMeasures` option:

```
Source = SapBusinessWarehouse.Cubes("someserver", "00", "900",
[Implementation="2.0"])

Source = SapBusinessWarehouse.Cubes("sapbw73", "00", "900",
[Implementation="2.0"], Query="SELECT {
[Measures].[006EI86RZI4D2RTX6HQP12JIY] }ON 0, NON EMPTY CROSSJOIN ... ON 1 FROM
[0D_SD_C03/REP_20180117030589]")
```

### Note

The `ScaleMeasures` option is deprecated in this implementation. The connector now always shows unscaled values.

5. If the query doesn't already include an options record, add it. For example, change the following entry:

```
Source = SapBusinessWarehouse.Cubes("someserver", "00", "900")
```

to:

```
Source = SapBusinessWarehouse.Cubes("someserver", "00", "900",
[Implementation="2.0"])
```

### ⓘ Note

Implementation 2.0 of the SAP BW Connector should be compatible with version 1. However, there might be some differences because of the different SAP BW MDX execution modes. To resolve any discrepancies, try switching between execution modes.

## Troubleshooting

This section provides some troubleshooting situations and solutions for the SAP BW connector. For more information, see [SAP Business Warehouse connector troubleshooting](#).

### Numeric data from SAP BW returns misformatted numeric data

In this issue, SAP BW returns numeric data with decimal points instead of commas. For example, 1,000,000 returns as 1.000.000.

SAP BW returns decimal data with either a comma or a period as the decimal separator. To specify which of these characters SAP BW should use for the decimal separator, the Power BI Desktop driver makes a call to `BAPI_USER_GET_DETAIL`. This call returns a structure called `DEFAULTS`, which has a field called `DCPFM` that stores *Decimal Format Notation* as one of the following values:

- ' ' (space) = Decimal point is comma: `N.NNN,NN`
- 'X' = Decimal point is period: `N,NNN.NN`
- 'Y' = Decimal point is N: `NNN NNN,NN`

With this issue, the call to `BAPI_USER_GET_DETAIL` fails for a particular user, who gets the misformatted data, with an error message similar to the following message:

#### XML

```
You are not authorized to display users in group TI:
<item>
```

```

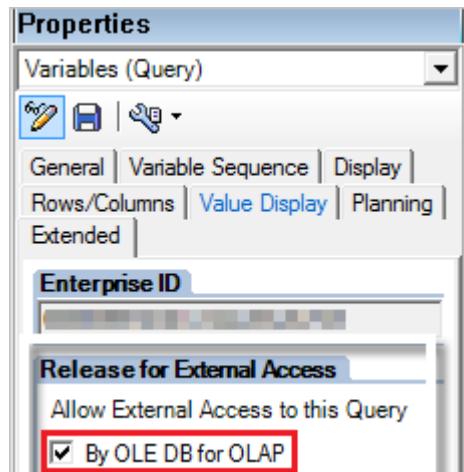
<TYPE>E</TYPE>
<ID>01</ID>
<NUMBER>512</NUMBER>
<MESSAGE>You are not authorized to display users in group
TI</MESSAGE>
<LOG_NO/>
<LOG_MSG_NO>000000</LOG_MSG_NO>
<MESSAGE_V1>TI</MESSAGE_V1>
<MESSAGE_V2/>
<MESSAGE_V3/>
<MESSAGE_V4/>
<PARAMETER/>
<ROW>0</ROW>
<FIELD>BNAME</FIELD>
<SYSTEM>CLNTPW1400</SYSTEM>
</item>

```

To solve this error, the SAP admin must grant the Power BI SAP BW user the right to execute `BAPI_USER_GET_DETAIL`. Also, verify that the user's data has the correct `DCPFM` value.

## Need connectivity for SAP BEx queries

You can do BEx queries in Power BI Desktop by enabling the **Release for External Access** property, as shown in the following image:



## Navigator doesn't display a data preview

In this issue, **Navigator** doesn't display a data preview and instead shows an **Object reference not set to an instance of an object** error message.

SAP users need access to the following specific BAPI function modules to get metadata and retrieve data from SAP BW's InfoProviders:

- `BAPI_MDPROVIDER_GET_CATALOGS`

- BAPI\_MDPROVIDER\_GET\_CUBES
- BAPI\_MDPROVIDER\_GET\_DIMENSIONS
- BAPI\_MDPROVIDER\_GET\_HIERARCHYS
- BAPI\_MDPROVIDER\_GET\_LEVELS
- BAPI\_MDPROVIDER\_GET\_MEASURES
- BAPI\_MDPROVIDER\_GET\_MEMBERS
- BAPI\_MDPROVIDER\_GET\_VARIABLES
- BAPI\_IOBJ\_GETDETAIL

To solve this issue, verify that the user has access to the `MDPROVIDER` modules and `BAPI_IOBJ_GETDETAIL`.

## Enable tracing

To further troubleshoot these or similar issues, you can enable tracing:

1. In Power BI Desktop, select **File > Options and settings > Options**.
2. In **Options**, select **Diagnostics**, and then select **Enable tracing** under **Diagnostic Options**.
3. Try to get data from SAP BW while tracing is active, and examine the trace file for more detail.

## SAP BW Connection support

The following table describes current Power BI support for SAP BW.

<b>Product</b>	<b>Mode</b>	<b>Authentication</b>	<b>Connector</b>	<b>SNC Library</b>	<b>Supported</b>
Power BI Desktop	Any	User / password	Application Server	N/A	Yes
Power BI Desktop	Any	Windows	Application Server	sapcrypto + gsskrb5/gx64krb5	Yes
Power BI Desktop	Any	Windows via impersonation	Application Server	sapcrypto + gsskrb5/gx64krb5	Yes
Power BI Desktop	Any	User / password	Message Server	N/A	Yes
Power BI Desktop	Any	Windows	Message Server	sapcrypto + gsskrb5/gx64krb5	Yes
Power BI Desktop	Any	Windows via impersonation	Message Server	sapcrypto + gsskrb5/gx64krb5	Yes

<b>Product</b>	<b>Mode</b>	<b>Authentication</b>	<b>Connector</b>	<b>SNC Library</b>	<b>Supported</b>
Power BI Gateway	Import	Same as Power BI Desktop			
Power BI Gateway	DirectQuery	User / password	Application Server	N/A	Yes
Power BI Gateway	DirectQuery	Windows via impersonation (fixed user, no SSO)	Application Server	sapcrypto + gsskrb5/gx64krb5	Yes
Power BI Gateway	DirectQuery	Use SSO via Kerberos for DirectQuery queries option	Application Server	sapcrypto + gsskrb5/gx64krb5	Yes
Power BI Gateway	DirectQuery	User / password	Message Server	N/A	Yes
Power BI Gateway	DirectQuery	Windows via impersonation (fixed user, no SSO)	Message Server	sapcrypto + gsskrb5/gx64krb5	Yes
Power BI Gateway	DirectQuery	Use SSO via Kerberos for DirectQuery queries option	Message Server	gsskrb5/gx64krb5	No
Power BI Gateway	DirectQuery	Use SSO via Kerberos for DirectQuery queries option	Message Server	sapcrypto	Yes

## Next steps

- [SAP BW fundamentals](#)
- [DirectQuery and SAP HANA](#)
- [DirectQuery and SAP Business Warehouse \(BW\)](#)
- [Use DirectQuery in Power BI](#)
- [Power BI data sources](#)
- [Power BI and SAP BW whitepaper ↗](#)

# Use OneDrive for work or school links in Power BI Desktop

Article • 02/13/2023

Many people have Excel workbooks stored in OneDrive for work or school that would be great for use with Power BI Desktop. With Power BI Desktop, you can use online links for Excel files stored in OneDrive for work or school to create reports and visuals. You can use a OneDrive for work or school group account or your individual OneDrive for work or school account.

Getting an online link from OneDrive for work or school requires a few specific steps. The following sections explain those steps, which let you share the file link among groups, across different machines, and with your coworkers.

## Get a link from Excel

1. Navigate to your OneDrive for work or school location using a browser. Select the ellipses (...) to open the **More** menu, then select **Details**.

My files > New Folder

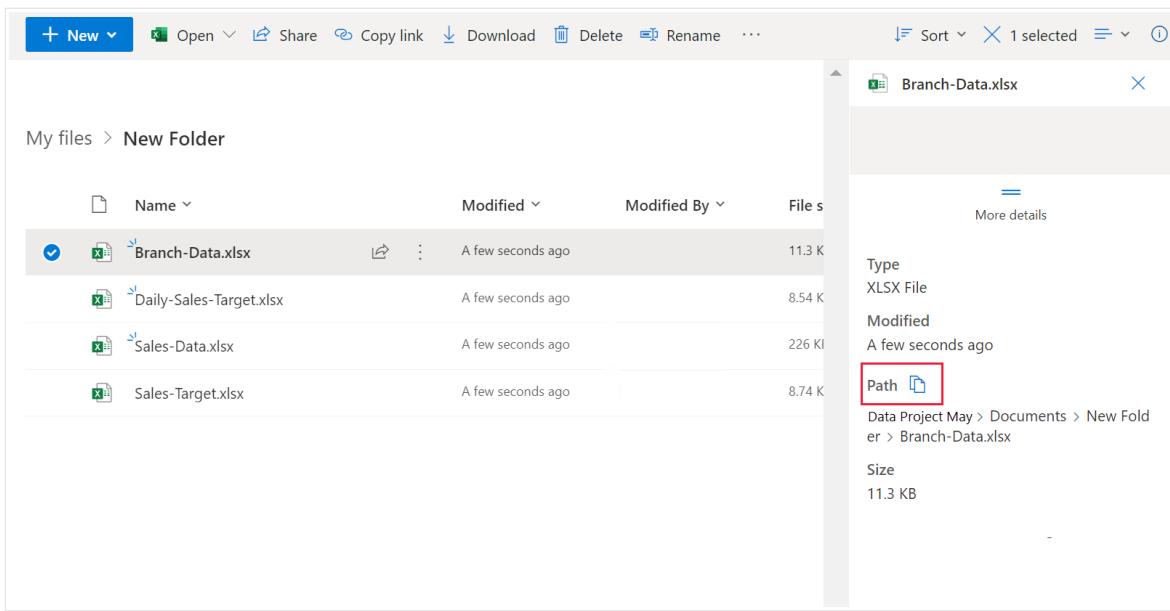
The screenshot shows a list of Excel files in a OneDrive folder. A context menu is open over the first file, "Branch-Data.xlsx". The menu items are: Open, Preview, Share, Copy link, Manage access, Download, Delete, Move to, Copy to, Rename, Automate, Version history, and Details. The "Details" item at the bottom of the menu is highlighted with a red box.

Name	Modified	Modified By
Branch-Data.xlsx	Open	>
Daily-Sales-Target.xlsx	Preview	
Sales-Data.xlsx	Share	
Sales-Target.xlsx	Copy link	
	Manage access	
	Download	
	Delete	
	Move to	
	Copy to	
	Rename	
	Automate >	
	Version history	
	Details	

**Note**

Your browser interface might not look exactly like this image. There are many ways to select **Open in Excel** for files in your OneDrive for work or school browser interface. You can use any option that allows you to open the file in Excel.

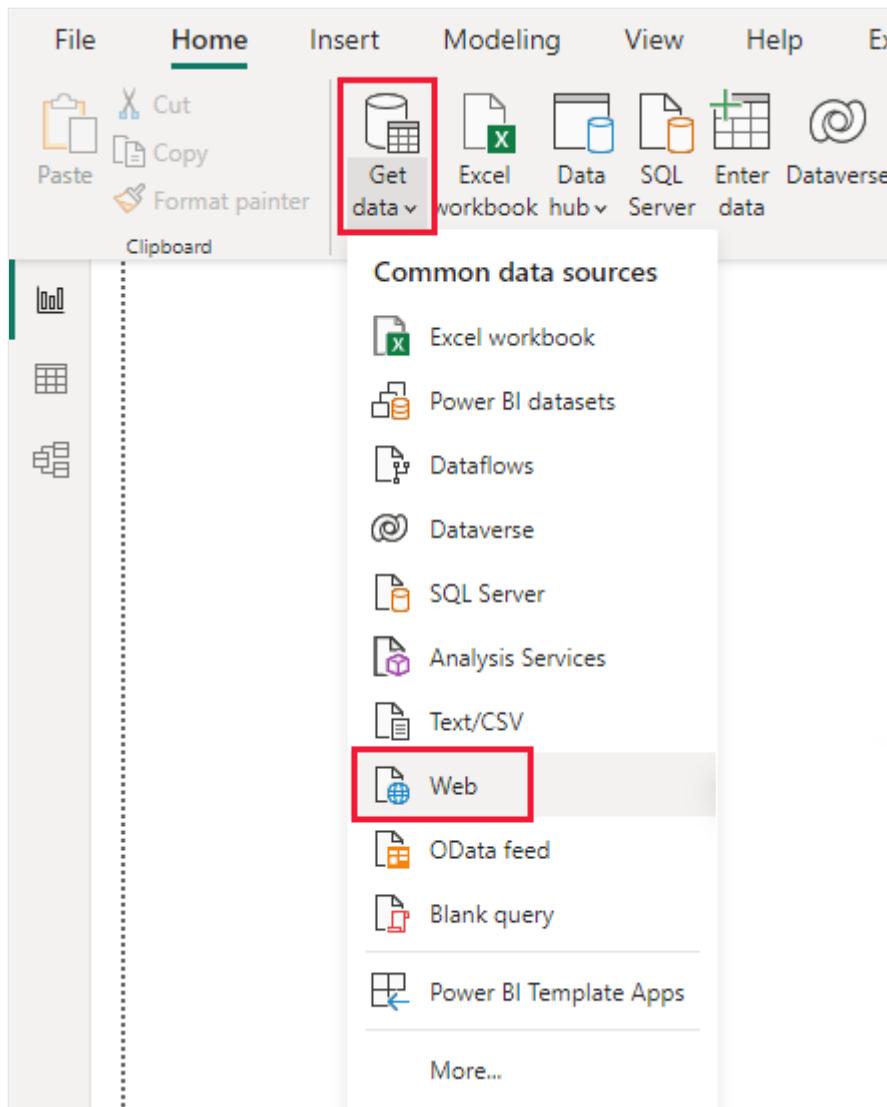
2. In the **More details** pane that appears, select the *copy* icon next to **Path**.



## Use the link in Power BI Desktop

In Power BI Desktop, you can use the link that you just copied to the clipboard. Take the following steps:

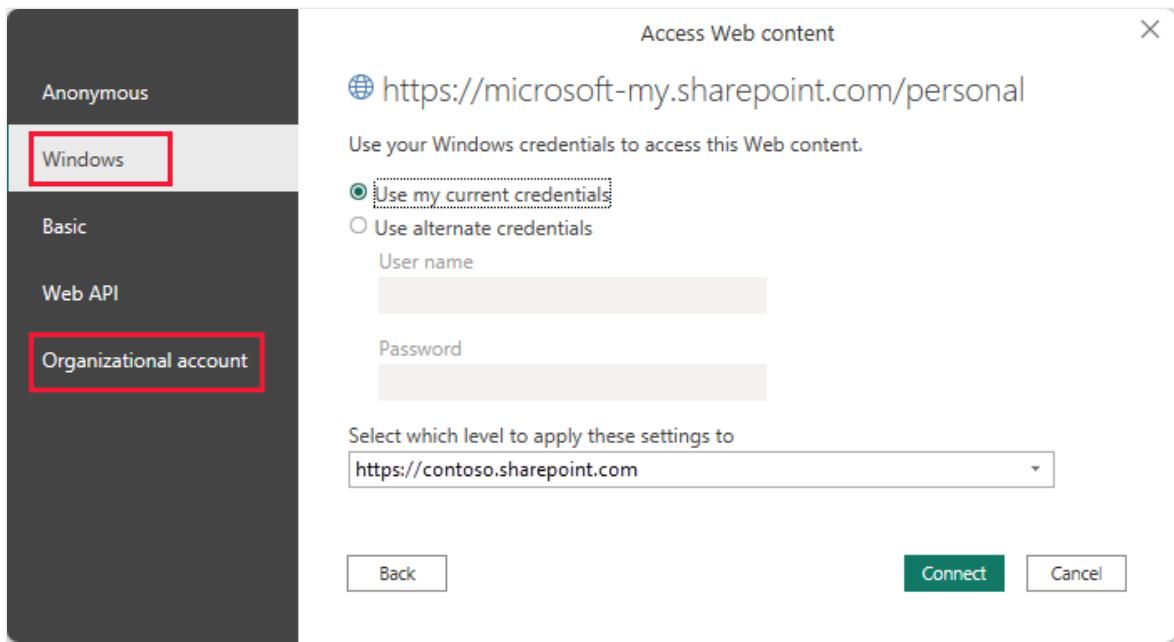
1. In Power BI Desktop, select **Get data > Web**.



2. With the **Basic** option selected, paste the link into the **From Web** dialog.



3. If Power BI Desktop prompts you for credentials, choose either **Windows** for on-premises SharePoint sites or **Organizational Account** for Microsoft 365 or OneDrive for work or school sites.



A **Navigator** dialog appears. It allows you to select from the list of tables, sheets, and ranges found in the Excel workbook. From there, you can use the OneDrive for work or school file just like any other Excel file. You can create reports and use it in datasets like you would with any other data source.

 **Note**

To use a OneDrive for work or school file as a data source in the Power BI service, with **Service Refresh** enabled for that file, make sure you select **OAuth2** as the **Authentication method** when you configure refresh settings. Otherwise, you might encounter an error when you attempt to connect or to refresh, such as, *Failed to update data source credentials*. Selecting **OAuth2** as the authentication method avoids that credentials error.

# Connect to Project Online data through Power BI Desktop

Article • 01/19/2023

You can connect to data in Project Online through Power BI Desktop.

## Step 1: Download Power BI Desktop

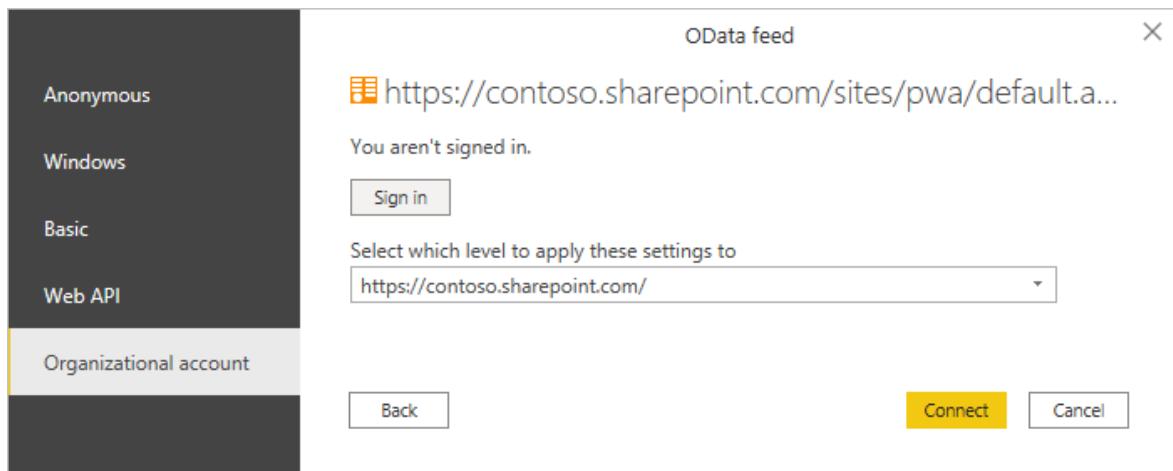
[Download Power BI Desktop](#), then run the installer to get Power BI Desktop on your computer.

## Step 2: Connect to Project Online with OData

1. Open Power BI Desktop.
2. On the *Welcome* screen, select **Get data**.
3. Select **OData Feed** and choose **Connect**.
4. Enter the address for your OData feed in the URL box, and then select **OK**.

If the address for your Project Web App site resembles <https://<tenantname>.sharepoint.com/sites/pwa>, then the address to enter for your OData Feed is [https://<tenantname>.sharepoint.com/sites/pwa/\\_api/Projectdata](https://<tenantname>.sharepoint.com/sites/pwa/_api/Projectdata).

5. Power BI Desktop prompts you to authenticate with your work or school account. Select **Organizational account** and then enter your credentials.



The account you use to connect to the OData feed must have at least Portfolio Viewer access to the Project Web App site.

From here, you can choose which tables you would like to connect to and build a query.

# Get data from files for Power BI

Article • 07/24/2023

In Power BI, you can connect to or import data and reports from these types of files:

- Microsoft Excel .xlsx and .xlsm files
- Power BI Desktop .pbix report files
- Comma-separated value (CSV) .csv files

## What it means to get data from a file

In Power BI, the data you explore comes from a dataset. To have a dataset, you need some data. This article focuses on getting data from files.

To better understand the importance of datasets and how to get data for them, consider an automobile. Sitting in your car and looking at the dashboard is like sitting in front of your computer looking at a dashboard in Power BI. The dashboard shows all the things your car is doing, like how fast the engine is revving, the temperature, what gear you're in, and your speed.

In Power BI, a dataset is like the engine in your car. The dataset provides the data, metrics, and information that's displayed in your Power BI dashboard. Your engine, or dataset, needs fuel, and data is the fuel in Power BI. Your car has a fuel tank that provides gas to the engine. Power BI also needs a fuel tank of data you can feed your dataset. That fuel tank can be a Power BI Desktop file, Excel workbook file, or CSV file.

To take it one step further, a fuel tank in a car has to be filled with gas. The gas for a Power BI Desktop, Excel, or CSV file is data from a data source that you put into the Excel, Power BI Desktop, or CSV file. You can manually enter rows of data into an Excel workbook or CSV file, or you can connect to the external data source to query and load data into your file. After you have a file that contains some data, you can get the file into Power BI as a dataset.

### Note

When you import Excel data into Power BI, the data must be in a table or data model.

## Where to save your file

Where you save your file makes a difference.

- **Local.** If you save your workbook file to a drive on your computer or another location in your organization, you can *import* your file into Power BI. Your file remains on the source drive. When you import the file, Power BI creates a new dataset in your site and loads your data, and in some cases your data model, into the dataset. Any reports in your file appear in **My workspace** as **Reports**.
- **OneDrive for work or school.** If you have OneDrive for work or school, sign in with the same account that you use for Power BI. This method is the most effective way to keep your work in Excel, Power BI Desktop, or CSV files in sync with your Power BI dataset, reports, and dashboards. Both Power BI and OneDrive are in the cloud, and Power BI connects to your file on OneDrive about once an hour. If Power BI finds any changes, it automatically updates your Power BI dataset, reports, and dashboards.

 **Note**

You can't upload files from personal OneDrive accounts, but you can upload files from your computer.

- **SharePoint team site.** Saving your Power BI Desktop files to a SharePoint team site is much like saving to OneDrive for work or school. The biggest difference is how you connect to the file from Power BI. You can specify a URL or connect to the root folder.

 **Note**

You can't update datasets imported from OneDrive for work or school from local files. For Power BI to update the dataset, you must replace the file in OneDrive for work or school. Alternatively, you can delete the dataset and its related items and then import again from a local file.

## Next steps

- [Get data from Excel workbook files](#)
- [Get data from Power BI Desktop files](#)
- [Get data from comma-separated value \(CSV\) files](#)

# Get data from Excel workbook files

Article • 08/28/2023

Microsoft Excel is one of the most widely used business applications and one of the most common data sources for Power BI.

## Supported workbooks

Power BI supports importing or connecting to workbooks created in Excel 2007 and later. Some features that this article describes are available only in later versions of Excel. Workbooks must be in the .xlsx or .xlsm file type and be smaller than 1 GB.

### Important

The following capabilities are deprecated and will no longer be available starting September 29th, 2023:

- Upload of local workbooks to Power BI workspaces will no longer be allowed.
- Configuring scheduling of refresh and refresh now for Excel files that don't already have scheduled refresh configured will no longer be allowed.

The following capabilities are deprecated and will no longer be available starting October 31, 2023:

- Scheduled refresh and refresh now for existing Excel files that were previously configured for scheduled refresh will no longer be allowed.
- Local workbooks uploaded to Power BI workspaces will no longer open in Power BI.

After October 31, 2023:

- You can download existing local workbooks from your Power BI workspace.
- You can publish your Excel data model as a Power BI dataset and schedule refresh.
- You can import Excel workbooks from OneDrive and SharePoint Document libraries to view them in Power BI.

If your organization uses these capabilities, see more details in [Migrating your Excel workbooks](#).

## Workbooks with ranges or tables of data

If your workbook contains simple worksheets with ranges of data, be sure to format those ranges as tables to get the most out of your data in Power BI. When you create reports in Power BI, the named tables and columns in the **Tables** pane make it much easier to visualize your data.

## Workbooks with data models

A workbook can contain a data model that has one or more tables of data loaded into it via linked tables, Power Query, **Get & Transform Data** in Excel, or Power Pivot. Power BI supports all data model properties, like relationships, measures, hierarchies, and key performance indicators (KPIs).

### Note

You can't share workbooks that contain data models across Power BI tenants. For example, a user who signs in to Power BI with a `contoso.com` account can't share a workbook containing data models with a user who signs in with a `woodgrovebank.com` account.

## Workbooks with connections to external data sources

If your Excel workbook connects to an external data source, after your workbook is in Power BI, you can create reports and dashboards based on data from that connected source. You can also set up scheduled refresh to automatically connect to the data source and get updates. You no longer need to refresh manually by using **Get Data** in Excel. Visualizations in reports and dashboard tiles that are based on the data source update automatically. For more information, see [Data refresh in Power BI](#).

## Workbooks with PivotTables and charts

Whether and how your PivotTables and charts appear in Power BI depends on where you save your workbook file, and how you choose to get the file into Power BI. The rest of this article explains the options.

## Data types

Assign data specific data types in Excel to improve your Power BI experience. Power BI supports these data types:

- Whole number
- Decimal number
- Currency
- Date
- True/false
- Text

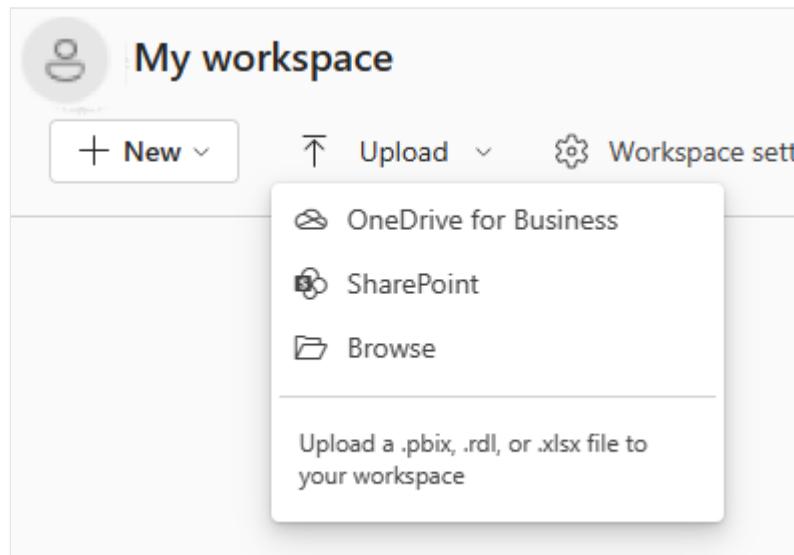
## Import or upload Excel data

There are two ways to explore Excel data in Power BI: upload and import. When you upload your workbook, it appears in Power BI just like it would in Excel Online. But you also have some great features to help you pin elements from your worksheets to your dashboards. When you import your data, Power BI imports any supported data in tables and any data model into a new Power BI dataset.

## Upload to Power BI

You can use the **Upload** button to upload files to the Power BI service. In the workspace where you want to add the file, select **Upload** at the top of the page. In the drop-down list, select:

- **OneDrive for Business** to connect to files that are stored in OneDrive for Business.
- **SharePoint** to connect to files on any SharePoint site that you have access to.
- **Browse** to upload files from your computer.



If you upload a local file, Power BI adds a copy of the file to the workspace. If you use the **OneDrive for Business** or **SharePoint** options, Power BI creates a connection to the file. As you make changes to the file in SharePoint or OneDrive, Power BI automatically syncs those changes about once an hour.

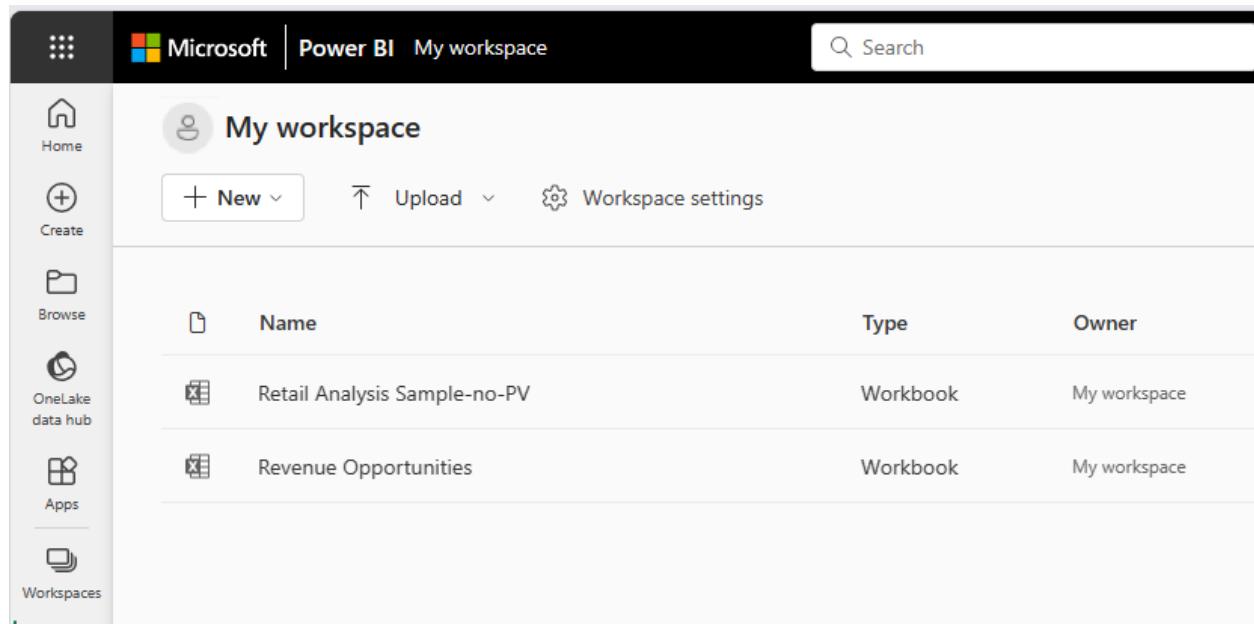
When you connect to an Excel file by using OneDrive for Business, you can't edit your workbook in Power BI. If you need to make changes, you can select **Edit** and then choose to edit your workbook in Excel Online or open it in Excel on your computer. Changes are saved to the workbook on OneDrive.

You should connect to or upload data if you have only data in worksheets, or if you have ranges, PivotTables, and charts that you want to pin to dashboards.

Local Excel workbooks open in Excel Online within Power BI. Unlike Excel workbooks stored on OneDrive or SharePoint team sites, you can't edit local Excel files within Power BI.

If you use Excel 2016 and later, you can also use **File > Publish > Upload** from Excel. For more information, see [Publish to Power BI from Microsoft Excel](#).

After your workbook uploads, it appears in the list of content in the workspace:

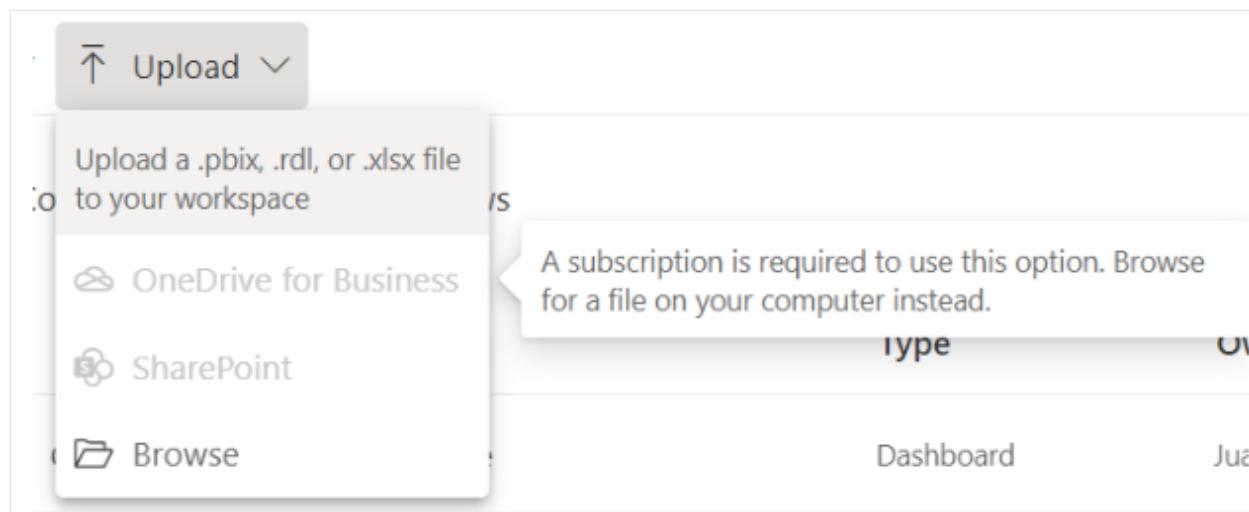


The screenshot shows the Microsoft Power BI web interface. At the top, there's a navigation bar with icons for Home, Create, Browse, OneLake data hub, Apps, and Workspaces. The 'Workspaces' icon is highlighted with a blue bar at the bottom. The main title is 'Microsoft Power BI My workspace'. Below the title, there are buttons for '+ New', 'Upload', and 'Workspace settings'. The main content area is titled 'My workspace' and lists two workbooks: 'Retail Analysis Sample-no-PV' and 'Revenue Opportunities'. A table provides details for each: Name, Type, and Owner. Both are listed as 'Workbook' type and owned by 'My workspace'. The 'Workspaces' menu item is also highlighted in blue at the bottom of the sidebar.

Name	Type	Owner
Retail Analysis Sample-no-PV	Workbook	My workspace
Revenue Opportunities	Workbook	My workspace

This upload method is easy to use, and the **OneDrive for Business** and **SharePoint** options use the same file selection interface as many other Microsoft products. Rather than entering a URL to a SharePoint or OneDrive location, you can select one of your sites by using the **Quick access** section or selecting **More places**.

If you don't have a subscription, the **OneDrive for Business** and **SharePoint** options are unavailable, but you can still select **Browse** to get local files from your computer. This image shows the unavailable options, but the **Browse** option is enabled:



You can't use **Upload** to get files from personal OneDrive accounts, but you can upload files from your computer.

## Import Excel data into Power BI

To import Excel data into Power BI, in **My workspace**, select **New > Dataset > Excel**, and then find the file.

The **My files** list allows you to add files from your documents folder and other personal sources.

You can use the **Quick access** list on the left side of the window to add files from SharePoint sites and other shared sources.

Select **Browse this device** to add files from the device you're currently using.

When you import Excel data, Power BI imports any supported data in tables and any data model into a new Power BI dataset.

You should import your data if you used **Get & Transform Data** or **Power Pivot** to load data into a data model.

If you upload from OneDrive for Business, when you save changes, Power BI synchronizes them with the dataset in Power BI, usually within about an hour. You can also select **Publish** to export your changes immediately. Any visualizations in reports and dashboards also update, based on the following refresh triggers:

Report tiles	Dashboard tiles
Open the report, after the cache expires.	Open the dashboard, after the cache refreshes.
Select <b>Refresh</b> in the report.	Select <b>Refresh</b> in the dashboard.

Report tiles	Dashboard tiles
	Automatically for pinned tiles when the cache refreshes, if the dashboard is already open.

 **Note**

Pinned report pages don't support the automatic refresh feature.

## Prepare your workbook for Power BI

Watch this video to learn more about how to make sure your Excel workbooks are ready for Power BI:

 **Note**

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

<https://www.youtube-nocookie.com/embed/I2wy4XgQlu0> ↗

## Where to save your workbook file

Where you save your workbook file makes a difference.

- **Local.** If you save your workbook file to a drive on your computer or another location in your organization, you can load your file into Power BI. Your file actually remains on the source drive. When you import the file, Power BI creates a new dataset and loads data and any data model from the workbook into the dataset.

Local Excel workbooks open in Excel Online within Power BI. Unlike Excel workbooks stored on OneDrive or SharePoint team sites, you can't edit local Excel files within Power BI.

Excel also has a **Publish** command on the **File** menu. Using this **Publish** command is effectively the same as using **Upload > Browse** from Power BI. If you regularly make changes to the workbook, it's often easier to update your dataset in Power BI.

- **OneDrive for Business.** Signing in to OneDrive for Business with the same account as Power BI is the most effective way to keep your work in Excel in sync with your Power BI dataset, reports, and dashboards. Both Power BI and OneDrive are in the

cloud, and Power BI connects to your workbook file on OneDrive about once an hour. If Power BI finds any changes, it automatically updates your Power BI dataset, reports, and dashboards.

As when you have a file saved to a local drive, you can use **Publish** in Excel to update your Power BI dataset and reports immediately. Otherwise, Power BI automatically synchronizes, usually within an hour.

- **SharePoint team site.** Saving your Power BI Desktop files to a SharePoint team site is almost the same as saving them to OneDrive for Business. The biggest difference is how you connect to the file from Power BI. You can specify a URL or connect to the root folder.

## Publish from Excel to your Power BI site

Using the Excel **Publish to Power BI** feature is effectively the same as using Power BI to import or connect to your file. For more information, see [Publish to Power BI from Microsoft Excel](#).

### ⓘ Note

If you upload an Excel workbook that's connected to an on-premises SQL Server Analysis Services (SSAS) cube, you can't refresh the underlying data model in the Power BI service.

## Migrating your Excel workbooks

For local Excel workbooks uploaded to a Power BI workspace, use the **Download Excel file** option to download the workbook. Then save it to OneDrive for Business or a SharePoint Document library (ODSP). You can then import the workbook from ODSP to the workspace again.

The screenshot shows the 'My workspace' page in Power BI. At the top, there's a user profile picture, the title 'My workspace', and navigation buttons for '+ New', 'Upload', and 'Workspace settings'. Below this is a table listing workbooks. The first entry is 'Workbook with Model', which has a context menu open over it. The menu includes options like 'Refresh now', 'Replace file', 'Settings', 'Delete', and 'Download Excel file'. The 'Download Excel file' option is highlighted with a red box and a cursor icon pointing to it.

	Name	Type	Owner
	Workbook with Model	Workbook	My workspace

To refresh data in Excel data models, you'll need to publish the data model as a Power BI dataset. We recommend using the Power BI Desktop to import the model because it upgrades your data model to the latest version. This gives you the best future experience. Use the **Import from Power Query**, **Power Pivot**, **Power View** option on Power BI Desktop's **File** menu.

To build new workbooks connected to a semantic data model in your Excel workbook, you should first publish the data model as a Power BI dataset. Then in Excel use the **From Power BI (Microsoft)** option to connect your workbook to the dataset. This option is available in the **Data ribbon**, under **Get Data** in the **From Power Platform** menu.

For cases where you include a workbook in a Power BI organizational app, remember to republish the app with the new items.

To learn which workbooks can be affected by the deprecation of local workbooks and refresh capabilities, use the **workbooks** Power BI admin REST API. It lists the workbooks in your organization. You must be a member of the Power BI admin role or a Global Administrator to call this API.

```
GET https://api.powerbi.com/v1.0/myorg/admin/workbooks
```

The API provides a list of all the Excel workbooks published in your organization. The list is formatted in JSON.

Below is an example output for the API.

```
[
 {
 "DisplayName": "Workbook without a Data Model",
 "WorkspaceName": "My workspace",
 "HasDataModel": false,
 "HasScheduledRefreshOnDataModel": false,
 "UploadedOn": "2023-07-28T10:54:17.093"
 },
 {
 "DisplayName": "Workbook with Data Model",
 "WorkspaceName": "My workspace",
 "HasDataModel": true,
 "HasScheduledRefreshOnDataModel": true,
 "UploadedBy": "user@contoso.com",
 "UploadedOn": "2022-11-16T09:51:17.497"
 }
]
```

You can check if the Excel workbook is a local workbook by navigating to it in Power BI and seeing if it has the Download Excel file option is available.

You can use PowerShell to call the API as shown in the example below:

```
Invoke-PowerBIRestMethod -Url
"https://api.powerbi.com/v1.0/myorg/admin/workbooks" -Method GET
```

To use PowerShell, first install the required **MicrosoftPowerBIMgmt** module. See [Power BI Cmdlets reference](#) for details. You will need to call **Login-PowerBIServiceAccount** commandlet before calling **Invoke-PowerBIRestMethod**.

## Troubleshooting and limitations

- If your workbook file is too large, see [Reduce the size of an Excel workbook to view it in Power BI](#).
- The upload of Excel workbooks to a Power BI workspace isn't supported for sovereign cloud customers.
- You can't use scheduled refresh for Excel workbooks that have connections to on-premises SSAS tabular models through a gateway.

# Next steps

- **Explore your data.** After you upload data and reports from your file into Power BI, you can select the new dataset to explore the data. When you select the workbook, it opens in Power BI the same as if it were in Excel Online.
- **Schedule refresh.** If your Excel workbook connects to external data sources, or if you imported from a local drive, you can set up scheduled refresh to make sure your dataset or report is always up-to-date. In most cases, setting up scheduled refresh is easy to do. For more information, see [Data refresh in Power BI](#).
- **Publish to Power BI from Microsoft Excel.**

# Get data from Power BI Desktop files

Article • 02/13/2023



Power BI Desktop makes business intelligence and reporting easy. Whether you're connecting to many different data sources, querying and transforming data, modeling your data, and creating powerful and dynamic reports, Power BI Desktop makes business intelligence tasks intuitive and fast. If you're not familiar with Power BI Desktop, check out [Getting started with Power BI Desktop](#).

Once you bring data into Power BI Desktop and create a few reports, it's time to get your saved file into the Power BI service.

## Where your file is saved makes a difference

There are several locations where you might store Power BI Desktop files:

- **Local.** If you save your file to a local drive on your computer or another location in your organization, you can *import* your file, or you can *publish* from Power BI Desktop to get its data and reports into the Power BI service.

Your file remains on your local drive. The whole file isn't moved into Power BI. A new dataset is created in Power BI and data and the data model from the Power BI Desktop file are loaded into the dataset. If your file has any reports, those reports appear in your Power BI service site under **Reports**.

- **OneDrive for work or school.** By far, the most effective way to keep your work in Power BI Desktop in sync with the Power BI service is to use your OneDrive for work or school and sign in with the same account as the Power BI service. Your work includes dataset, reports, and dashboards. Because both the Power BI service and OneDrive are in the cloud, Power BI *connects* to your file on OneDrive about every hour. If it finds any changes, your dataset, reports, and dashboards are updated in the Power BI service.
- **OneDrive - Personal.** If you save your files to your own OneDrive account, you get many of the same benefits as you would with OneDrive for work or school. The biggest difference is when you first connect to your file, you need to sign in to

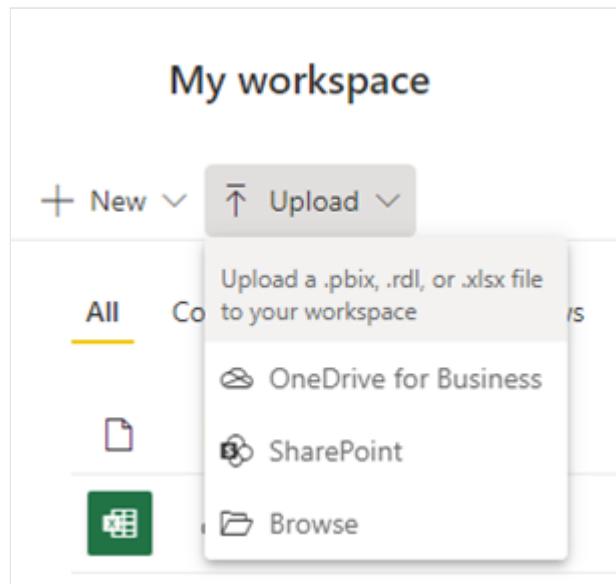
your OneDrive with your Microsoft account. This account is usually different from what you use to sign in to the Power BI service.

When signing in with your OneDrive with your Microsoft account, be sure to select the **Keep me signed in** option. This way, the Power BI service can connect to your file about every hour and make sure that your dataset in the Power BI service is in-sync.

- **SharePoint Team-Sites.** Saving your Power BI Desktop files to SharePoint – Team Sites is much the same as saving to OneDrive for work or school. The biggest difference is how you connect to the file from the Power BI service. You can specify a URL or connect to the root folder. You can also [set up a Sync folder](#) that points to the SharePoint folder. Files in that folder sync up with the ones on SharePoint.

## Streamlined upload to Power BI

Beginning in November 2022, there's a new and streamlined experience for uploading files to the Power BI service. In the workspace into which you want to add files, you see an **Upload** dropdown menu option next to the **New** button. You can use the dropdown menu to connect to files stored in *OneDrive for work or school* or any *SharePoint* site to which you have access, or you can upload them from your computer through the *Browse* menu option. The following image shows the menu options.

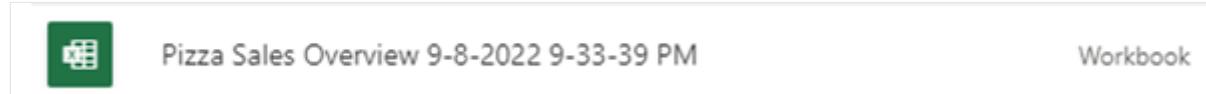


If you choose to upload a local file, a copy of the file is added to the workspace. If you use the *OneDrive for work or school* or *SharePoint* option, the Power BI service creates a connection to the file and as you make changes to the file in SharePoint, Power BI can automatically sync those changes approximately each hour.

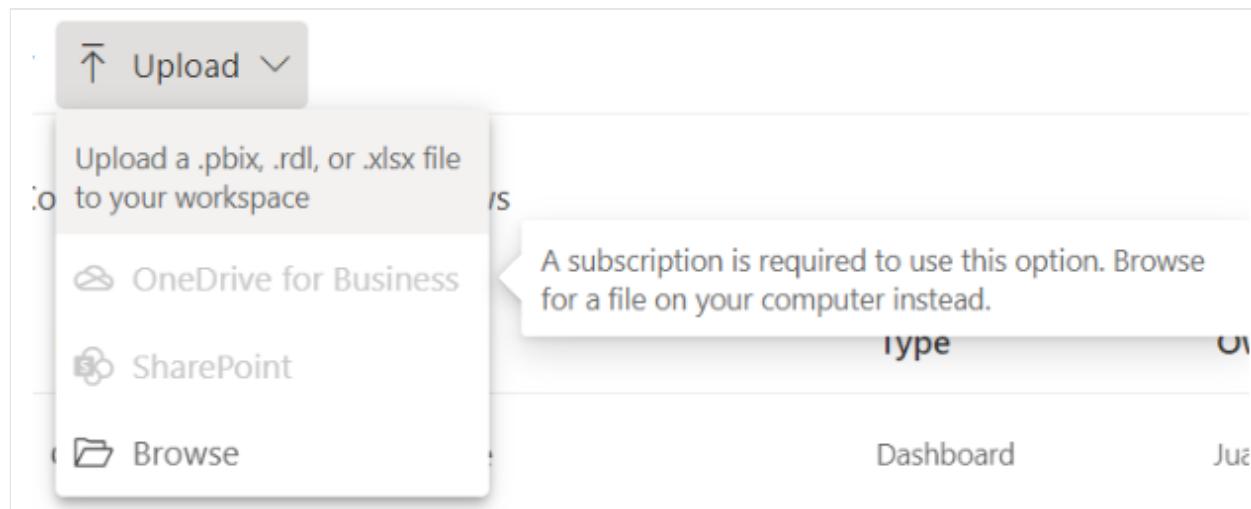
A benefit of uploading files this way, in addition to being easy to use, is that the *OneDrive for work or school* and *SharePoint* options use the same file selection interface used in many other Microsoft products.

Rather than having to paste a direct URL to a given SharePoint site, which was previously required, you can now simply select one of your sites through the *Quick access* section or the *More places* links.

When you upload an Excel file this way, your workbook appears in the Power BI service just like it would in Excel Online, as shown in the following image.



If you don't have a subscription, *OneDrive for work or school* and *SharePoint* options are disabled, but you can still *browse* for local files on your computer. The following image shows the subscription options disabled, with the *Browse* option highlighted.



#### ⓘ Note

You can't upload files from SharePoint Document set folder or from personal OneDrive accounts.

## Publish a file from Power BI Desktop to the Power BI service

Using **Publish** from Power BI Desktop is similar to uploading files in the Power BI service. Both initially import your file data from a local drive or connect to it on OneDrive. However, there are differences. If you upload from a local drive, refresh that data frequently to ensure the online and local copies of the data are current with each other.

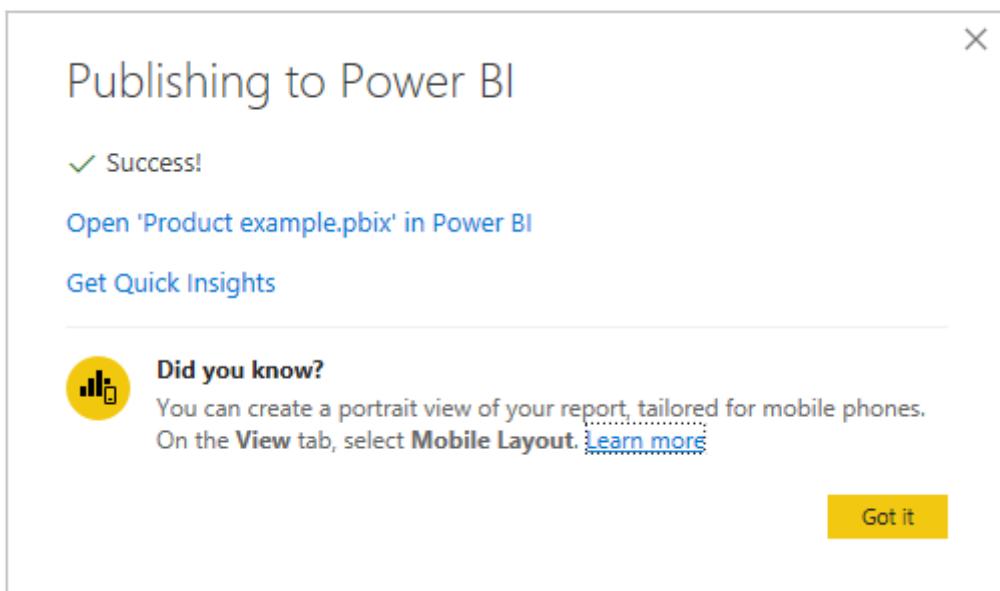
Here's the quick how to, but you can see [Publish from Power BI Desktop](#) to learn more.

1. In Power BI Desktop, select **File > Publish > Publish to Power BI**, or select **Publish** on the ribbon.



2. Sign in to the Power BI service. You only need to sign in the first time.

When complete, you get a link to open your report in your Power BI site.



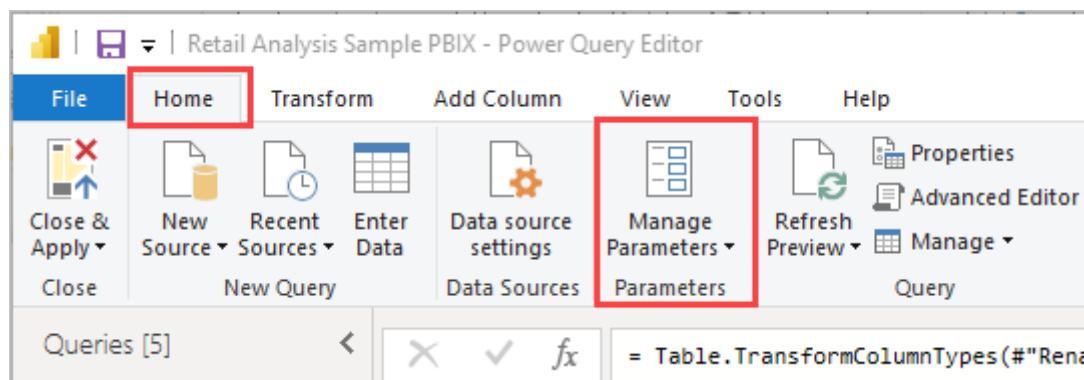
## Next steps

- **Explore your data:** Once you get data and reports from your file into the Power BI service, it's time to explore. If your file already has reports in it, they appear in the navigator pane in **Reports**. If your file just had data, you can create new reports; just right-click the new dataset and then select **Explore**.
- **Refresh external data sources:** If your Power BI Desktop file connects to external data sources, you can set up scheduled refresh to make sure your dataset is always up-to-date. In most cases, setting up scheduled refresh is easy to do, but going into the details is outside the scope of this article. See [Data refresh in Power BI](#) to learn more.

# Edit parameter settings in the Power BI service

Article • 04/14/2023

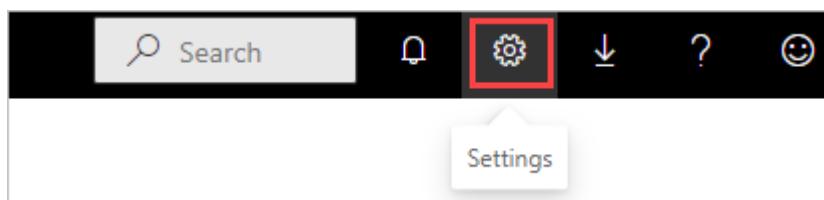
Report creators add query parameters to reports in Power BI Desktop. Parameters allow you to make parts of reports depend on one or more parameter *values*. For example, a report creator might create a parameter that restricts the data to a single country/region, or a parameter that defines acceptable formats for fields like dates, time, and text.



## Review and edit parameters in Power BI service

As a report creator, you define parameters in Power BI Desktop. When you [publish that report to the Power BI service](#), the parameter settings and selections travel with it. You can review and edit parameter settings in the Power BI service, but not create them.

1. In the Power BI service, select the cog icon and then choose **Settings**.



2. Select the tab for **Datasets** and highlight a dataset in the list.

The screenshot shows the Power BI Datasets page. At the top, there are navigation links: General, Alerts, Subscriptions, Dashboards, **Datasets**, Workbooks, and Reports. The 'Datasets' link is highlighted with a red box. On the left, a sidebar lists datasets: AdventureWorks (selected), Customer Profitability Sample, Financial Sample, IT Spend Analysis Sample, Opportunity Analysis Sample, Retail Analysis Sample, Sales & Returns, Sales & Returns Sample v201912, and Sales and Marketing Sample PBIX current. To the right, under 'Settings for AdventureWorks', there are several expandable sections: View dataset, Refresh history, Dataset description, Gateway connection, Data source credentials, Sensitivity label, Parameters, and Scheduled refresh.

3. Expand **Parameters**. If the selected dataset has no parameters, you see a message with a link to **Learn more** about query parameters. If the dataset does have parameters, expand the **Parameters** heading to reveal those parameters.

The screenshot shows the Power BI Datasets page with the 'Datasets' tab selected. The sidebar lists the same datasets as before. On the right, under 'Settings for AdventureWorks', the 'Parameters' section is expanded, indicated by a red box. It contains two entries: 'CountryRegion' (Defines country or region) with a dropdown menu set to 'All', and 'DateFormat' (Define the format for dates) with a dropdown menu set to 'Short'. At the bottom right of the expanded section are 'Apply' and 'Discard' buttons.

4. Review the parameter settings and make changes if needed.

# Considerations and limitations

Grayed-out fields aren't editable. *Any* and *Binary* type parameters work in Power BI Desktop. The Power BI service doesn't currently support them for security reasons.

## Next steps

An ad-hoc way to add simple parameters is by [modifying filters](#).

# Get data from comma separated value (CSV) files

Article • 01/27/2023



Comma separated value files, often known as a CSV, are simple text files with rows of data where each value is separated by a comma. These types of files can contain large amounts of data within a relatively small file size, making them an ideal data source for Power BI. You can download a [sample CSV file](#).

If you have a CSV, it's time to get it into your Power BI site as a dataset where you can begin exploring your data, create some dashboards, and share your insights with others.

## Tip

Many organizations output a CSV with updated data each day. To make sure your dataset in Power BI stays in-sync with your updated file, be sure the file is saved to OneDrive with the same name.

## Where your file is saved makes a difference

**Local** - If you save your CSV file to a local drive on your computer or another location in your organization, you can *import* your file into Power BI. Your file will actually remain on your local drive, so the whole file isn't imported into Power BI. What really happens is a new dataset is created in Power BI and data from the CSV file is loaded into the dataset.

**OneDrive for work or school** – If you have OneDrive for work or school and you sign into it with the same account you use to sign into Power BI, this method is the most effective way to keep your CSV file and your dataset, reports, and dashboards in Power BI in-sync. Because both Power BI and OneDrive are in the cloud, Power BI *connects* to your file on OneDrive about every hour. If any changes are found, your dataset, reports, and dashboards are automatically updated in Power BI.

**OneDrive - Personal** – If you save your files to your own OneDrive account, you'll get many of the same benefits as you would with OneDrive for work or school. The biggest difference is when you first connect to your file you'll need to sign in to your OneDrive

with your Microsoft account, which is different from what you use to sign in to Power BI. When signing into your OneDrive with your Microsoft account, be sure to select the **Keep me signed in** option. This way, Power BI will be able to connect to your file about every hour and make sure your dataset in Power BI is in-sync.

**SharePoint** – Saving your Power BI Desktop files to SharePoint is much the same as saving to OneDrive for work or school. The biggest difference is how you connect to the file from Power BI. You can specify a URL or connect to the root folder.

## Import or connect to a CSV file

### **Important**

The maximum file size you can import into Power BI is 1 GB.

1. In a Power BI workspace, select **+ New**, and choose **Upload a file**.



Home



Create



Browse



Data hub



Metrics



Apps

Deployment  
pipelines

Learn



Workspaces

my-test-  
workspace

## my-test-workspace

Test

+ New

Upload

Create deplo...

Report

Visualize your data

Paginated report

Build a paginated report

Scorecard

Track related metrics together

Dashboard

Build a single-page data story

Dataset

Create a dataset to use in a report

Dataflow

Prep, clean, and transform data

Datamart (Preview)

Build a business-focused datamart

Streaming dataset

Build visuals from real-time data

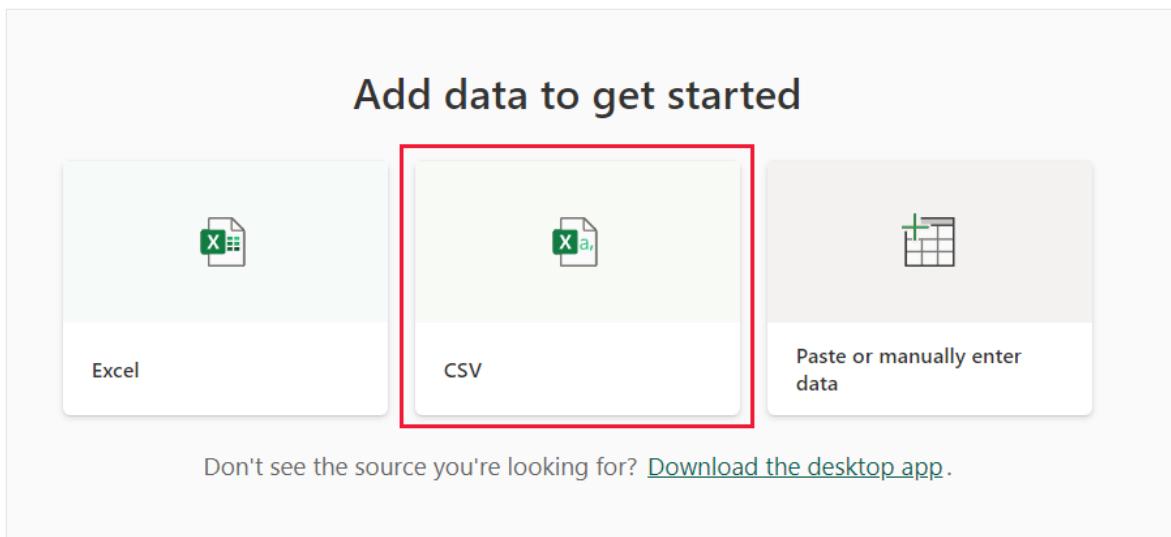
Streaming dataflow

Combine and transform streaming ...

Upload a file

Open a .pbix, .rdl, .xlsx, or .csv in Po...

## 2. Select CSV.



3. Go to the file you want to upload and then choose **Import**. A new **Dataset details** window appears in the main pane of Power BI.

The screenshot shows the 'Dataset details' page for a dataset named 'Financial Sample'. The left sidebar includes options like Home, Create, Browse, Data hub, Metrics, Apps, Deployment pipelines, Learn, and Workspaces. The main area displays dataset details: Workspace (my-test-workspace), Refreshed (1/26/23, 5:05:38 PM), and Sensitivity (Confidential/Microsoft Extended). It also has sections for Visualize this data and Share this data. The right side features a 'Tables' pane listing tables such as Financial Sample, Segment, Country, Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Gross Sales, Discounts, and Sales.

## Next steps

**Explore your data** - Once you get data from your file into Power BI, it's time to explore. Select **More options (...)**, and then choose an option from the menu.

The screenshot shows the Power BI workspace interface. At the top, there's a circular profile icon with three people, followed by the workspace name "my-test-workspace" and a diamond icon. Below the name, the word "Test" is displayed. On the left, there are navigation links: "+ New", "Upload", and "Create deployment pipeline". Below these are tabs: "All" (which is underlined in green), "Content", "Datasets + dataflows", and "Datamarts (Preview)".

The main area displays a table with four rows:

	Name	Type
	FabrikamComments	Dataset
	Financial Sample	Dataset
	sample-comments	Analyze in Excel

A context menu is open over the third row ("sample-comments"). The menu items are:

- ... (highlighted with a red box)
- Analyze in Excel
- Create report
- Auto-create report
- Create paginated report
- Create formatted table
- Delete
- Get quick insights
- Rename
- Settings
- Manage permissions
- View lineage

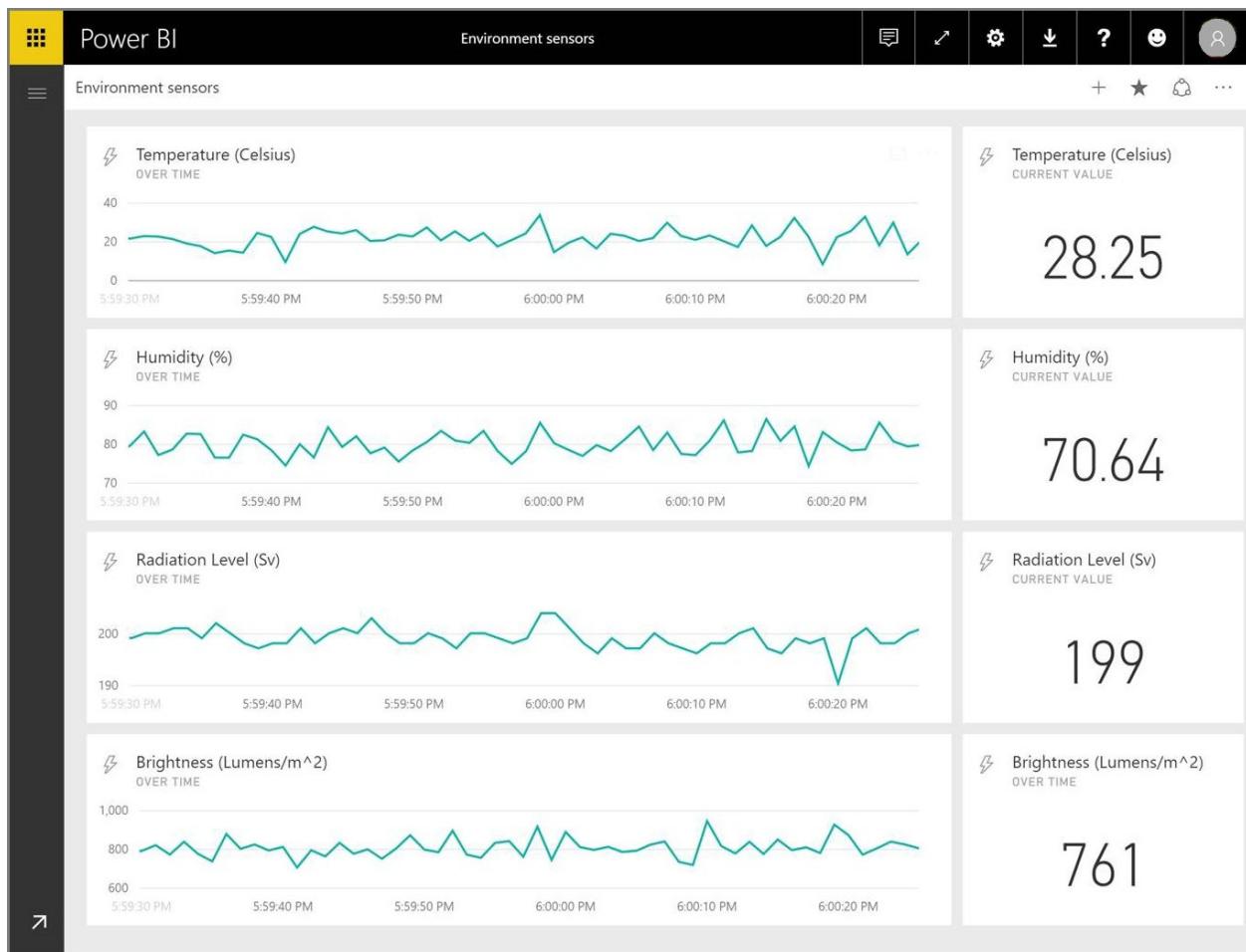
**Schedule refresh** - If your file is saved to a local drive, you can schedule refreshes so your dataset and reports in Power BI stay up-to-date. To learn more, see [Data refresh in Power BI](#). If your file is saved to OneDrive, Power BI will automatically synchronize with it about every hour.

# Real-time streaming in Power BI

Article • 01/12/2023

Power BI with real-time streaming helps you stream data and update dashboards in real time. Any visual or dashboard created in Power BI can display and update real-time data and visuals. The devices and sources of streaming data can be factory sensors, social media sources, service usage metrics, or many other time-sensitive data collectors or transmitters.

This article shows you how to set up and use real-time streaming datasets in Power BI.



## Types of real-time datasets

First, it's important to understand the types of real-time datasets that are designed to display in tiles and dashboards, and how those datasets differ.

The following three types of real-time datasets are designed for display on real-time dashboards:

- Push dataset
- Streaming dataset

- PubNub streaming dataset

This section explains how these datasets differ from one another. Later sections describe how to push data into each of these datasets.

## Push dataset

With a *push dataset*, data is pushed into the Power BI service. When the dataset is created, the Power BI service automatically creates a new database in the service to store the data.

Because there's an underlying database that stores the data as it arrives, you can create reports with the data. These reports and their visuals are just like any other report visuals. You can use all of Power BI's report building features, such as Power BI visuals, data alerts, and pinned dashboard tiles.

Once you create a report using the push dataset, you can pin any of the report visuals to a dashboard. On that dashboard, visuals update in real time whenever the data is updated. Within the Power BI service, the dashboard triggers a tile refresh every time new data is received.

There are two considerations to note about pinned tiles from a push dataset:

- Pinning an entire report by using the **Pin live** option won't result in the data automatically being updated.
- Once you pin a visual to a dashboard, you can use **Q&A** to ask questions about the push dataset in natural language. After you make a **Q&A** query, you can pin the resulting visual back to the dashboard, and that visual will also update in real time.

## Streaming dataset

A *streaming dataset* also pushes data into the Power BI service, with an important difference: Power BI stores the data only into a temporary cache, which quickly expires. The temporary cache is used only to display visuals that have some transient history, such as a line chart that has a time window of one hour.

A streaming dataset has no underlying database, so you can't build report visuals by using the data that flows in from the stream. Therefore, you can't use report functionality such as filtering, Power BI visuals, and other report functions.

The only way to visualize a streaming dataset is to add a tile and use the streaming dataset as a *custom streaming data* source. The custom streaming tiles that are based on a streaming dataset are optimized for quickly displaying real-time data. There's little

latency between pushing the data into the Power BI service and updating the visual, because there's no need for the data to be entered into or read from a database.

In practice, it's best to use streaming datasets and their accompanying streaming visuals in situations when it's critical to minimize the latency between pushing and visualizing data. You should have the data pushed in a format that can be visualized as-is, without any more aggregations. Examples of data that's ready as-is include temperatures and pre-calculated averages.

## PubNub streaming dataset

With a *PubNub streaming dataset*, the Power BI web client uses the [PubNub SDK](#) to read an existing PubNub data stream. The Power BI service stores no data. Because the web client makes this call directly, if you allow only approved outbound traffic from your network, you must list traffic to PubNub as allowed. For instructions, see the [support article about approving outbound traffic for PubNub](#).

As with the streaming dataset, with the PubNub streaming dataset there's no underlying Power BI database. You can't build report visuals against the data that flows in, and can't use report functionality like filtering or Power BI visuals. You can visualize a PubNub streaming dataset only by adding a tile to the dashboard and configuring a PubNub data stream as the source.

Tiles based on a PubNub streaming dataset are optimized for quickly displaying real-time data. Since Power BI is directly connected to the PubNub data stream, there's little latency between pushing the data into the Power BI service and updating the visual.

## Streaming dataset matrix

The following table describes the three types of datasets for real-time streaming and lists their capabilities and limitations.

Capability	Push	Streaming	PubNub
Dashboard tiles update in real time as data is pushed in	Yes. For visuals built via reports and then pinned to dashboard.	Yes. For custom streaming tiles added directly to the dashboard.	Yes. For custom streaming tiles added directly to the dashboard.
Dashboard tiles update with smooth animations	No.	Yes.	Yes.

Capability	Push	Streaming	PubNub
Data stored permanently in Power BI for historic analysis	Yes.	No. Data is temporarily stored for one hour to render visuals.	No.
Build Power BI reports on top of the data	Yes.	No.	No.
Max rate of data ingestion	1 requests 16 MB/request	5 requests 15 KB/request	N/A Data isn't being pushed into Power BI.
Limits on data throughput	1M rows/hour	None.	N/A Data isn't being pushed into Power BI.

## Push data to datasets

This section describes how to create and push data into the three primary types of real-time datasets that you can use in real-time streaming.

You can push data into a dataset by using the following methods:

- The Power BI REST APIs
- The Power BI streaming dataset UI
- Azure Stream Analytics

## Use Power BI REST APIs to push data

You can use *Power BI REST APIs* to create and send data to push datasets and to streaming datasets. When you create a dataset by using Power BI REST APIs, the `defaultMode` flag specifies whether the dataset is push or streaming.

If no `defaultMode` flag is set, the dataset defaults to a push dataset. If the `defaultMode` value is set to `pushStreaming`, the dataset is both a push and streaming dataset, and provides the benefits of both dataset types.

### Note

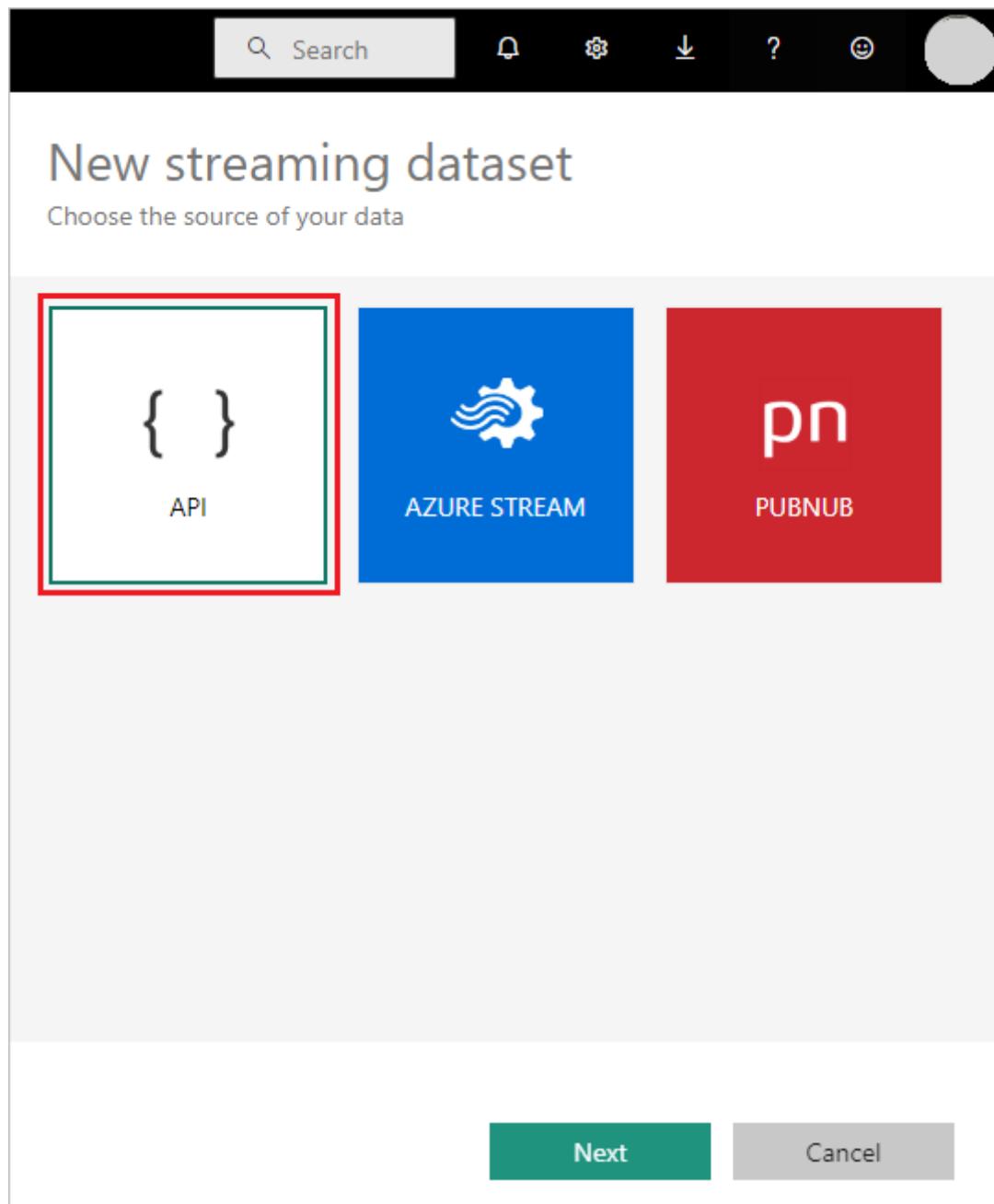
When you use datasets with the `defaultMode` flag set to `pushStreaming`, if a request exceeds the 15 KB size restriction for a streaming dataset, but is less than the 16

MB size restriction for a push dataset, the request succeeds and the data updates in the push dataset. However, any streaming tiles temporarily fail.

Once a dataset is created, you can use the [PostRows](#) REST APIs to push data. All requests to REST APIs are secured by using *Azure Active Directory (Azure AD) OAuth*.

## Use the streaming dataset UI to push data

In the Power BI service, you can create a dataset by selecting the **API** approach, as shown in the following screenshot:



When you create the new streaming dataset, you can enable **Historic data analysis**, as shown in the following screenshot. This selection has a significant impact.

# New streaming dataset

Create a streaming dataset and integrate our API into your device or application to send data. [Learn more about the API.](#)

Dataset name \*

What do you want to name your dataset?

Values from stream \*

Enter a new value name

Text



Historic data analysis



Back

Create

Cancel

When **Historic data analysis** is disabled, as it is by default, you create a streaming dataset as described earlier. When **Historic data analysis** is enabled, the dataset you create becomes both a streaming dataset and a push dataset. This setting is equivalent to using the Power BI REST APIs to create a dataset with its `defaultMode` set to `pushStreaming`, as described earlier.

## ⓘ Note

Streaming datasets created by using the Power BI service UI don't require Azure AD authentication. In such datasets, the dataset owner receives a URL with a *rowkey*, which authorizes the requestor to push data into the dataset without using an Azure AD OAuth bearer token. However, the Azure AD approach still works to push data into the dataset.

## Use Azure Stream Analytics to push data

You can add Power BI as an output within Azure Stream Analytics, and then visualize those data streams in the Power BI service in real time. This section describes the technical details of that process.

Azure Stream Analytics uses the Power BI REST APIs to create its output data stream to Power BI, with `defaultMode` set to `pushStreaming`. The resulting dataset can use both push and streaming. When you create the dataset, Azure Stream Analytics sets the `retentionPolicy` flag to `basicFIFO`. With that setting, the database that supports the push dataset stores 200,000 rows, and drops rows in a first-in first-out (FIFO) fashion.

 **Important**

If your Azure Stream Analytics query results in very rapid output to Power BI, for example once or twice per second, Azure Stream Analytics begins batching the outputs into a single request. This batching might cause the request size to exceed the streaming tile limit, and streaming tiles might fail to render. In this case, the best practice is to slow the rate of data output to Power BI. For example, instead of a maximum value every second, request a maximum value over 10 seconds.

## Set up your real-time streaming dataset in Power BI

To get started with real-time streaming, you choose one of the following ways to consume streaming data in Power BI:

- *Tiles* with visuals from streaming data
- *Datasets* created from streaming data that persist in Power BI

For either option, you need to set up streaming data in Power BI. To get your real-time streaming dataset working in Power BI:

1. In either an existing or new dashboard, select **Add a tile**.
2. On the **Add a tile** page, select **Custom Streaming Data**, and then select **Next**.

## Add a tile

### Select source

#### MEDIA



Web content



Image



Text box



Video

#### REAL-TIME DATA



Custom Streaming  
Data

Next

Cancel

3. On the **Add a custom streaming data tile** page, you can select an existing dataset, or select **Manage datasets** to import your streaming dataset if you already created one. If you don't have streaming data set up yet, select **Add streaming dataset** to get started.

# Add a custom streaming data tile

Choose a streaming dataset

+ Add streaming dataset

## YOUR DATASETS

weather forecast

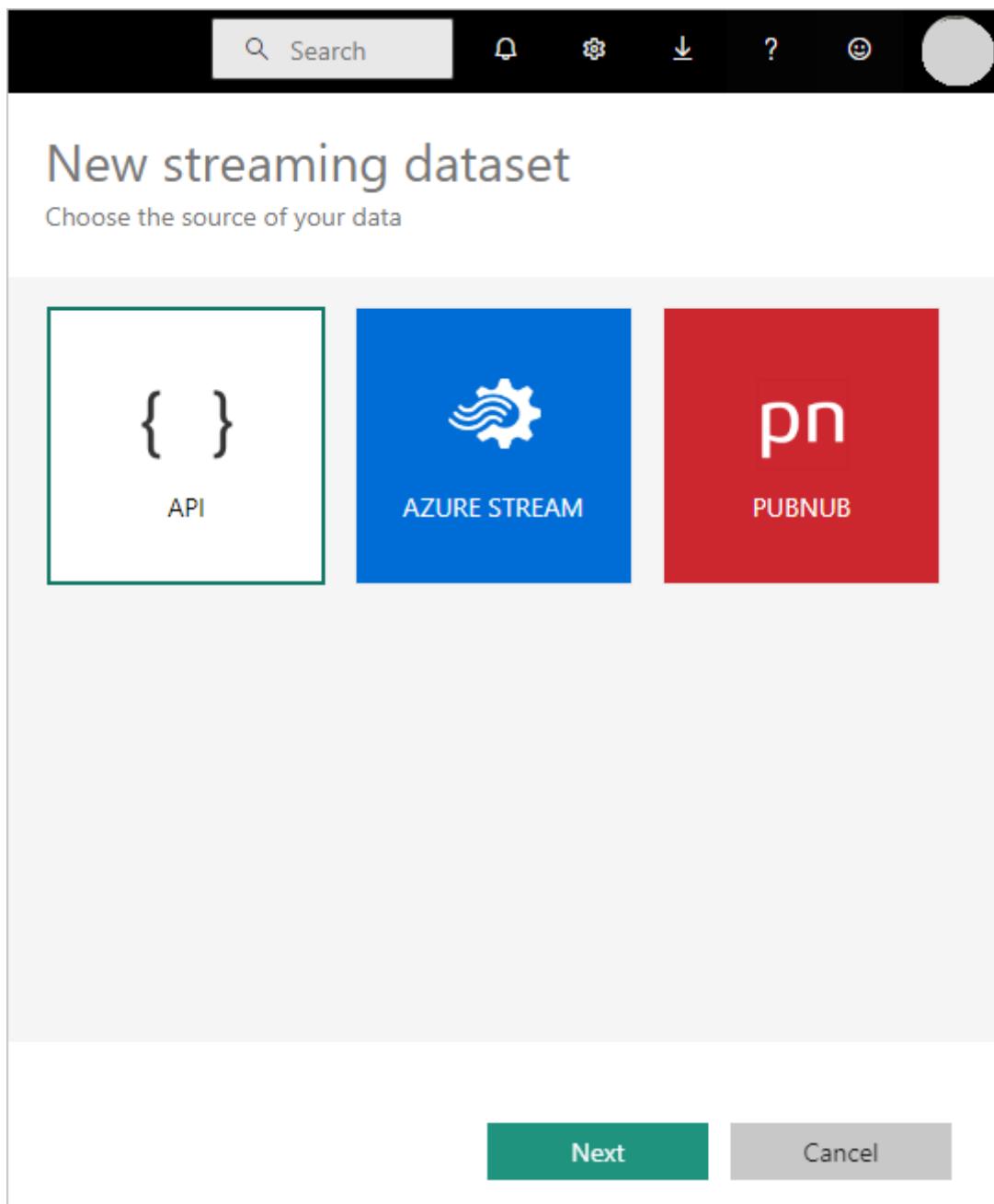
[Manage datasets](#)

[Back](#)

[Next](#)

[Cancel](#)

4. On the **New streaming dataset** page, select **API**, **Azure Stream**, or **PubNub**, and then select **Next**.



## Create a streaming dataset

There are three ways to create a real-time streaming data feed that Power BI can consume and visualize:

- Power BI REST API using a real-time streaming endpoint
- Azure Stream
- PubNub

This section describes the Power BI REST API and PubNub options, and explains how to create a streaming tile or dataset from the streaming data source. You can then use the dataset to build reports. For more information about the Azure Stream option, see [Power BI output from Azure Stream Analytics](#).

# Use the Power BI REST API

The Power BI REST API makes real-time streaming easier for developers. After you select API on the **New streaming dataset** screen and select **Next**, you can provide entries that enable Power BI to connect to and use your endpoint. For more information about the API, see [Use the Power BI REST APIs](#).

The screenshot shows the 'New streaming dataset' configuration screen. At the top, it says 'New streaming dataset'. Below that, a descriptive text reads: 'Create a streaming dataset and integrate our API into your device or application to send data. [Learn more about the API.](#)'

There are three main input fields:

- Dataset name \***: A text input field containing the placeholder 'What do you want to name your dataset?'
- Values from stream \***: A row with two inputs. The first is a text input field 'Enter a new value name' and the second is a dropdown menu set to 'Text'.
- Historic data analysis**: A toggle switch that is currently off.

At the bottom, there are three buttons: 'Back', 'Create', and 'Cancel'.

If you want Power BI to store the data this data stream sends, so you can do reporting and analysis on the collected data, enable **Historic data analysis**.

After you successfully create your data stream, you get a REST API URL endpoint. Your application can call the endpoint by using **POST** requests to push your streaming data to the Power BI dataset. In your **POST** requests, ensure that the request body matches the

sample JSON that the Power BI user interface provided. For example, wrap your JSON objects in an array.

 **Caution**

For streaming datasets you create in the Power BI service UI, the dataset owner gets a URL that includes a *resource key*. This key authorizes the requestor to push data into the dataset without using an Azure AD OAuth bearer token. Keep in mind the implications of having a secret key in the URL when you work with this type of dataset and method.

## Use PubNub

The integration of PubNub streaming with Power BI helps you create and use your low-latency PubNub data streams in Power BI. When you select **PubNub** on the **New streaming dataset** screen and select **Next**, you see the following screen:

# New streaming dataset

For customers of the PubNub data stream network, subscribe to a channel to display data on your dashboard. [Learn more about PubNub.](#)

Dataset name \*

Sub-key \*

Channel name \*

PAM Auth Key

Back

Next

Cancel

## ⓘ Important

You can secure PubNub channels by using a PubNub Access Manager (PAM) authentication key. This key is shared with all users who have access to the dashboard. For more information about PubNub access control, see [Manage Access](#).

PubNub data streams are often high volume, and aren't always suitable for storage and historical analysis in their original form. To use Power BI for historical analysis of PubNub data, you must aggregate the raw PubNub stream and send it to Power BI, for example by using [Azure Stream Analytics](#).

# Example of real-time streaming in Power BI

Here's an example of how real-time streaming in Power BI works. This sample uses a publicly available stream from PubNub. Follow along with the example to see the value of real-time streaming for yourself.

1. In the Power BI service, select or create a new dashboard. At the top of the screen, select **Edit > Add a tile**.
2. On the **Add a tile** screen, select **Custom Streaming Data**, and then select **Next**.

## Add a tile

### Select source

#### MEDIA



Web content



Image



Text box



Video

#### REAL-TIME DATA



Custom Streaming  
Data

Next

Cancel

3. On the Add a custom streaming data tile page, select Add streaming dataset.

## Add a custom streaming data tile

Choose a streaming dataset

+ Add streaming dataset

### YOUR DATASETS

weather forecast

[Manage datasets](#)

[Back](#)

[Next](#)

[Cancel](#)

4. On the **New streaming dataset** page, select **PubNub**, and then select **Next**.
5. On the next screen, enter a **Dataset name**, enter the following values into the next two fields, and then select **Next**.
  - **Sub-key:** *sub-c-99084bc5-1844-4e1c-82ca-a01b18166ca8*
  - **Channel name:** *pubnub-sensor-network*

## New streaming dataset

For customers of the PubNub data stream network, subscribe to a channel to display data on your dashboard. [Learn more about PubNub.](#)

Dataset name \*

Sub-key \*

Channel name \*

PAM Auth Key

6. On the next screen, keep the automatically populated values, and select **Create**.

## New streaming dataset

\* Required

Dataset name \*

PNdemo

Values from stream \*

sensor\_uuid

Text



humidity

Number



photosensor

Number



ambient\_temperature

Number



radiation\_level

Number



timestamp

DateTime



Enter a new value name

Text



```
[
 {
 "sensor_uuid" : "AAAAAA5555555",
 "humidity" : 98.6,
 "photosensor" : 98.6,
 "ambient_temperature" : 98.6,
 "radiation_level" : 98.6,
 "timestamp" : "2023-01-03T21:52:03.422Z"
 }
]
```

Create

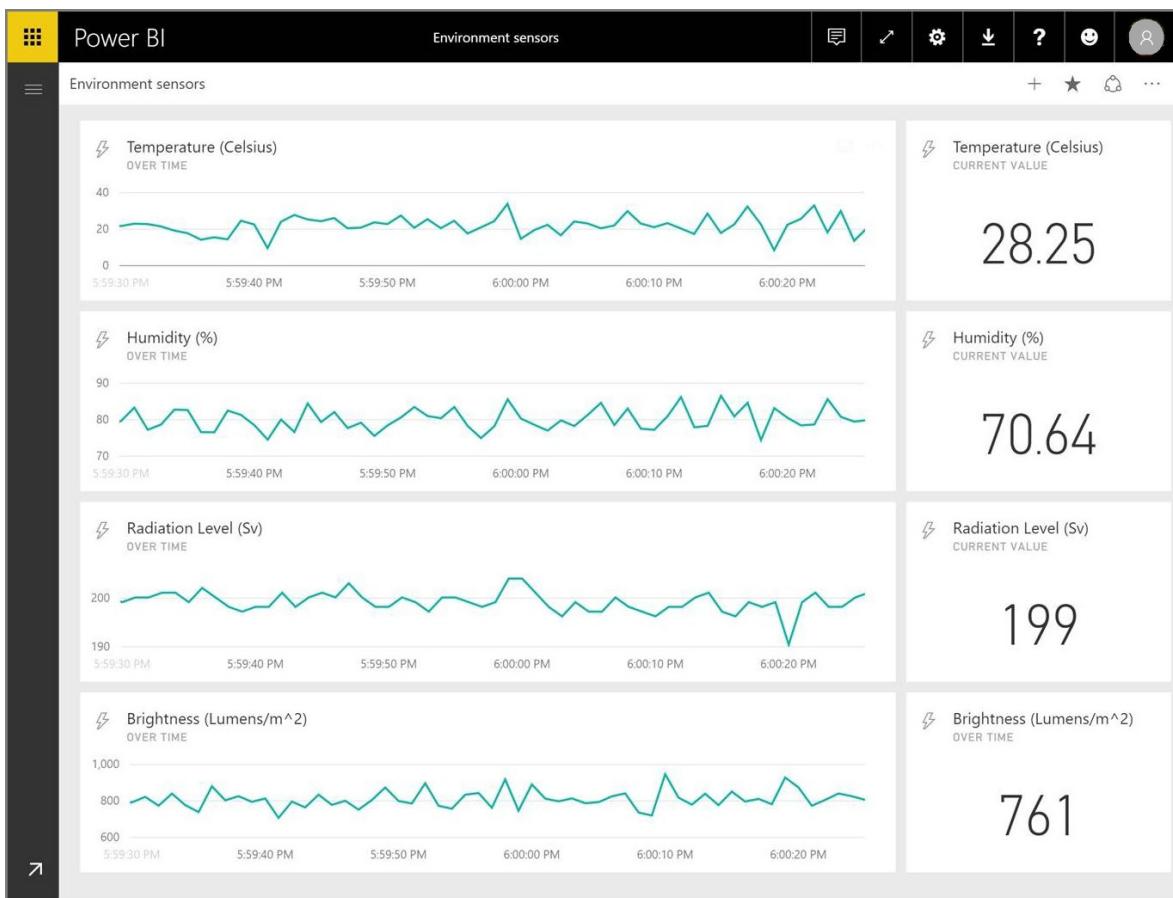
Cancel

7. Back in your Power BI workspace, create a new dashboard, and at the top of the screen, select **Edit > Add a tile**.

8. Select **Custom Streaming Data**, and select **Next**.

9. On the **Add a custom streaming data tile** page, select your new streaming dataset, and then select **Next**.

Play around with the sample dataset. By adding value fields to line charts and adding other tiles, you can get a real-time dashboard that looks like the following screenshot:



Go on to create your own datasets and stream live data to Power BI.

## Questions and answers

Here are some common questions and answers about real-time streaming in Power BI.

### Can you use filters on push or streaming datasets?

Streaming datasets don't support filtering. For push datasets, you can create a report, filter the report, and then pin the filtered visuals to a dashboard. However, there's no way to change the filter on the visual once it's on the dashboard.

You can pin the live report tile to the dashboard separately, and then you can change the filters. However, live report tiles won't update in real time as data is pushed in. You have to manually update the visual by selecting the Refresh icon at top right on the dashboard page.

When you apply filters to push datasets that have `DateTime` fields with millisecond precision, equivalence operators aren't supported. Operators such as greater than `>` or

less than < operate properly.

## How do you see the latest value on push or streaming datasets?

Streaming datasets are designed to display the latest data. You can use the Card streaming visual type to easily see the latest numeric values. Card visuals don't support `DateTime` or `Text` data types.

For push datasets, if you have a timestamp in the schema, you can try creating a report visual with the `last N` filter.

## How can you do modeling on real-time datasets?

Modeling isn't possible on a streaming dataset, because the data isn't stored permanently. For a push dataset, you can use the [create dataset REST API](#) to create a dataset with relationship and measures, and use the [update table REST APIs](#) to add measures to existing tables.

## How can you clear all the values on a push or streaming dataset?

On a push dataset, you can use the [delete rows REST API](#) call. There's no way to clear data from a streaming dataset, although the data will clear itself after an hour.

## If you set up an Azure Stream Analytics output to Power BI but you don't see it in Power BI, what's wrong?

Take these steps to troubleshoot the issue:

1. Restart the Azure Stream Analytics job.
2. Try reauthorizing your Power BI connection in Azure Stream Analytics.
3. Make sure that you're checking the same workspace in the Power BI service that you specified for the Azure Stream Analytics output.
4. Make sure the Azure Stream Analytics query explicitly outputs to the Power BI output by using the `INTO` keyword.
5. Determine whether the Azure Stream Analytics job has data flowing through it. The dataset is created only when data is being transmitted.
6. Look into the Azure Stream Analytics logs to see if there are any warnings or errors.

## Automatic page refresh

You can use automatic page refresh at a report page level to set a refresh interval for visuals that's only active when the page is being consumed. Automatic page refresh is available only for DirectQuery data sources. The minimum refresh interval depends on the type of workspace where the report is published and capacity admin settings for Premium workspaces.

For more information about automatic page refresh, see [Automatic page refresh in Power BI](#).

## Next steps

- [Overview of the Power BI REST API with real-time data](#)
- [Load data in a Power BI streaming dataset and build a dataflows monitoring report with Power BI](#)

# Publish to Power BI from Microsoft Excel

Article • 08/28/2023

## Important

The following capabilities are deprecated and will no longer be available starting September 29th, 2023:

- Upload of local workbooks to Power BI workspaces will no longer be allowed.
- Configuring scheduling of refresh and refresh now for Excel files that don't already have scheduled refresh configured will no longer be allowed.

The following capabilities are deprecated and will no longer be available starting October 31, 2023:

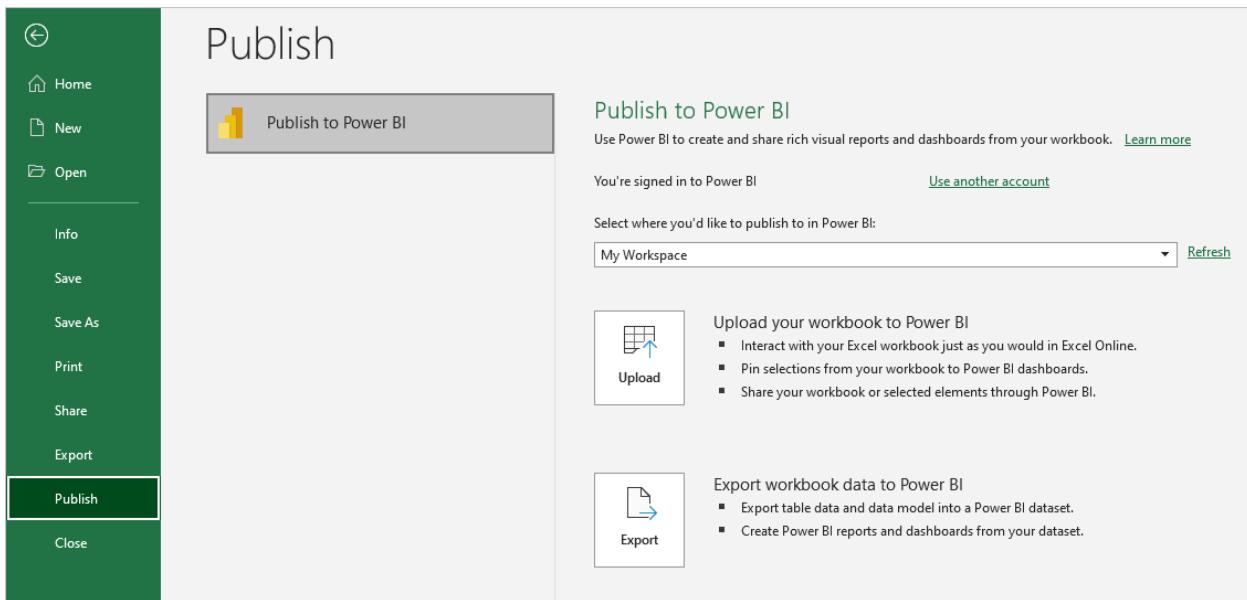
- Scheduled refresh and refresh now for existing Excel files that were previously configured for scheduled refresh will no longer be allowed.
- Local workbooks uploaded to Power BI workspaces will no longer open in Power BI.

After October 31, 2023:

- You can download existing local workbooks from your Power BI workspace.
- You can publish your Excel data model as a Power BI dataset and schedule refresh.
- You can import Excel workbooks from OneDrive and SharePoint Document libraries to view them in Power BI.

If your organization uses these capabilities, see more details in [Migrating your Excel workbooks](#).

With Microsoft Excel 2016 and later, you can publish your Excel workbooks directly to your [Power BI](#) workspace. In Power BI, you can create highly interactive reports and dashboards based on your workbook data. You can then share your insights with others in your organization.



When you publish a workbook to Power BI, there are few things to consider:

- You must use the same account to sign in to Office, OneDrive for work or school if your workbooks are saved there, and Power BI.
- You can't publish an empty workbook, or a workbook that doesn't have any Power BI supported content.
- You can't publish encrypted or password protected workbooks, or workbooks with Information Protection Management applied.
- Publishing to Power BI requires modern authentication to be enabled, the default. Otherwise, the **Publish** option isn't available from the **File** menu.
- Publishing to Power BI from Excel Desktop isn't supported for sovereign clouds.

## Publish your Excel workbook

To publish your Excel workbook to Power BI, in Excel, select **File** > **Publish** and select either **Upload** or **Export**. The following screenshot shows the two options for how to get your workbook into Power BI:

The screenshot shows the 'Publish' dialog in Excel. On the left, a sidebar lists options: Home, New, Open, Info, Save, Save As, Print, Share, Export, Publish (which is selected and highlighted in green), and Close. The main area is titled 'Publish to Power BI' with the sub-instruction 'Use Power BI to create and share rich visual reports and dashboards from your workbook.' Below this, it says 'You're signed in to Power BI' and 'Select where you'd like to publish to in Power BI:' followed by a dropdown menu set to 'My Workspace' with a 'Refresh' button. Two large buttons are shown: 'Upload' (with an icon of a grid and an upward arrow) and 'Export' (with an icon of a document and an arrow). A red box highlights the 'Upload' and 'Export' sections.

- If you select **Upload**, you can interact with the workbook just as you would in Excel Online. You can also pin selections from your workbook onto Power BI dashboards, and share your workbook or selected elements through Power BI.
- If you select **Export**, you can export table data and its data model into a Power BI dataset, and use the dataset to create Power BI reports and dashboards.

When you select **Publish**, you can select the workspace to publish to. If your Excel file is on OneDrive for work or school, you can publish only to your *My Workspace*. If your Excel file is on a local drive, you can publish to *My Workspace* or to a shared workspace you can access.

This screenshot shows the 'Publish to Power BI' dialog. It includes the same top-level sections as the previous screenshot. The 'Select where you'd like to publish to in Power BI:' dropdown is highlighted with a red box. Below it, the 'Upload' section is shown again with its description: 'Upload your workbook to Power BI' and 'Interact with your Excel workbook just as you would in Excel Online.'

## Publish local files

Excel supports publishing local Excel files. Files don't need to be saved to OneDrive for work or school or to SharePoint Online.

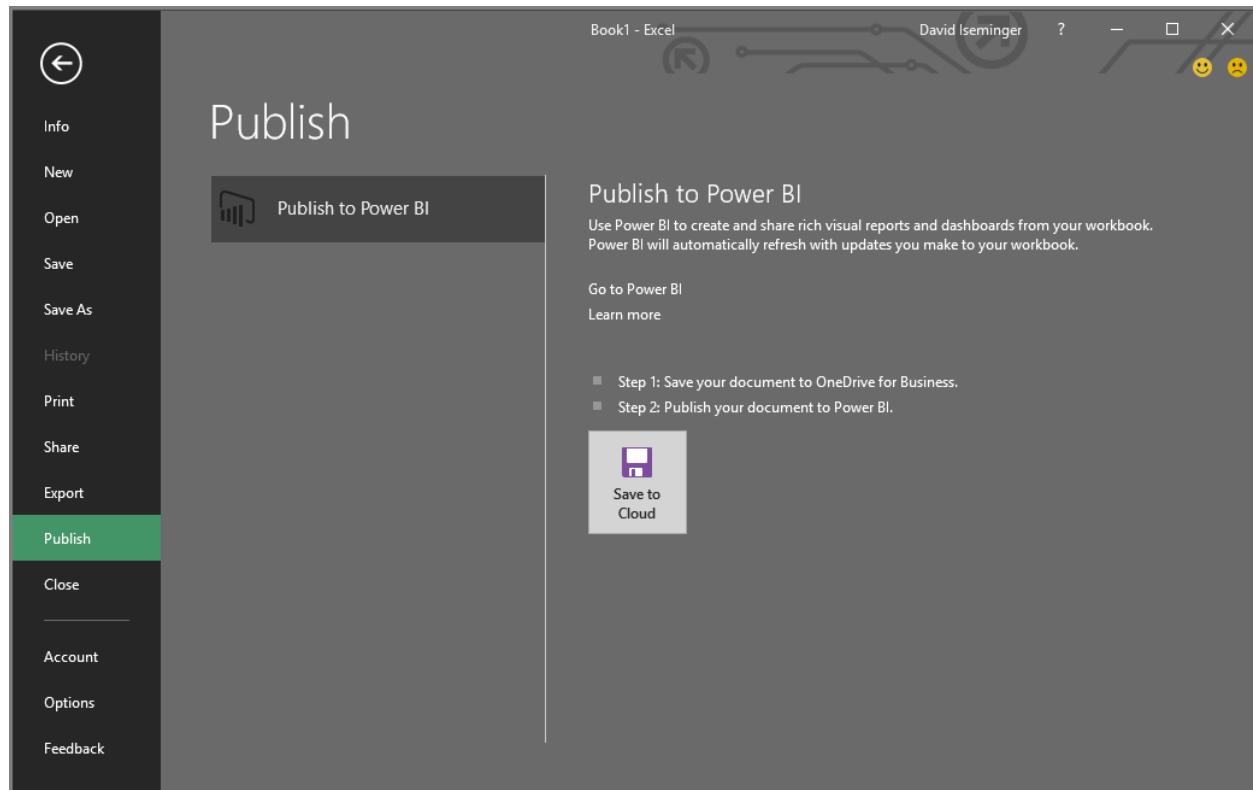
### ⓘ Important

You can publish local files only if you're using Excel 2016 or later with a Microsoft 365 subscription. Excel 2016 standalone installations can publish to Power BI, but only when the workbook is saved to OneDrive for work or school or to SharePoint Online.

Once published, the workbook content you publish imports into Power BI, separate from the local file. If you want to update the file in Power BI, you must update the local file and publish the updated version again. Or, you can refresh the data by configuring scheduled refresh on the workbook or dataset in Power BI.

## Publish from a standalone Excel installation

When you publish from a standalone Excel installation, you must save the workbook to OneDrive for work or school. Select **Save to Cloud** and choose a location in OneDrive for work or school.



Once you save your workbook to OneDrive for work or school, when you select **Publish**, you can use the **Upload** or **Export** options to get your workbook into Power BI.

The screenshot shows the 'Publish' dialog box in Excel. On the left, a sidebar lists options: Home, New, Open, Info, Save, Save As, Print, Share, Export, Publish (which is selected and highlighted in green), and Close. The main area is titled 'Publish to Power BI' with a sub-section 'Upload your workbook to Power BI'. It includes a 'Upload' button with an icon of a document with an upward arrow, a list of three bullet points, and a 'Export' button with an icon of a document with an arrow pointing right, followed by a list of two bullet points.

## Upload your workbook to Power BI

When you choose the **Upload** option, your workbook appears in Power BI just as it would in Excel Online. But unlike in Excel Online, you have options to let you pin elements from your worksheets to dashboards.

If you choose **Upload**, you can't edit your workbook in Power BI. If you need to change the data, you can select **Edit**, and then choose to edit your workbook in Excel Online or open it in Excel on your computer. Any changes you make are saved to the workbook on OneDrive for work or school.

When you choose **Upload**, no dataset is created in Power BI. Your workbook appears in your workspace navigation pane under **Reports**. Workbooks uploaded to Power BI have an Excel icon that identifies them as uploaded Excel workbooks.

Choose the **Upload** option if you only have data in worksheets, or you have PivotTables and Charts you want to see in Power BI.

Using **Upload** from **Publish to Power BI** in Excel is a similar experience to using **Upload > OneDrive for Business > Upload** in Power BI, and then opening the file in Excel Online from Power BI in your browser.

## Export workbook data to Power BI

When you choose the **Export** option, any supported data in tables and/or a data model are exported into a new dataset in Power BI. You can continue editing your workbook. When you save your changes, they synchronize with the dataset in Power BI, usually within about an hour. If you need more immediate updates, you can select **Publish**.

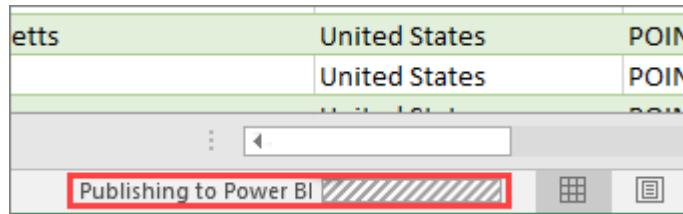
again from Excel to export your changes immediately. Any visualizations in reports and dashboards update too.

Choose the **Export** option if you used the **Get & Transform data** or **Power Pivot** features to load data into a data model.

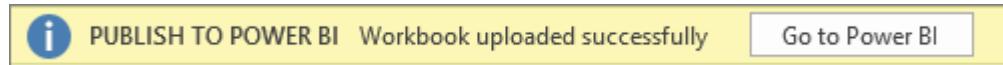
Using **Export** is similar to using **New > Upload a file > Excel > Import** from Power BI in your browser.

## Publish

When you choose either **Upload** or **Export**, Excel signs in to Power BI with your current account and publishes your workbook to your Power BI workspace. You can monitor the status bar in Excel to see publishing progress.



When publishing is complete, you can go to Power BI directly from Excel.



## Next steps

- [Excel data in Power BI](#)
- More questions? [Try the Power BI Community.](#) ↗

# Reduce the size of an Excel workbook to view it in Power BI

Article • 02/09/2023

You can upload any Excel workbook smaller than 1 GB to Power BI. An Excel workbook can have two parts: a Data Model, and the rest of the report—the core worksheet contents. If the report meets the following size limits, you can save it to **OneDrive for work or school**, connect to it from Power BI, and view it in Excel Online:

- The workbook can be up to 1 GB.
- The core worksheet contents can be up to 30 MB.

## What makes core worksheet contents larger than 30 MB

Here are some elements that can make the core worksheet contents larger than 30 MB:

- Images
- [Shaded cells](#)
- [Colored worksheets](#)
- Text boxes
- Clip art

Consider removing these elements, if possible.

If the report has a Data Model, you have some other options:

- Remove data from Excel worksheets and store it in the Data Model. For more information, see the [Remove data from worksheets](#) section.
- [Create a memory-efficient Data Model](#) to reduce the overall size of the report.

To make any of these changes, you need to edit the workbook in Excel.

For more information, see [File size limits for Excel workbooks in SharePoint Online](#).

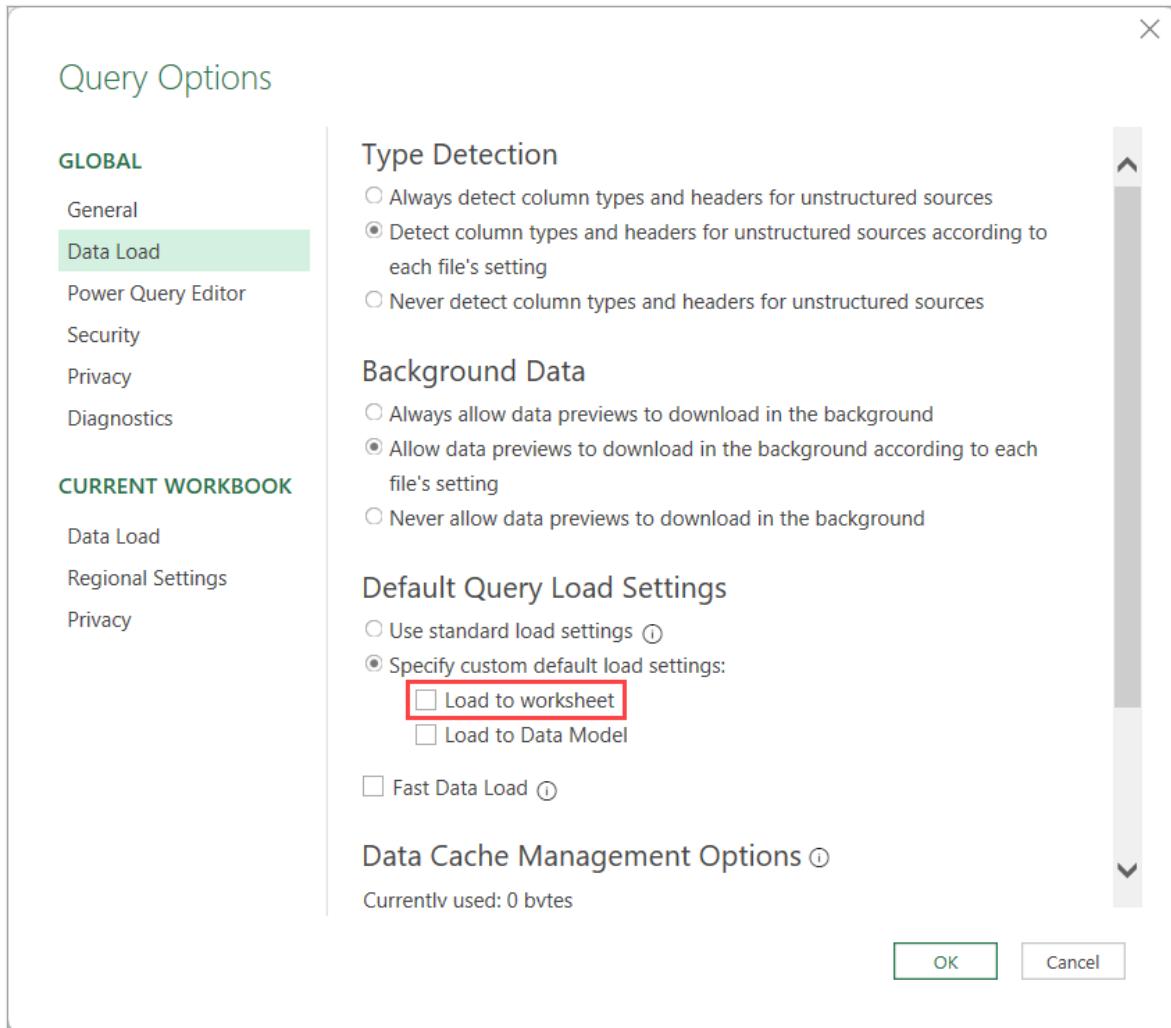
## Remove data from worksheets

If you import data into Excel from the Power Query Editor or the Excel Data tab, the workbook might have the same data in an Excel table and in the Data Model. Large tables in Excel worksheets might make the core worksheet contents more than 30 MB.

Removing the table in Excel and keeping the data in the Data Model can greatly reduce the core worksheet contents of the report.

When you import data into Excel, follow these tips:

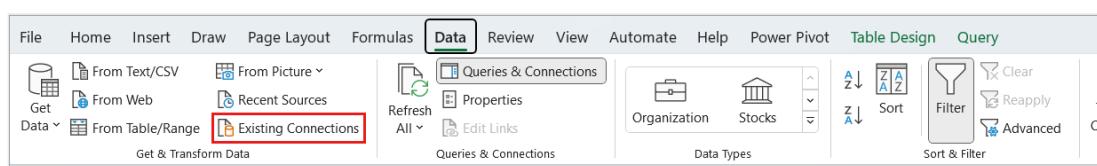
- In Power Query Editor: Clear the Load to worksheet box under File > Options and settings > Query Options.



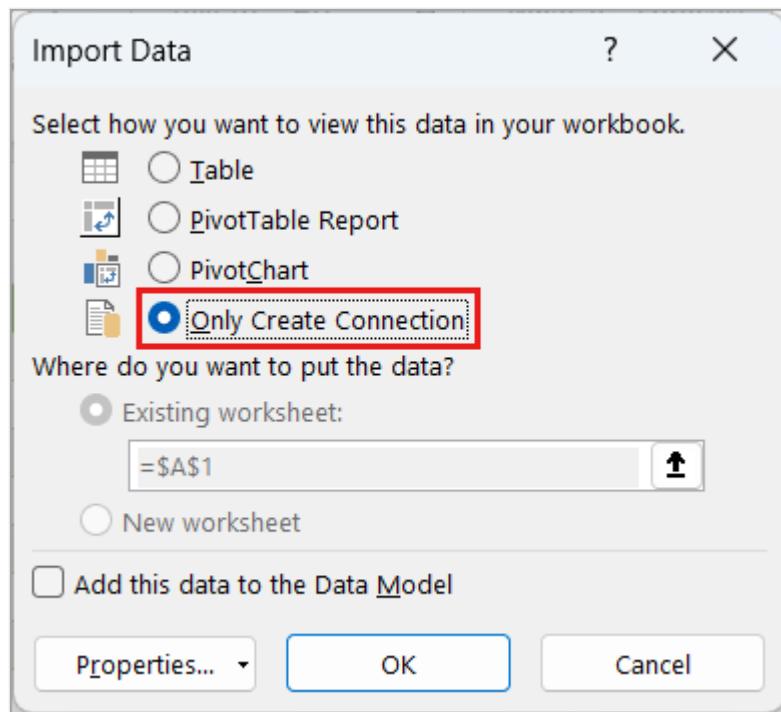
This imports data only into the Data Model, with no data in Excel worksheets.

- From the Excel Data tab, if you previously selected Table in the import wizard:

1. Go to Existing Connections.



2. Select the connection, and then select Open. Select Only Create Connection.



3. Delete the original table or tables created during the initial import.

## Workbook size optimizer

If your workbook contains a data model, you can run the workbook size optimizer to reduce the size of your workbook. For more information, see [Download Workbook Size Optimizer ↗](#).

## Related info

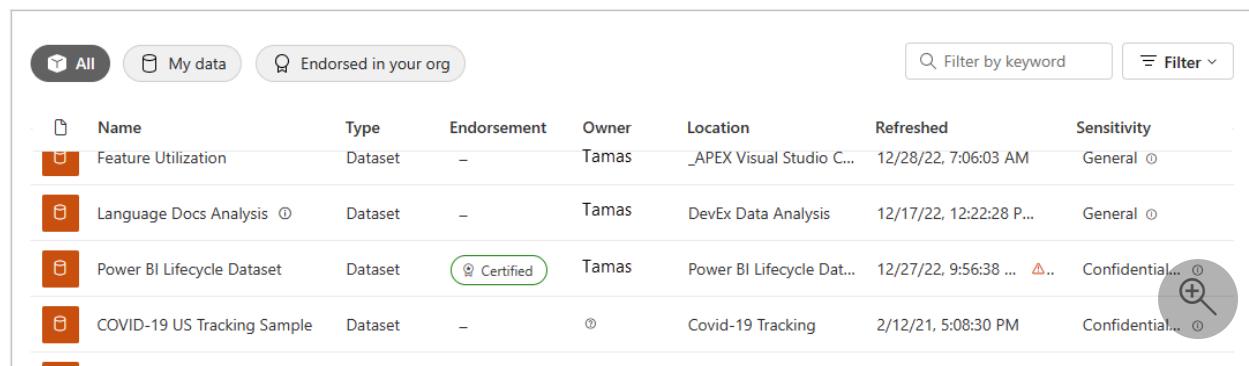
[Create a memory-efficient Data Model by using Excel and the Power Pivot add-in ↗](#)

[Use OneDrive for work or school links in Power BI Desktop](#)

# Introduction to datasets across workspaces

Article • 01/09/2023

Business intelligence is a collaborative activity. It's important to establish standardized datasets that can be the *one source of truth*. Then discovering and reusing those standardized datasets is key. When expert data modelers in your organization create and share optimized datasets, report creators can start with those datasets to build accurate reports. Then your organization has consistent data for making decisions, and a healthy data culture.



All	My data	Endorsed in your org	Filter by keyword		Filter	
Name	Type	Endorsement	Owner	Location	Refreshed	Sensitivity
Feature Utilization	Dataset	-	Tamas	_APEX Visual Studio C...	12/28/22, 7:06:03 AM	General ⓘ
Language Docs Analysis ⓘ	Dataset	-	Tamas	DevEx Data Analysis	12/17/22, 12:22:28 P...	General ⓘ
Power BI Lifecycle Dataset	Dataset	Certified	Tamas	Power BI Lifecycle Dat...	12/27/22, 9:56:38 ... ⓘ	Confidential... ⓘ
COVID-19 US Tracking Sample	Dataset	-	①	Covid-19 Tracking	2/12/21, 5:08:30 PM	Confidential... ⓘ

In Power BI, dataset creators can control who has access to their data by using the [Build permission](#). Dataset creators can also *certify* or *promote* datasets so others can discover them. That way, report authors know which datasets are high quality and official, and they can use those datasets wherever they author in Power BI. Administrators have a new tenant setting to [govern the use of datasets across workspaces](#).

## Dataset sharing and workspaces

Building reports based on datasets in different workspaces, and copying reports to different workspaces, are tightly coupled with the [workspace](#):

- In the Power BI service, when you open the dataset catalog from a workspace, the dataset catalog shows datasets in your **My workspace** and in other workspaces.
- In Power BI Desktop, you can publish Live Connect reports to different workspaces.

## Discover datasets

When you build a report on top of an existing dataset, the first step is to connect to the dataset, either in the Power BI service or Power BI Desktop. Read about [discovering datasets from different workspaces](#)

# Copy a report

When you find a report you like, in a workspace or an app, you can make a copy of it, and then modify it to fit your needs. You don't have to worry about creating the data model. The data model is already created for you. And it's much easier to modify an existing report than it is to start from scratch. Read more about [copying reports](#).

# Build permission for datasets

With **Build** permission type, if you're a dataset creator, you can determine who in your organization can build new content on your datasets. People with **Build** permission can also build new content on the dataset outside Power BI, such as Excel sheets via Analyze in Excel, XMLA, and export. Read more about the [Build permission](#).

# Promotion and certification

If you create datasets, when you create one that others can benefit from, you can make it easier for them to discover it by [promoting your dataset](#). You can also request that experts in your organization [certify your dataset](#).

# Licensing

The specific features and experiences built on shared dataset capabilities are licensed according to their existing scenarios. For example:

- In general, discovering and connecting to shared datasets is available to anyone. It isn't a feature restricted to Premium.
- Users without a Pro or Premium Per User (PPU) license can only use datasets across workspaces for report authoring if those datasets reside in the users' personal **My workspace** or in a Premium-backed workspace. The same licensing restriction applies whether they author reports in Power BI Desktop or in the Power BI service.
- Copying reports between workspaces requires a Pro or Premium Per User license.
- Copying reports from an app requires a Pro or Premium Per User license.
- Promoting and certifying datasets requires a Pro or Premium Per User license.

# Considerations and limitations

- As an app publisher, you have to make sure that your audience has access to datasets outside of the workspace. Otherwise, users will encounter issues when

interacting with your app: reports won't open without dataset access and dashboard tiles will show as locked. Also, users won't be able to open the app if the first item in its navigation is a report without access to the dataset.

- By design, *Publish to web* doesn't work for a report based on a shared dataset.
- If two people are members of a workspace that is accessing a shared dataset, it's possible that only one of them can see the related dataset in the workspace. Only people with at least **Read** access to the dataset can see the shared dataset.

## Next steps

- [Promote datasets](#)
- [Certify datasets](#)
- [Request dataset certification](#)
- [Govern the use of datasets across workspaces](#)
- Questions? [Try asking the Power BI Community](#)↗

# Dataset description

Article • 02/08/2023

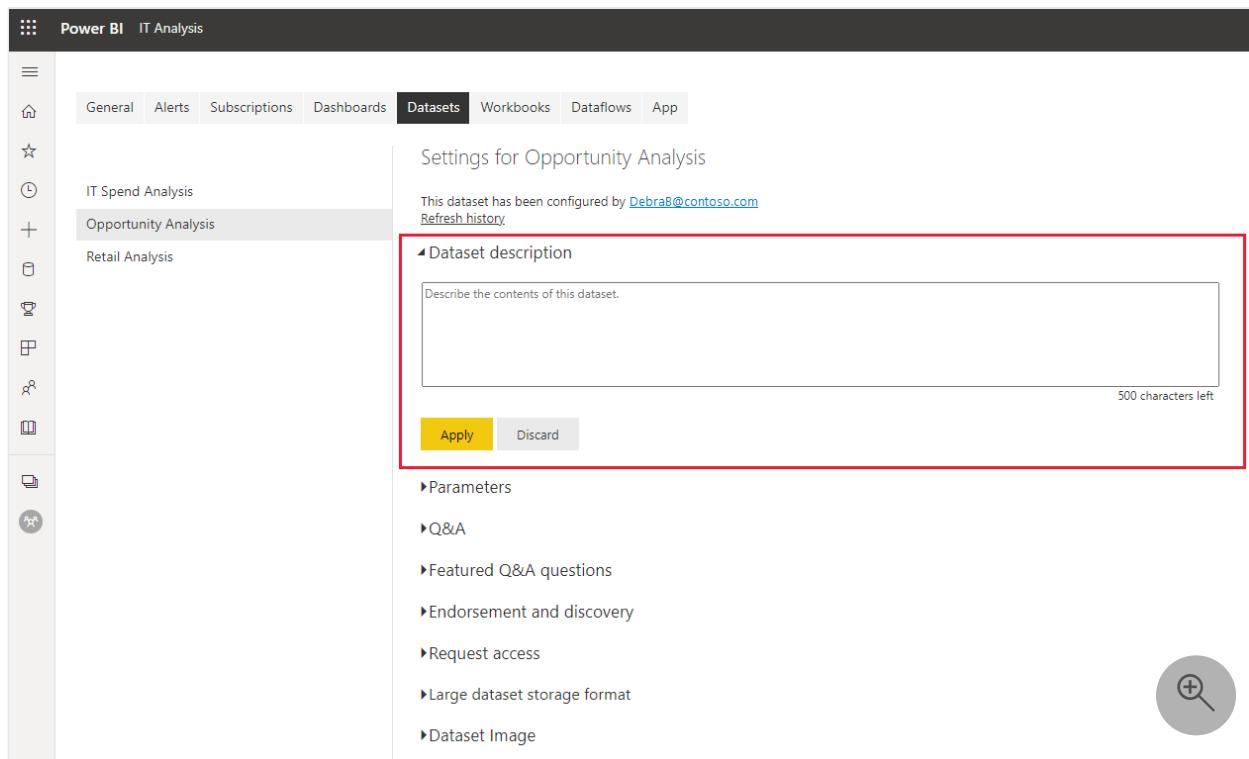
To help members of your organization quickly identify datasets that might be useful for them, provide a concise, informative description of your dataset in the dataset's settings. Users see this description in the tooltip next to the dataset's name in the datasets hub and on the dataset's details page.

Providing a meaningful description helps foster dataset reuse. For instance, based on a dataset's description, users may decide to explore reports that are based on the dataset, or to create their own reports based on the dataset.

## Provide a dataset description

To provide a description for a dataset, go to dataset's settings page, find the **Dataset description** section, and enter your description in the text box.

In the settings page, find the **Dataset description** section, and enter your description in the text box. Select **Apply** to save your description.



## Next steps

- [Data hub](#)
- Questions? [Try asking the Power BI Community](#) ↗



# Share access to a dataset (preview)

Article • 08/24/2022

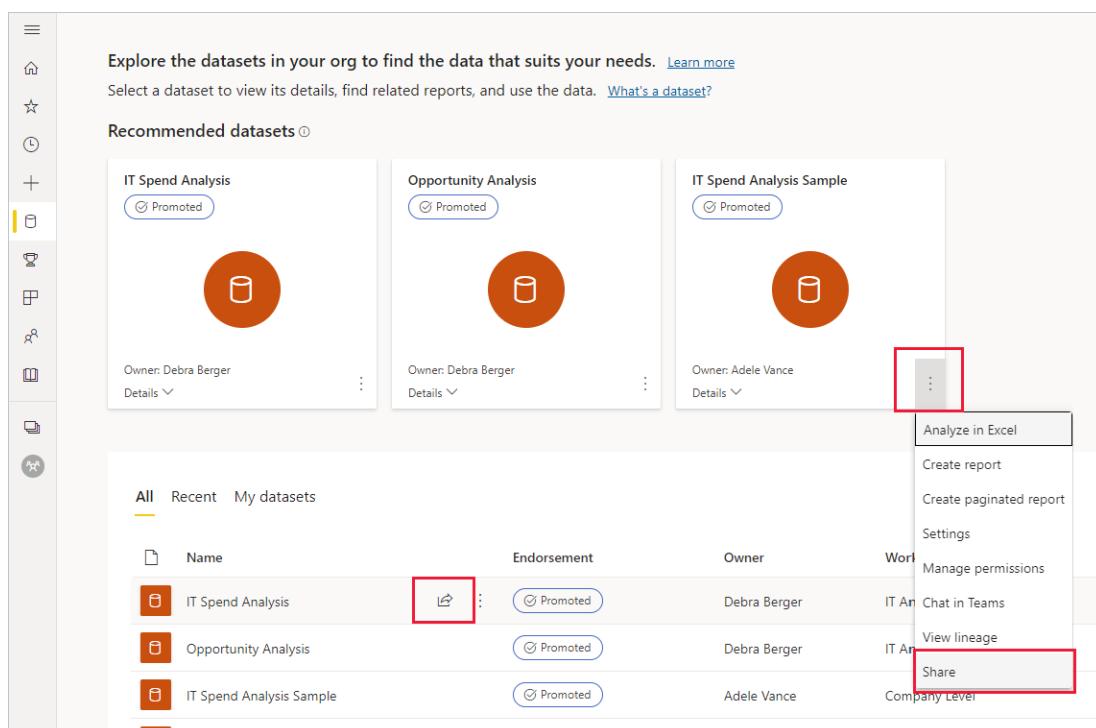
To make it possible for other users to take advantage of a dataset, you can *share* it with them. Sharing a dataset means granting access to it. This document shows you how to grant access to a dataset using the **Share dataset** dialog.

## Share a dataset

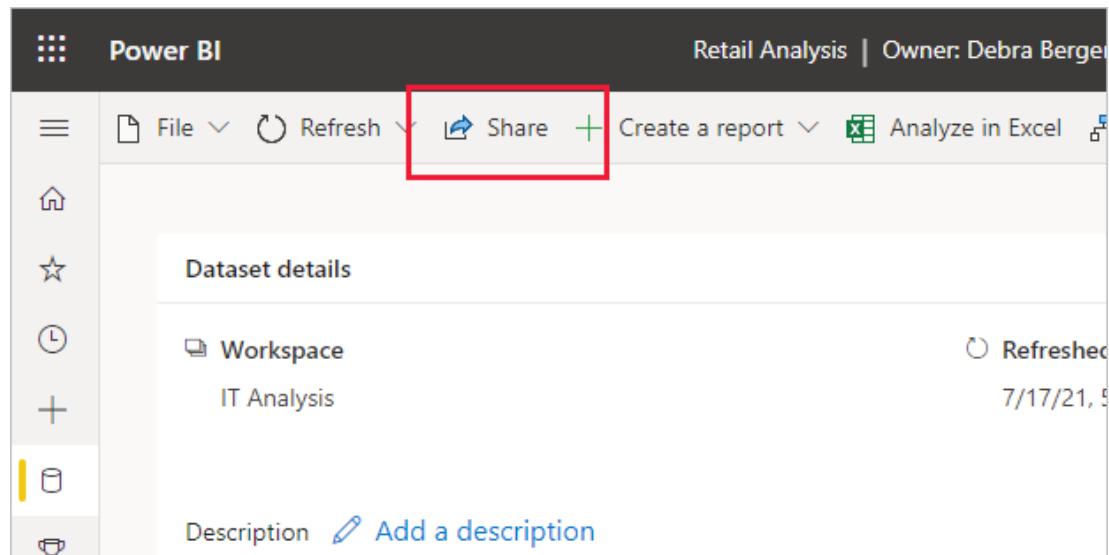
To share a dataset

1. On either the [data hub page](#) or on the [data details page](#), choose **Share** as follows:

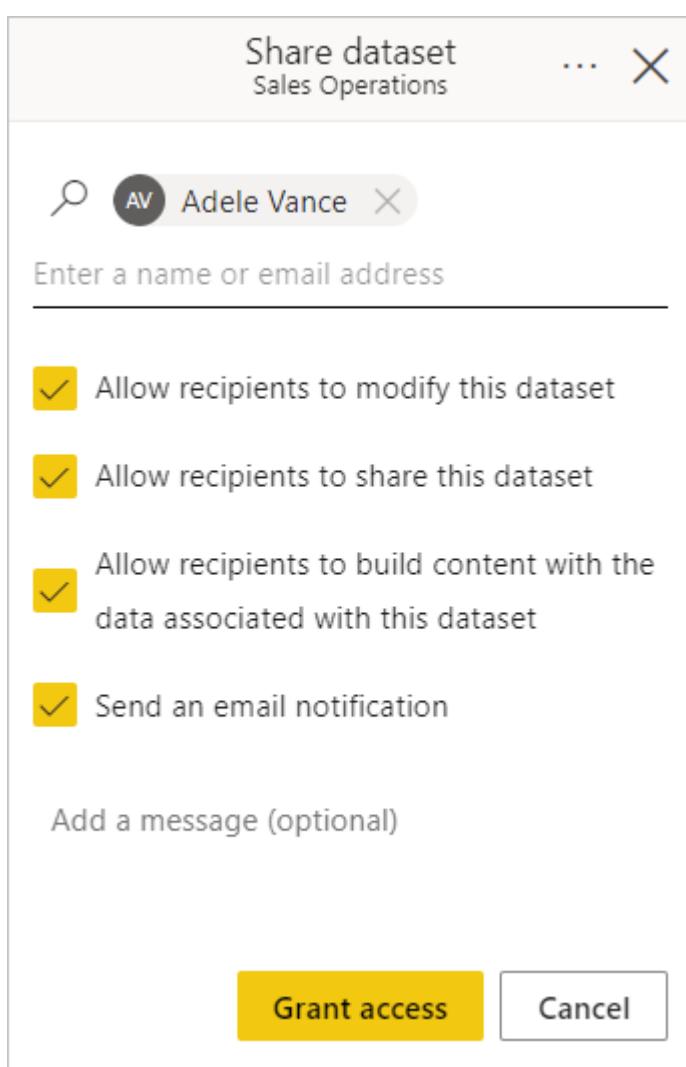
- **Data hub:** In the data items list, click the **Share** icon. On a recommended data item tile, choose **Share** on the **More options (...)** menu.



- **Dataset info page:** Click the **Share** icon on the action bar at the top of the page.



2. In the **Share dataset** dialog that appears, enter the names or email addresses of the specific people or groups (distribution groups or security groups) that you want to grant access to, then choose the types of access you wish to grant. You can optionally choose to send them an email notifying them that they've been granted access.



- **Allow allow recipients to modify this dataset:** This option allows the recipients to modify the dataset.

- **Allow recipients to share this dataset:** This option allows the recipients to grant access to other users via sharing.
- **Allow recipients to build content with the data associated with this dataset:** This option grants the recipients [Build permission](#) on the dataset, which enables them to build new reports and dashboards based on the data associated with it.

If you clear this checkbox, the user will get **read-only** permission on the dataset. Read-only permission allows them to explore the dataset on the [dataset's info page](#) but doesn't allow them to build new content based on the dataset.
- **Send an email notification:** When this option is selected, an email will be sent to the recipients notifying them that they have been granted access to the dataset. You can add an optional message to the email message.

3. Click **Grant access**.

 **Note**

When you press **Grant access**, access is granted automatically. No further approval is required.

To monitor, change, or remove user access to your dataset, see [Manage dataset access permissions](#).

## Next steps

- [Dataset permissions](#)
- [Manage dataset access permissions](#)
- [Use datasets across workspaces](#)
- [Share a report via link](#)
- Questions? [Try asking the Power BI Community](#) 

# Dataset permissions

Article • 03/14/2023

This article describes dataset permissions in the Power BI service and how these permissions are acquired by users.

## What are the dataset permissions?

The table below describes the four levels of permission that control access to datasets in the Power BI service. It also describes the permissions that the dataset owner has on the dataset, and other actions that only the dataset owner can perform.

Permission	Description
Read	<p>Allows user to access reports and other solutions, such as composite models on Premium/PPU workspaces, that read data from the dataset.</p> <p>Allows user to view dataset settings.</p>
Build	<p>Allows user to build new content from the dataset, as well as find content that uses the dataset.</p> <p>Allows user to access reports that access composite models on Power BI Pro workspaces.</p> <p>Allows user to build composite models.</p> <p>Allows user to pull the data into Analyze in Excel.</p> <p>Allows querying using external APIs such as XMLA.</p>
Reshare	Allows user to grant dataset access.
Write	<p>Allows user to republish the dataset.</p> <p>Allows user to <a href="#">backup and restore the dataset</a>.</p> <p>Allows user to make changes to the dataset via XMLA.</p> <p>Allows user to edit dataset settings, except data refresh, credentials, and automatic aggregations.</p>
Owner	<p>The dataset owner is not a permission per se, but rather a conceptual role that has all the permissions on a dataset. The first dataset owner is the person who created the dataset, and afterwards the last person to configure the dataset after taking it over in the dataset settings.</p> <p>The dataset owner can do all of the things mentioned in the permissions above, as well as configure dataset refresh, credentials, and automatic aggregations.</p>

## How are the dataset permissions acquired?

## Permissions acquired implicitly via workspace role

A user's role in a workspace implicitly grants them permissions on the datasets in the workspace, as described in the following table.

	Admin	Member	Contributor	Viewer
Read	✓	✓	✓	✓
Build	✓	✓	✓	✗
Reshare	✓	✓	✗	✗
Write	✓	✓	✓	✗

### ⓘ Note

Permissions inherited via workspace role can only be changed or taken away from a user by changing or removing their role in the workspace. They can't be changed or removed explicitly using the [manage permissions page](#).

## Permissions granted explicitly via the manage dataset permissions page

A user with an Admin or Member role in the workspace can explicitly grant permissions to other users using the [manage permissions page](#).

## Permissions acquired via a link

When users share reports or datasets, links are created that provide permissions on the dataset. Users authorized to use those links will be able to access the dataset. Users with Admin or Member roles in the workspace where a dataset is located can manage these links on the [manage permissions page](#).

## Permissions granted in an app

Users may acquire permissions on a dataset used in an app if the app owner allows this in the [app permissions configuration](#).

## Permissions granted via REST APIs

Dataset permissions can be set via REST APIs. For more information, see [Dataset permissions in the context of the Power BI REST APIs](#).

## Dataset permissions and row-level security (RLS)

Row-level security may affect the ability of users with read or build permission on a dataset to read data from the dataset.

- When RLS **isn't** defined on the dataset, users with write, read, or build permission on the dataset can read data from the dataset.
- When RLS **is** defined on the dataset:
  - Users with only read or build permission on the dataset will not be able to read data from the dataset unless they belong to one of its RLS roles.
  - Users with write permission on the dataset will be able to read data from the dataset regardless of whether or not they belong to any of its RLS roles.

## Next steps

- [Share access to a dataset](#)
- [Manage dataset permissions](#)
- [Dataset permissions in the context of the Power BI REST APIs](#)

# Manage dataset access permissions (preview)

Article • 08/18/2022

The dataset manage permissions page enables you to monitor and manage access to your dataset. It has two tabs that help you control access to your dataset:

- **Direct access:** Enables you to monitor, add, modify, or delete access permissions for specific people or groups (distribution groups or security groups).
- **Shared report links:** Shows you [links that were generated for sharing reports](#). Such links sometimes also give access to your dataset. On this tab you can review them and remove them if necessary.

This document explains how to use the dataset manage permissions page.

## ⓘ Note

In order to be able to access a dataset's manage permissions page, you must have an **admin** or **member** role in the workspace where the dataset is located.

## Open the dataset manage permissions page

To open the dataset manage permissions page:

- From the [data hub](#): Select **Manage permissions** on the **More options (...)** menu.

The screenshot shows the Power BI Datasets page. On the left, there's a sidebar with various icons. In the center, there are three dataset cards: 'IT Spend Analysis' (Owner: Debra Berger), 'IT Spend Analysis Sample' (Owner: Adele Vance), and 'Opportunity Analysis' (Owner: Debra Berger). Below these cards is a table of datasets. The 'Opportunity Analysis' row has a red box around its 'More options' button. A context menu is open over this button, with 'Manage permissions' highlighted. Another red box highlights the 'Manage permissions' option in this menu. To the right of the table, another context menu is open over a dataset card, also with 'Manage permissions' highlighted.

- From the [dataset details page](#): Select the **Share** icon on the action bar at the top of the page and choose **Manage permissions**.

The screenshot shows the 'Opportunity Analysis' dataset details page. At the top, there's a 'File' dropdown menu with a red box around it. Below the menu, a 'Share' dialog is open, also with a red box around its 'Manage permissions' button. The rest of the page shows dataset metadata like workspace, refresh time, and description.

- From the [Share dataset dialog](#): In the dialog header, select **Manage permissions** on the **More options (...)** menu. This opens the **Manage permissions** side pane. In the side pane, choose **Advanced** at the bottom of the pane.

The left screenshot shows the 'Share dataset' dialog for 'Sales Operations'. It includes a search bar, a list of permissions (Allow recipients to modify this dataset, Allow recipients to share this dataset, Allow recipients to build content with the data associated with this dataset, Send an email notification), an optional message field, and 'Grant access' and 'Cancel' buttons. The 'Manage permissions' button is highlighted with a red box. The right screenshot shows the 'Manage permissions' page for 'Sales Operations', which lists users with 'Direct access': Adele Vance (Owner), Debra Berger (Owner), Diego Siciliani, Lynne Robbins, System Administrator (Owner), and a placeholder user f33856de-7b3b-4077-9c6d-1e... The 'Advanced' button at the bottom right is highlighted with a red box.

These actions will open the datasets manage permissions page. The manage permissions page has two tabs to help you manage dataset access.

## Manage direct access

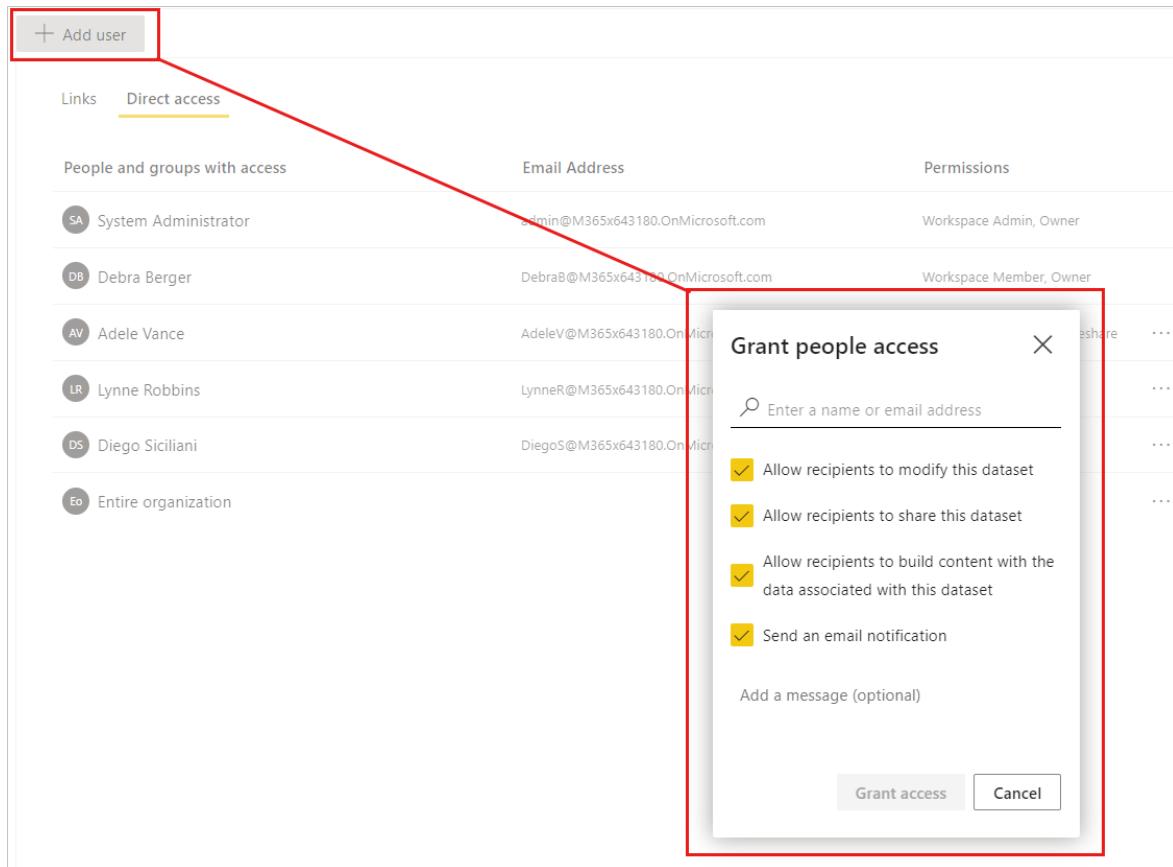
The direct access tab lists users who have been granted access. For each user, you can see their email address and the permissions they have.

- To modify a user's permissions, select **More options (...)** and choose one of the available options.

+ Add user		
Links	<b>Direct access</b>	
People and groups with access	Email Address	Permissions
SA System Administrator	admin@M365x643180.OnMicrosoft.com	Workspace Admin, Owner
DB Debra Berger	DebraB@M365x643180.OnMicrosoft.com	Workspace Member, Owner
AV Adele Vance	AdeleV@M365x643180.OnMicrosoft.com	Workspace Contributor, Reshare
LR Lynne Robbins	LynneR@M365x643180.OnMicrosoft.com	Workspace Contributor
DS Diego Siciliani	DiegoS@M365x643180.OnMicrosoft.com	Owner
Eo Entire organization		Read, Build

A red arrow points from the 'Direct access' tab in the top navigation to the 'More options (...)' menu for the last user in the list, which is highlighted by a red box. The menu options shown are Remove reshare, Remove build, Remove write, and Remove access.

- To grant dataset access to another user, click **+ Add user**. The **Share dataset dialog** will open.



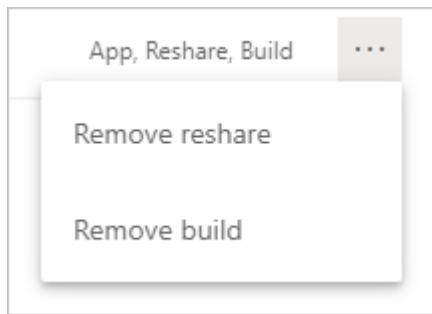
## Managing permissions granted through an app

Permissions on the dataset that have been granted through an app are indicated by the word "App" followed by the permissions enclosed in parentheses, as shown in the image below.



You can't modify permissions granted through an app directly from the Direct access tab - you must first remove them from the app configuration. To remove such permissions:

1. [Edit the app](#) and unselect the relevant permissions on the Permissions tab of the app's configuration settings.
2. Republish the app.
3. Go to the Direct access tab of the dataset's manage dataset permissions page as described [above](#). The user will still have the permissions that were granted via the app before update, but now they won't be tied to the app (note that the parentheses are gone). Now you can remove whatever permissions you desire.



## Manage links generated for report sharing

The shared report links tab lists [links that have been created to shared reports](#) that are based on your dataset. Such links may also grant access to the report's underlying dataset, and so they are listed here. You can see what permissions the link carries and who created the link. You can also delete the link from the system if you so desire.

### ⚠️ Warning

Deleting a link removes it from the system. Users who use the link to access a report may lose access to that report.

A screenshot of the Power BI service interface. On the left is a navigation pane with icons for Home, Dashboards, Reports, and Workbooks. The main area shows a report titled 'IT Spend Analysis Sample'. Under 'Related content', there are sections for 'Dashboards', 'Reports' (which includes 'IT Spend Analysis Sample', 'IT interim report - Fall', and 'Workbooks'), and 'Shared report links'. A red box highlights the 'Shared report links' tab. Below it is a table with columns: 'Links', 'Who has Access', 'Permissions', 'Creator', and 'Email Address'. Three rows of data are shown, each with a red box highlighting the 'Delete' icon (a small trash bin icon) in the last column. The table has a header row with these column names.

## Next steps

- [Dataset permissions](#)
- [Share access to a dataset](#)
- [Use datasets across workspaces](#)
- [Share a report via link](#)
- Questions? [Try asking the Power BI Community](#) ↗

# Build permission for shared datasets

Article • 03/14/2023

When you create a report in Power BI Desktop, the data in that report is stored in a *data model*. When you publish a report to the Power BI service, the data model is also published to the service as a *dataset* at the same time. When you share the report with others, you can give them *Build permission* for the dataset that the report is built on, so they can discover and reuse it for their own reports, dashboards, etc. This article explains how you control access to the dataset using Build permission.

Build permission applies to datasets. When you give users Build permission, they can build new content on your dataset, such as reports, dashboards, pinned tiles from Q&A, paginated reports, and Insights discovery. If a report outside the dataset workspace uses your dataset, you can't delete the dataset. If you try to do so, you get an error message.

Users also need Build permission to do the following actions:

- Export underlying Power BI data.
- Build new content on the dataset, such as with [Analyze in Excel](#).
- Access the data via the XML for Analysis (XMLA) endpoint.

## How users get Build permission

Users get Build permission for a dataset in a few different ways:

- Users that have at least a Contributor role in a workspace have Build permission on the datasets in that workspace, as well as permission to copy reports in that workspace. For more information about roles in workspaces, see [Roles in workspaces in Power BI](#).
- Dataset owners can assign Build permission to specific users or security groups on the [Manage permissions](#) page. For more information, see [Manage dataset access permissions](#).
- A user with an Admin or Member role in the workspace where the dataset resides can decide during app publishing that users with permission for the app also get Build permission for the underlying datasets. For more information, see [Create and manage multiple audiences](#).
- If you have Reshare and Build permission on a dataset, and you share a report or dashboard you built on that dataset, you can specify that the recipients also get

Build permission for the dataset. For more information, see [Share Power BI reports and dashboards with coworkers and others](#).

## Remove Build permission

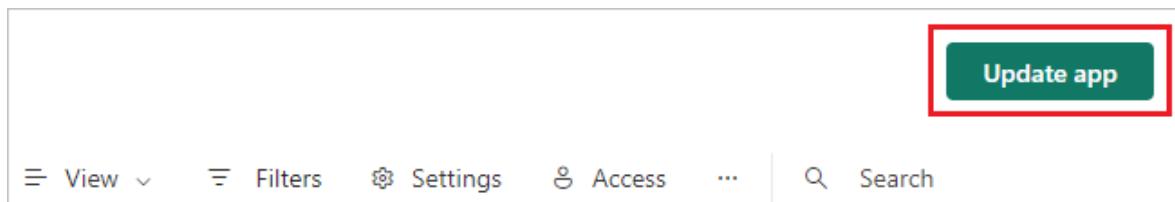
To remove Build permission for users of a shared dataset, follow the instructions at [Manage direct access](#).

If you remove Build permission, the people whose permission you revoked can still see the report, but can no longer edit the report or export underlying data. Users with only read permission can still export summarized data.

## Remove Build permission for a dataset in an app

If you distribute an app from a workspace, removing people's access to the app doesn't automatically remove their build and reshare permissions. To remove their Build permissions, take the following steps:

1. In the workspace, in list view, select **Update app**.



2. Select the **Audience** tab, and then in the **Manage Audience Access** side pane, hover over the person or group whose access you want to delete and select the trash icon that appears. When you're done, select **Update app**.

① Setup   ② Content   ③ Audience

Audience

Manage your audiences and their permissions. Select what content each audience can see by toggling the eye icon.

MyAppWorkspace + New Audience

Sales and Marketing

Overall Sales

Quick summary  
Sales and Marketing Sample PBIX

Total Category Volume by Segment

Segment	Total Category Volume
Productivity	0.42M
Convenience	0.33M
Moderation	0.19M
Extreme	0.17M
Select	0.06M
Youth	0.06M
All Season	0.04M
Regular	0.02M
Urban	0.06M

Count of Product by Category

Category	Count
Urban	1320

Manage Audience Access >  
MyAppWorkspace

Grant access to

Entire organization [Learn more](#)

Specific users or groups

Enter a name or email address

> Advanced

Workspace users [Edit](#)

Diego Siciliani DiegoS@M365x04027507.OnMicrosoft.com

Lynne Robbins LynneR@M365x04027507.OnMicrosoft.com [Delete](#)

Pradeep Gupta PradeepG@M365x04027507.OnMicrosoft.c...

Update app Cancel

You'll see a message that you need to go to **Manage permissions** to remove permissions for users with existing access.

## Emissions Impact Dashboard 11/21/2022, 9:52:44 AM

You aren't allowing new users to do the following:

**Build** content from the underlying datasets in this app.  
**Reshare** this app and its contents, including the underlying datasets.

Users who already have these permissions on the underlying datasets will retain their permissions. To remove users' existing access to the app's underlying datasets, go to Manage permissions, and remove these permissions manually.

### DATASETS

Emissions Impact Dashboard [Manage permissions](#)

When you update an app that has large distribution, it might take a little while to process. Typically, the latest content will be available within 5-10 minutes, but it can take up to one day.

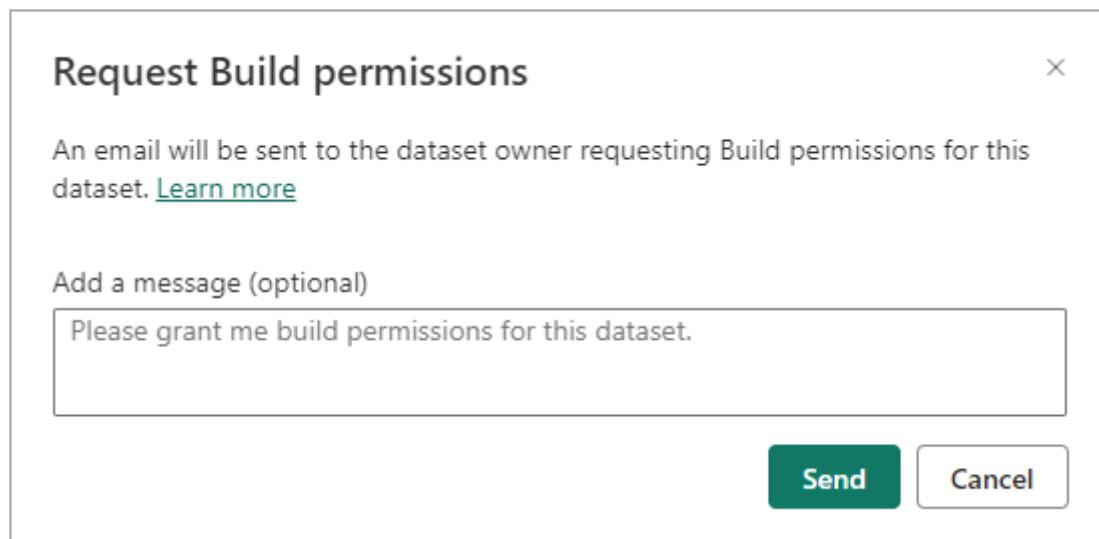
Update Cancel

3. Select **Update**.

4. Follow the instructions at [Manage permissions](#) to see how to remove permissions from users with existing access. When you take away a user's Build permission on a dataset, they can still see reports built on the dataset, but they can no longer edit the reports.

## Configure how users request Build permission

Certain actions, such as creating a report based on a dataset or accessing the details page of a dataset in the data hub, require Build permission on the dataset. By default, when users who don't have Build permission try these actions, they get a dialog box that lets them send email to the dataset owner requesting Build permission. The email includes the user's details, the name of the dataset they're requesting access to, and any other information they optionally provide.



## Change the access request behavior

If you have an [Admin](#), [Member](#), or [Contributor](#) role in the workspace where the dataset resides, you can change the default access request behavior for a dataset by going to the dataset's settings and configuring the **Request access** options as desired.

#### Request access

Select how users will request permissions to access content from this dataset. [Learn more](#)

A request for build permissions is sent in an email to the dataset owner

User requesting access will get the following instructions

(i) Your email address will be visible to users requesting access.

Instructions \*

In order to see and build reports on top of this data you need ContosoSales permissions, which you can get by following these steps:

1. Go to [aka.ms/ContosoSales](http://aka.ms/ContosoSales)
2. Select Accounts.
3. Select Join.

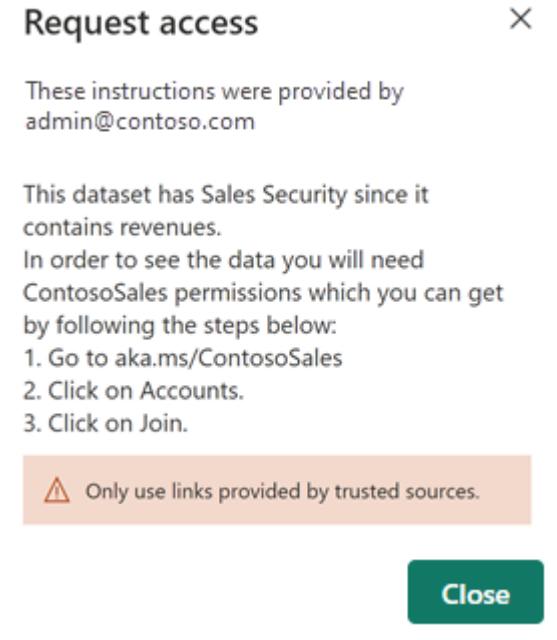
193 characters left

**Apply**

[Discard](#)

- The default option, not selected in the preceding image, is for Build permission requests to come to you via email. You're responsible for acting on the requests and notifying the requestors.
- The second option is for you to provide instructions about how to get Build permission, rather than receiving requests via email. You might choose this option, for example, if your organization uses an automated system for handling access requests. When users who don't have Build permission try an action that requires Build permission, they see a message with the instructions you provide.

The **Instructions** text area in the preceding **Request access** example shows sample instructions. Instructions must be in plain text. HTML or any other type of code formatting render as plain text, rather than the code format. The following example shows the instructions users see when they try an action they need Build permission for.



### ⓘ Note

When you provide specific instructions, your email address is visible to users who request access.

## More granular permissions

Power BI provides Build permission as a complement to Read and Reshare permissions. All users who already have Read permission for datasets via app permissions, sharing, or workspace access also get Build permission for those datasets. Those users get Build permission automatically because Read permission already grants them the right to build new content on the dataset by using **Analyze in Excel** or **Export**.

With the more granular Build permission, you can choose who can only view the content in an existing report or dashboard, and who can create content connected to the underlying dataset.

## Next steps

- [Use datasets across workspaces](#)
- [Share a dataset](#)
- [Roles in workspaces](#)
- [Manage dataset access permissions](#)

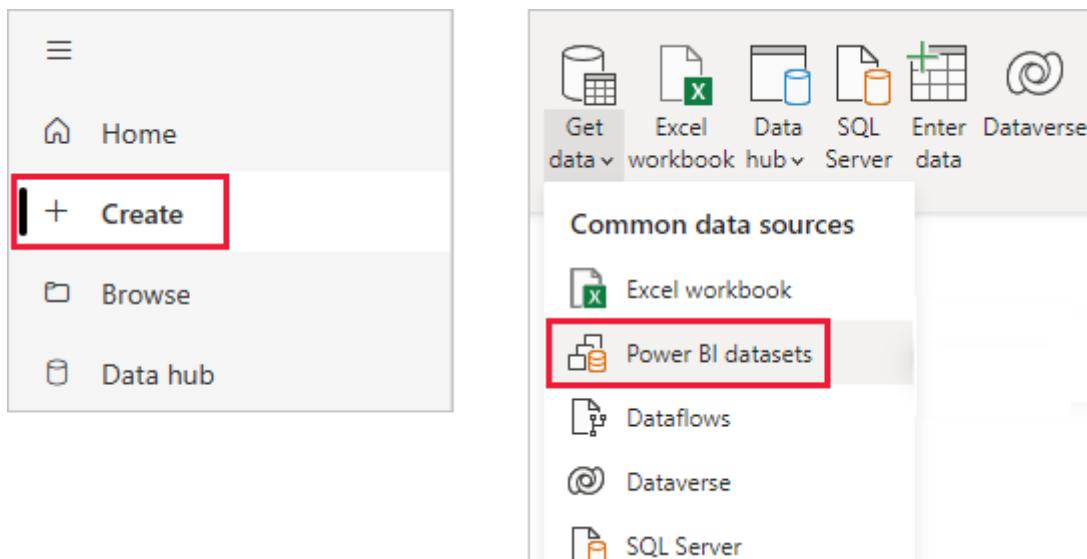
Questions? Try asking the Power BI Community ↗

# Create reports based on datasets from different workspaces

Article • 01/12/2023

Learn how you can create reports in your own workspaces based on datasets in other workspaces. To build a report on top of an existing dataset, you can start from Power BI Desktop or from the Power BI service, in your [My workspace](#) or in [another workspace](#).

- In the Power BI service: [Create > Report > Pick a published dataset](#).
- In Power BI Desktop: from the **Home** ribbon, select [Get data > Power BI datasets](#).



In both cases, the dataset discovery experience starts in the **Data hub**. You see all the datasets that you have access to, regardless of where they are:

The screenshot shows the Power BI Data hub interface. At the top, there's a header with the title 'Data hub' and a close button 'X'. Below the header, a sub-header says 'Discover data from across your org and use it to create reports. [Learn more](#)'. There are three tabs at the top left: 'All' (selected), 'My data', and 'Endorsed in your org'. To the right are two search/filter boxes: 'Filter by keyword' and 'Filter(1)'. The main area is a table with columns: Name, Endorsement, Owner, Workspace, Refreshed, and Sensitivity. The table contains seven rows of dataset information. Row 3 ('Opportunity-Analysis-Custo...') has a blue border around its 'Promoted' badge. Row 7 ('COVID-19 US Tracking Sample') has a blue border around its entire row. At the bottom right of the table area are two buttons: 'Connect' and 'Cancel'.

Name	Endorsement	Owner	Workspace	Refreshed	Sensitivity
AdventureWorks	-	②	My Workspace	1/5/23, 9:57:35 AM	General ⓘ
AdventureWorksProducts	-	②	Aquent vendor tea...	1/6/23, 2:53:04 PM	Confidential\Mi... ⓘ
Opportunity-Analysis-Custo...	Promoted	②	New Opportunity A...	12/2/20, 5:56:46 PM	Confidential\Mi... ⓘ
Opportunity-Analysis-City-Park	-	②	New Opportunity A...	12/2/20, 5:46:46 PM	Confidential\Mi... ⓘ
Opportunity Analysis Sample	-	②	New Opportunity A...	9/6/18, 3:06:05 PM	Confidential\Mi... ⓘ
SalesManager	-	②	Salesforce Analytics...	12/29/22, 12:53:12 PM	Confidential\Mi... ⓘ
COVID-19 US Tracking Sample	-	②	Covid-19 Tracking	2/12/21, 5:08:30 PM	Confidential\Mi... ⓘ

One of the datasets is labeled **Promoted**. Learn about that label in [Find an endorsed dataset](#), later in this article.

The datasets in this list meet at least one of the following conditions:

- The dataset is in a workspace that you're a member of. See [Considerations and limitations](#).
- You have Build permission for the dataset.
- The dataset is in your [My workspace](#).

### **ⓘ Note**

If you're a free user, you see only datasets in your [My workspace](#), or datasets for which you have Build permission that are in Premium-capacity workspaces.

When you select **Create**, you create a live connection to the dataset. The report creation experience opens with the full dataset available. You haven't made a copy of the dataset. The dataset still resides in its original location. You can use all tables and measures in the dataset to build your own reports. Row-level security (RLS) restrictions on the dataset are in effect, so you only see data you have permissions to see based on your RLS role.

You can save the report to the current workspace in the Power BI service, or publish the report to a workspace from Power BI Desktop. Power BI automatically creates an entry in the list of datasets if the report is based on a dataset outside of the workspace.

The entry shows information about the dataset, and a few select actions.

All	My data	Endorsed in your org	Filter by keyword		
Name	Owner	Location	Refreshed	Endorsement	Sensitivity
AdventureWorks	My Workspace		1/5/23, 9:57:35 AM	-	General

## Find an endorsed dataset

There are two different kinds of endorsed datasets. Dataset owners can *promote* a dataset that they recommend to you. Also, the Power BI admin can designate experts in your organization who can *certify* datasets for everyone to use. Promoted and certified datasets both display *badges* that you see both when looking for a dataset, and in the list of datasets in a workspace. The name of the person who certified a dataset is displayed in a tooltip during the dataset discovery experience. Hover over the **Certified** label and you see it.

- In the Power BI service: **Data hub**.
- In Power BI Desktop: **Get data > Power BI datasets**.

In Data hub, select **Endorsed in your org**.

All	My data	Endorsed in your org	Filter by keyword		
Name	Owner	Location	Refreshed	Endorsement	Sensitivity
Power BI Lifecycle Dataset	Power BI Lifecycle ...	Power BI Lifecycle ...	1/9/23, 8:43:28 PM	Certified	Confidential\Mi... ⓘ
Quality Shiproom	Azure Quality		11/6/22, 4:42:09 PM	Promoted	General ⓘ
Kusto Incident Analysis Dash...	WSD SIRV		1/10/23, 10:35:03 AM	Promoted	Confidential\An... ⓘ

## Next steps

- [Use datasets across workspaces](#)
- Questions? [Try asking the Power BI Community](#) ↗

# Copy reports from other workspaces

Article • 07/20/2023

When you find a report you like in a workspace or an app, you can make a copy of it and save it to a different workspace. Then you can modify your copy of the report, adding or deleting visuals and other elements. You don't have to worry about creating the data model - the copy of report will still reference the same dataset as the original report. And it's much easier to modify an existing report than it is to start from scratch. However, when you make an app from your workspace, sometimes you can't publish your copy of the report in the app. See [Considerations and limitations in the article "Use datasets across workspaces"](#) for details.

## Prerequisites

- To copy a report, you need a Pro or Premium Per User (PPU) license, even if the original report is in a workspace in a Premium capacity.
- To copy a report to another workspace, or to create a report in one workspace based on a dataset in another workspace, you need [Build permission for the dataset](#). If you have at least the Contributor role in the workspace where the dataset is located, you automatically have Build permission through your workspace role. You also need at least the Contributor role in the workspace where the report you're copying is located, and in the workspace where you want to create the copy of the report. See [Roles in workspaces](#) for details.

## Save a copy of a report in a workspace

1. In a workspace, find a report in the list. Open the **More options** menu and select [Save a copy](#).

	Name		Type	Owner
00	Opportunity report		Report	New Opportuni
	Opportunity report dataset		<ul style="list-style-type: none"><li>Analyze in Excel</li><li>Delete</li><li>Quick insights</li><li><b>Save a copy</b></li><li>Settings</li><li>View usage metrics report</li><li>View lineage</li><li>Create paginated report</li></ul>	IOD Administr

You only see the **Save a copy** option if you have [Build permission](#). Even if you have access to the workspace, you have to have Build permission for the dataset.

2. In **Save a copy of this report**, give the report a name and select the destination workspace.

## ⚙️ Save a copy of this report

Select the destination workspace to save a copy of this report into.

Save as

Opportunity report

Select a destination workspace

🔍 Search

My workspace

New Opportunity Analysis

New Retail Analysis

SalesWorkspace

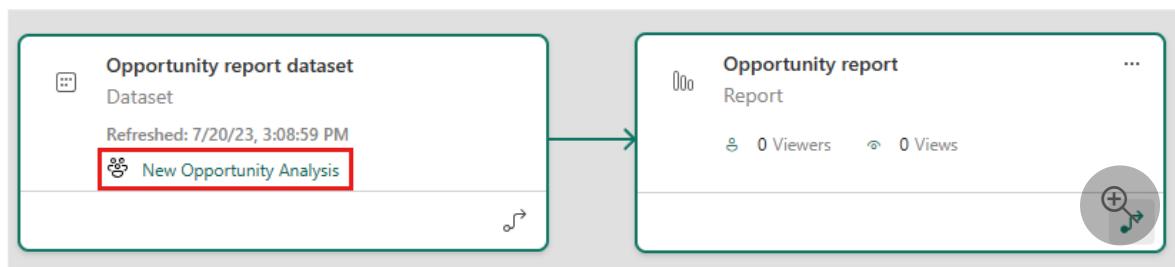
Save

Close

You can save the report to the current workspace or a different one in the Power BI service. You only see workspaces in which you're a member.

### 3. Select Save.

Power BI automatically creates a copy of the report in the workspace you selected. In the list view of that workspace, you won't see the referenced dataset if it is located in another workspace. To see the shared dataset, on the report copy in list view select **More options > View lineage**.

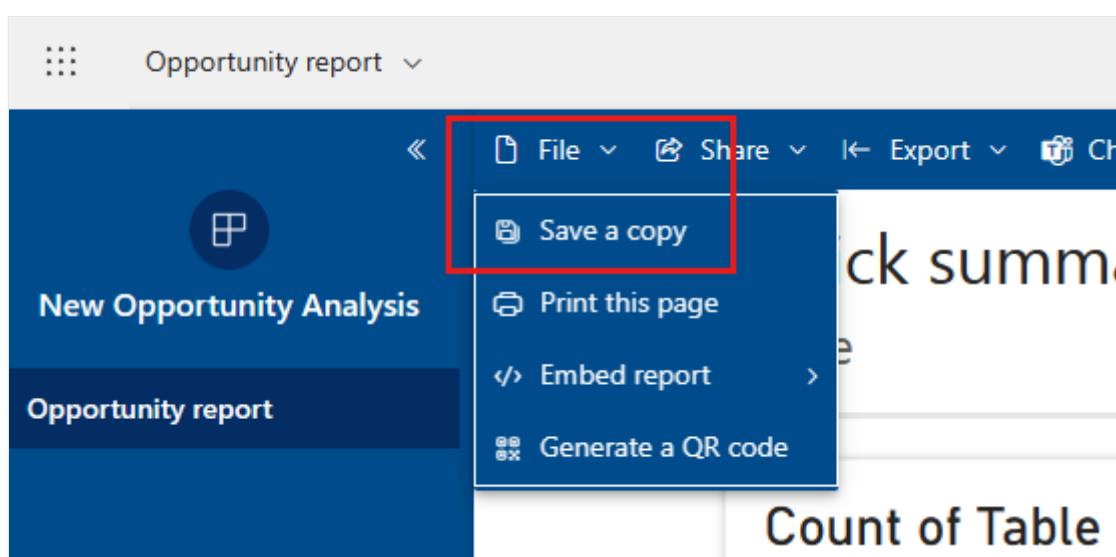


In lineage view, datasets that are located in other workspaces show the name of the workspace they're located in. This makes it easy to see which reports and dashboards use datasets that are outside the workspace.

See [Your copy of the report](#) in this article for more about the report and related dataset.

## Copy a report in an app

1. In an app, open the report you want to copy.
2. In the menu bar, select **File > Save a copy**.



You only see the **Save a copy** option if app permissions grant [Build permission](#) for the underlying dataset, and allow users to make copies of the report.

3. Give your report a name, select a destination workspace, and then select **Save**.

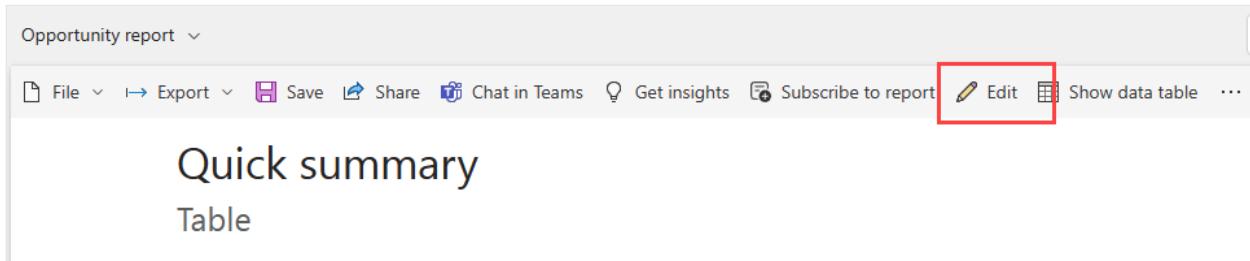
A screenshot of a "Save your report" dialog box. The dialog has a title bar "Save your report" and an "X" button. Inside, there is a text input field with the placeholder "Enter a name for your report \*". Below it is a dropdown menu labeled "Select a destination workspace" with the option "My workspace" selected. At the bottom are two buttons: a green "Save" button and a white "Cancel" button.

Your copy is automatically saved to the workspace you selected.

4. Select **Go to report** to open your copy.

## Your copy of the report

When you save a copy of the report, you create a live connection to the dataset, and you can open the report creation experience with the full dataset available.

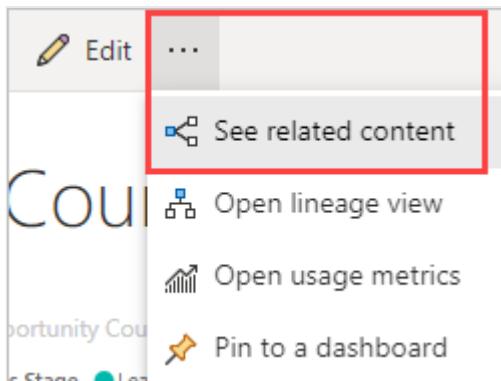


You haven't made a copy of the dataset. The dataset still resides in its original location. You can use all tables and measures in the dataset in your own report. Row-level security (RLS) restrictions on the dataset are in effect, so you only see data you have permissions to see based on your RLS role.

## View related datasets

When you have a report in one workspace based on a dataset in another workspace, you may need to know more about the dataset it's based on.

1. In the report, select **More options > See related content**.



2. The **Related content** dialog box shows all related items. In this list, the dataset looks like any other. There is no indication of where the dataset resides.

## Related content

### Opportunity Count Copy

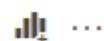
#### My workspace

##### DASHBOARDS

No related content

##### DATASETS

Opportunity Analysis Sample



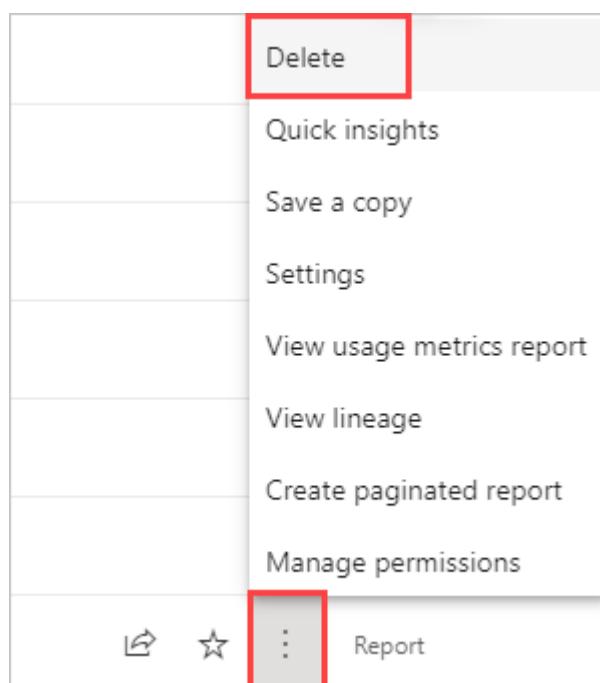
Last refresh: 9/6/2018, 3:06:05 PM

Show in lineage view

**Close**

## Delete a report copy

If you want to delete the copy of the report, in the list of reports in the workspace, hover over the report you want to delete, select **More options**, and choose **Delete**.



 **Note**

Deleting a report doesn't delete the dataset it is built on.

## Next steps

- [Use datasets across workspaces](#)
- Questions? [Try asking the Power BI Community](#) 

# Control the use of datasets across workspaces

Article • 08/24/2022

Using datasets across workspaces is a powerful way to drive data culture and data democratization within an organization. Still, if you're a Power BI admin, sometimes you want to restrict the flow of information within your Power BI tenant. With the tenant setting **Use datasets across workspaces**, you can restrict dataset reuse either completely or partially per security groups.

The screenshot shows the Power BI Admin portal interface. On the left, there's a sidebar with various navigation options: Usage metrics, Users, Audit logs, Tenant settings (which is highlighted), Capacity settings, Embed Codes, Organization visuals, Dataflow settings, and Workspaces. The main content area is titled 'Workspace settings' and contains two sections: 'Create workspaces (new workspace experience preview)' (Enabled for the entire organization) and 'Use datasets across workspaces' (Unapplied changes). A note below states: 'Users in the organization can use datasets across workspaces if they have the required Build permission.' There is a yellow toggle switch labeled 'Enabled'. Below the toggle, there's a section for 'Apply to:' with three options: 'The entire organization' (selected), 'Specific security groups', and 'Except specific security groups'. At the bottom are 'Apply' and 'Cancel' buttons.

Some of the effects of turning off this setting are listed below:

- The button to copy reports across workspaces isn't available.
- In a report based on a shared dataset, the **Edit report** button isn't available.
- In the Power BI service, the discovery experience only shows datasets in the current workspace.
- In Power BI Desktop, the discovery experience only shows datasets from workspaces where you're a member.
- In the Data hub, users see datasets that were shared with them outside of the workspace, but they can't interact with them.
- In Power BI Desktop, if users open a .pbix file with a live connection to a dataset outside any workspaces they are a member of, they see an error message asking them to connect to a different dataset.

# Provide a link for the certification process

As a Power BI admin, you can provide a URL for the [Learn more](#) link on the **Endorsement** setting page. See [Enable content certification](#) for detail. This link can go to documentation about your certification process. If you don't provide a destination for the [Learn more](#) link, by default it points to the [Endorse your content](#) article.

The screenshot shows the 'Admin portal' interface. On the left, a sidebar lists various tenant settings: Tenant settings (selected), Usage metrics, Users, Premium per user (preview), Audit logs, Capacity settings, Refresh summary, Embed Codes, Organizational visuals, Azure connections (preview), Workspaces, Custom branding, Protection metrics, and Featured content. The main content area is titled 'Certification'. It contains a note: 'Allow users in this org to certify datasets, dataflows, reports, and apps.' and a note: 'Note: When a user certifies an item, their contact details will be visible along with the certification badge.' Below this is a toggle switch labeled 'Enabled' which is set to 'On'. A section titled 'Specify URL for documentation page' contains a text input field labeled 'Enter URL'. Under 'Apply to:', there are two radio buttons: 'The entire organization' (unchecked) and 'Specific security groups' (checked). Below this is a text input field labeled 'Enter security groups'. At the bottom are 'Apply' and 'Cancel' buttons. A red box highlights the 'Certification' section and its configuration options.

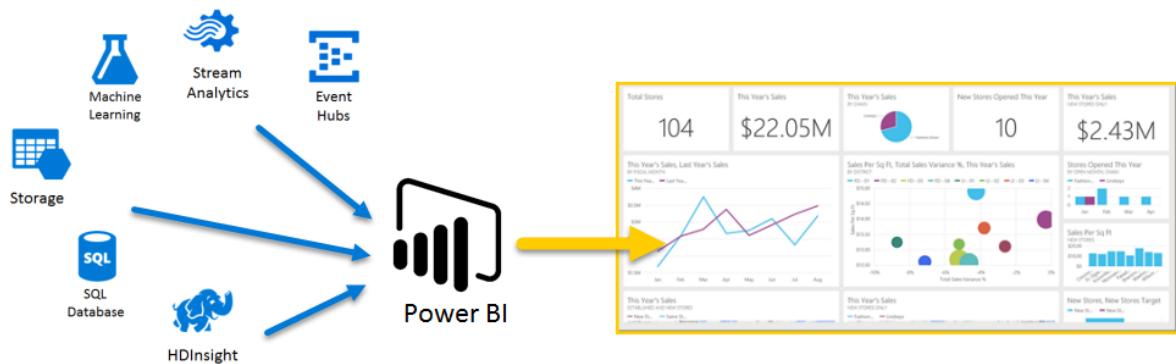
## Next steps

- [Use datasets across workspaces](#)
- Questions? [Try asking the Power BI Community](#) ↗

# Azure and Power BI

Article • 01/12/2023

With Azure services and Power BI, you can turn your data processing efforts into analytics and reports that provide real-time insights into your business. Whether your data processing is cloud-based or on-premises, straightforward, or complex, single-sourced or massively scaled, warehoused, or real-time, Azure and Power BI have the built-in connectivity and integration to bring your business intelligence efforts to life.



Power BI has a multitude of Azure connections available, and the business intelligence solutions you can create with those services are as unique as your business. You can connect as few as one Azure data source, or a handful, then shape and refine your data to build customized reports.

## Azure SQL Database and Power BI

You can start with a straightforward connection to an Azure SQL Database, and create reports to monitor the progress of your business. Using the [Power BI Desktop](#), you can create reports that identify trends and key performance indicators that move your business forward.

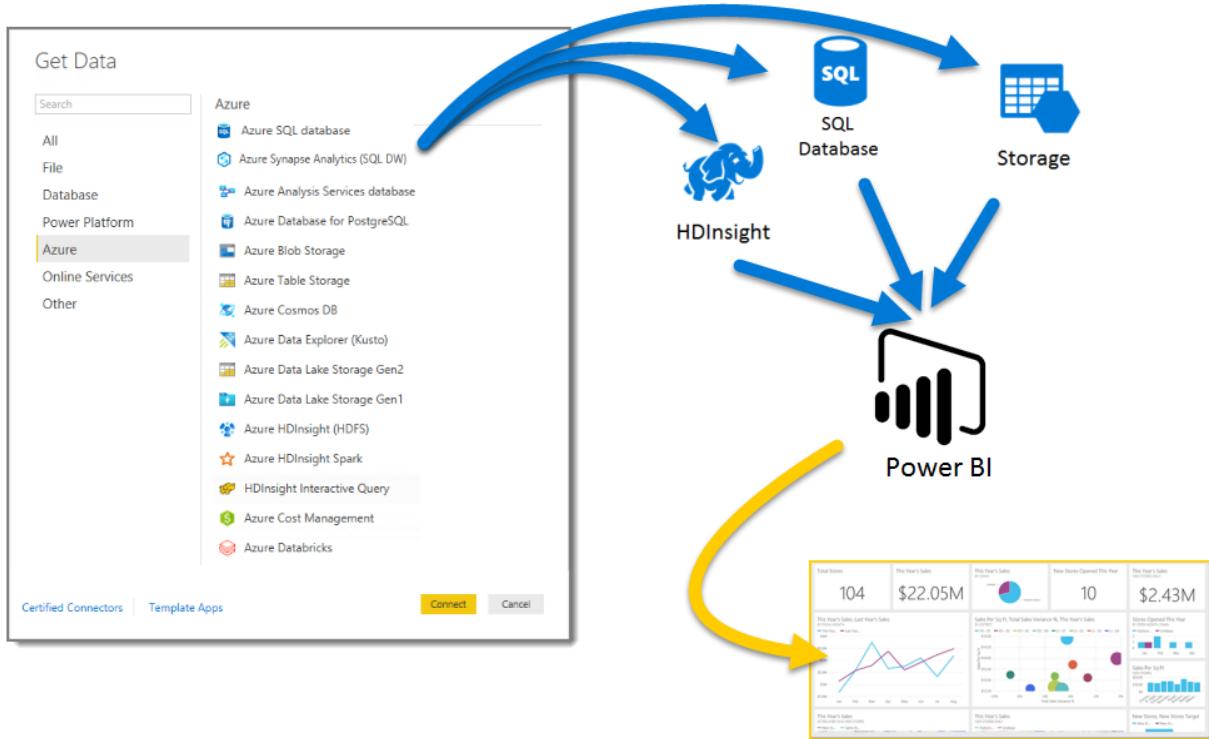


There's plenty more information for you to learn about [Azure SQL Database](#).

## Transform, shape, and merge your cloud data

Do you have more complex data, and all sorts of sources? No problem. With Power BI Desktop and Azure services, connections are just a tap of the **Get Data** dialog away. Within the same query you can connect to your Azure SQL Database, your Azure HDInsight data source, and your Azure Blob Storage or Azure Table Storage. Then select only the subsets within each that you need, and refine it from there.

You can create different reports for different audiences too, using the same data connections and even the same query. Just build a new report page, refine your visualizations for each audience, and watch it keep the business in the know.



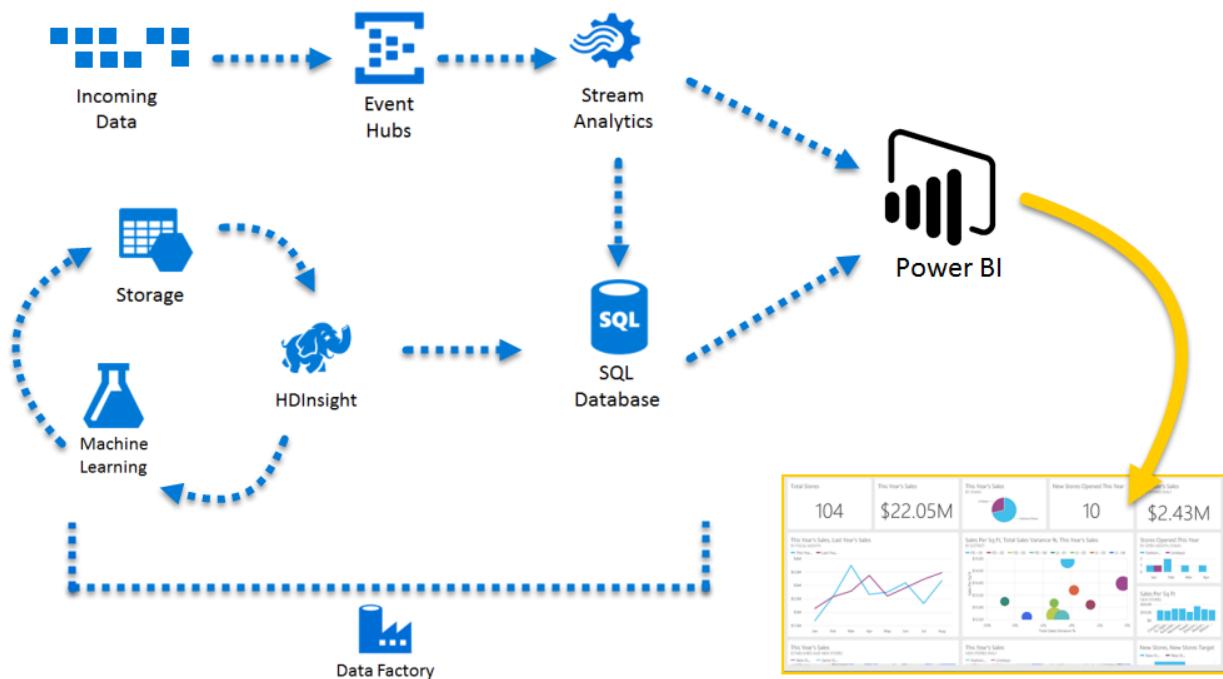
For more information, take a look at the following resources:

- [Azure SQL Database](#)
- [Azure HDInsight](#)
- [Azure Storage](#) (Blob storage and Table storage)

And for more information about Azure resources available in Power BI, see [Azure data sources](#).

## Get complex (and ahead) using Azure services and Power BI

You can expand as much as you need with Azure and Power BI. Harness multi-source data processing, make use of massive real-time systems, use [Stream Analytics](#) and [Event Hubs](#), and coalesce your varied SaaS services into business intelligence reports that give your business an edge.



## Context insights with Power BI Embedded analytics

Embed stunning, interactive data visualizations in applications, websites, portals, and more, to take advantage of your business data. With [Power BI Embedded as a resource in Azure](#), you can easily embed interactive reports and dashboards, so your users can enjoy consistent, high-fidelity experiences across devices. Power BI used with embedding analytics is to help you through your journey from Data to Knowledge to Insights to Actions. Furthermore, you can extend the value of Power BI and Azure also by embedding analytics in your [organization's internal applications and portals](#).

There's lots of information about Power BI APIs in the [Power BI Developer Portal](#).

For more information, see [Power BI Embedded](#).

## Embed your Power BI data within your app

Embed stunning, interactive data visualizations in applications, websites, portals, and more, to showcase your business data in context. Using [Power BI Embedded in Azure](#), you can easily embed interactive reports and dashboards, so your users can enjoy consistent, high-fidelity experiences across devices.

## What could you do with Azure and Power BI?

There are all sorts of scenarios where Azure and Power BI can be combined. The possibilities and opportunities are as unique as your business. For more information about Azure services, check out this [overview page](#), which describes Data Analytics Scenarios using Azure, and learn how to transform your data sources into intelligence that drives your business ahead.

# Power BI and Azure egress

Article • 02/06/2023

Data moving out, or *egress*, of Azure data centers can incur bandwidth charges. When using Power BI with Azure data sources, you can avoid Azure egress charges by making sure your Power BI service tenant is in the same region as your Azure data sources.

When your Power BI service tenant is deployed in the same Azure region as you deploy your data sources, you don't incur egress charges for scheduled refresh and DirectQuery interactions.

## Determining where your Power BI tenant is located

To find out where your Power BI tenant is located, see [Find the default region for your organization](#).

For Power BI Premium Multi-Geo customers, if your Power BI tenant isn't in the optimal location for some of your Azure-based data sources, you can deploy Power BI Premium Multi-Geo in the desired Azure region and benefit from having your Power BI tenant and Azure data sources in the same Azure region.

### ⓘ Note

Power BI Premium Per User (PPU) is not supported for Multi-Geo.

## Next steps

For more information about Power BI Premium or Multi-Geo, take a look at the following resources:

- [Azure bandwidth pricing details ↗](#)
- [What is Microsoft Power BI Premium?](#)
- [How to purchase Power BI Premium](#)
- [Multi-Geo support for Power BI Premium](#)
- [Where is my Power BI tenant located?](#)
- [Power BI Premium FAQ](#)

# Azure Synapse Analytics (formerly SQL Data Warehouse) with DirectQuery

Article • 02/22/2023

Azure Synapse Analytics (formerly SQL Data Warehouse) with DirectQuery allows you to create dynamic reports based on data and metrics you already have in Azure Synapse Analytics. With DirectQuery, queries are sent back to your Azure Synapse Analytics in real time as you explore the data. Real-time queries, combined with the scale of Synapse Analytics enables users to create dynamic reports in minutes against terabytes of data.

When you use the Azure Synapse Analytics connector:

- Specify the fully qualified server name when you connect (see details later in this article).
- Ensure firewall rules for the server are configured to "Allow access to Azure services".
- Every action such as selecting a column or adding a filter will directly query the data warehouse.
- Tiles are set to refresh approximately every 15 minutes and you don't need to schedule a refresh. You can adjust refresh in the Advanced settings when you connect.
- Q&A isn't available for DirectQuery datasets.
- Schema changes aren't picked up automatically.

These restrictions and notes can change as we continue to improve the experience. Steps to connect are in the next section.

## Build dashboards and reports in Power BI

### Important

We continually improve connectivity to Azure Synapse Analytics. For the best experience to connect to your Azure Synapse Analytics data source, use Power BI Desktop. After you've built your model and report, you can publish it to the Power BI service. The previously available direct connector for Azure Synapse Analytics in the Power BI service is no longer available.

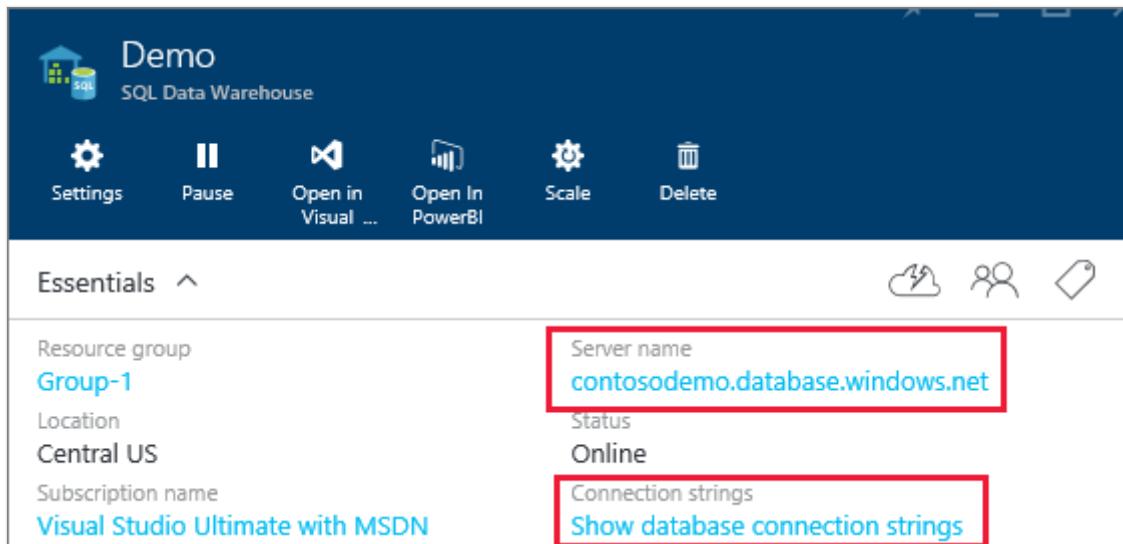
The easiest way to move between your Synapse Analytics and Power BI is to create reports in Power BI Desktop. To get started, [download and install Power BI Desktop](#).

# Connect through Power BI Desktop

You can connect to an Azure Synapse Analytics using the process described in [the Power Query article about Azure SQL Data Warehouse](#).

## Find Parameter Values

Your fully qualified server name and database name can be found in the Azure portal. Azure Synapse Analytics only has a presence in the Azure portal at this time.



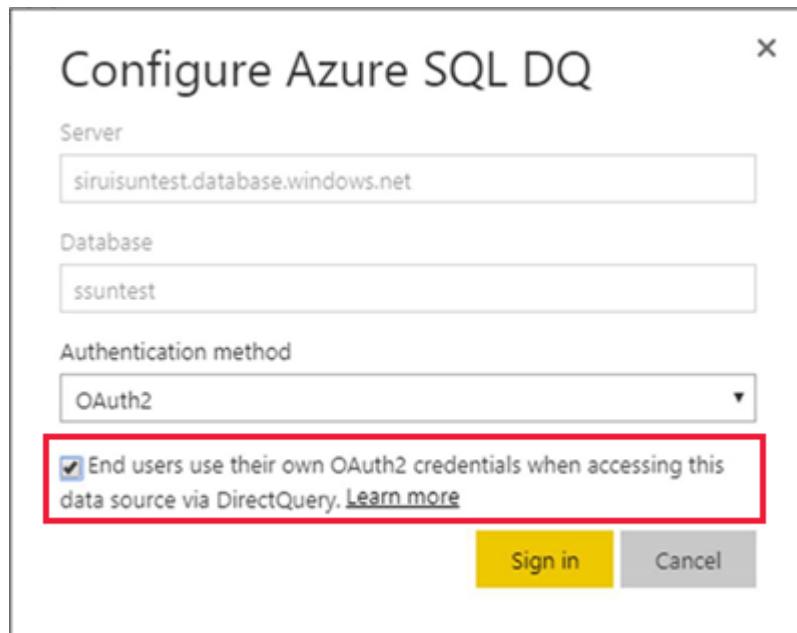
### ⓘ Note

If your Power BI tenant is in the same region as the Azure Synapse Analytics there will be no egress charges. To find where your Power BI tenant is located, see [Find the default region for your organization](#).

## Single sign-on

After you publish an Azure SQL DirectQuery dataset to the service, you can enable single sign-on (SSO) using Azure Active Directory (Azure AD) OAuth2 for your end users.

To enable SSO, go to settings for the dataset, open the **Data Sources** tab, and check the SSO box.



When the SSO option is enabled and your users access reports built atop the data source, Power BI sends their authenticated Azure AD credentials in the queries to the Azure SQL database or data warehouse. This option enables Power BI to respect the security settings that are configured at the data source level.

The SSO option takes affect across all datasets that use this data source. It does not affect the authentication method used for import scenarios.

#### ⓘ Note

For SSO to work properly, the dataset must be on the same tenant as the Azure SQL resource.

## Next steps

- [DirectQuery in Power BI](#)
- [What is Power BI?](#)
- [Data sources for the Power BI service](#)
- [What is dedicated SQL pool \(formerly SQL DW\) in Azure Synapse Analytics?](#)

More questions? [Ask the Power BI Community](#) ↗

# Azure SQL Database with DirectQuery

Article • 01/06/2023

Learn how you can connect directly to Azure SQL Database and create reports that use live data. You can keep your data at the source and not in Power BI.

With DirectQuery, queries are sent back to your Azure SQL Database as you explore the data in the report view. This experience is suggested for users who are familiar with the databases and entities they connect to.

## Important

This description assumes that Azure SQL database is not behind a VNET or has private link endpoint enabled.

## Notes:

- Specify the fully qualified server name when connecting (see below for more details).
- Ensure firewall rules for the database are configured to "[Allow access to Azure services](#)".
- Every action such as selecting a column or adding a filter will send a query back to the database.
- Tiles are refreshed every hour (refresh doesn't need to be scheduled). You can adjust how often to refresh in the Advanced settings when you connect.
- Schema changes aren't picked up automatically.
- Changing the data source connection string alias from `xxxx.database.windows.net` to `xxxx.domain.com` indicates to the Power BI service that it's an on-premises datasource and always requires a gateway connection to be established.

These restrictions and notes may change as we continue to improve the experiences. The steps to connect are detailed below.

## Important

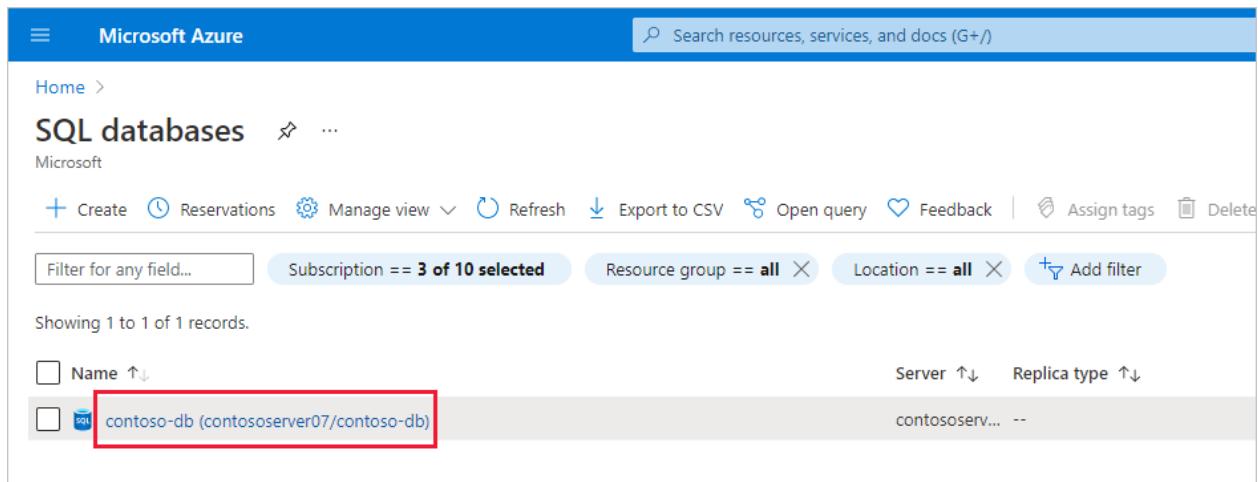
We have been improving our connectivity to Azure SQL Database. For the best experience to connect to your Azure SQL Database data source, use Power BI Desktop. Once you've built your model and report, you can publish it to the Power BI service. The direct connector for Azure SQL Database in the Power BI service is now deprecated.

# Power BI Desktop and DirectQuery

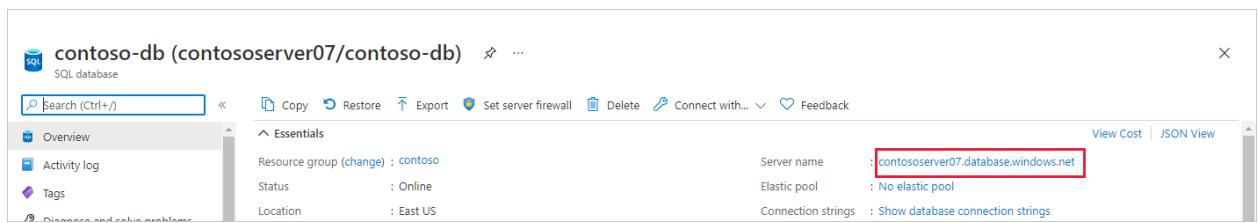
To connect to Azure SQL Database using DirectQuery, you must use Power BI Desktop. This approach provides more flexibility and capabilities. Reports created using Power BI Desktop can then be published to the Power BI service. To learn more about how to connect to Azure SQL Database in Power BI Desktop, see [Use DirectQuery in Power BI Desktop](#).

## Find parameter values

You can find your fully qualified server name and database name in the Azure portal.



The screenshot shows the Microsoft Azure portal interface. In the top navigation bar, 'Microsoft Azure' is selected. Below it, the 'SQL databases' section is shown under the 'Microsoft' category. The page includes a search bar and various filters like 'Subscription', 'Resource group', and 'Location'. The main table lists one record: 'contoso-db (contososerver07/contoso-db)'. The 'Server' column shows 'contososerver07' and the 'Replica type' column shows '--'. A red box highlights the database name 'contoso-db'.

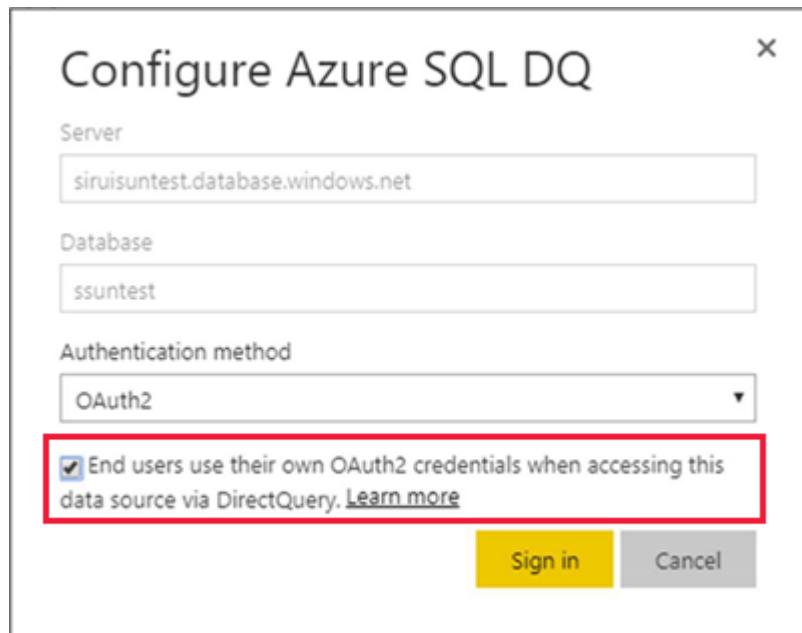


The screenshot shows the detailed view of the 'contoso-db' database. The left sidebar has options like 'Overview', 'Activity log', and 'Tags'. The main pane shows the 'Essentials' section with fields: 'Resource group (change)' set to 'contoso', 'Status' set to 'Online', 'Location' set to 'East US', 'Server name' set to 'contososerver07.database.windows.net', 'Elastic pool' set to 'No elastic pool', and 'Connection strings' with a link to 'Show database connection strings'. A red box highlights the 'Server name' field.

## Single sign-on

After you publish an Azure SQL DirectQuery dataset to the service, you can enable single sign-on (SSO) using Azure Active Directory (Azure AD) OAuth2 for your end users.

To enable SSO, go to settings for the dataset, open the **Data Sources** tab, and check the SSO box.



When the SSO option is enabled and your users access reports built atop the data source, Power BI sends their authenticated Azure AD credentials in the queries to the Azure SQL database or data warehouse. This option enables Power BI to respect the security settings that are configured at the data source level.

The SSO option takes affect across all datasets that use this data source. It does not affect the authentication method used for import scenarios.

#### ⓘ Note

For SSO to work properly, the dataset must be on the same tenant as the Azure SQL resource.

## Next steps

- [Use DirectQuery in Power BI Desktop](#)
- [What is Power BI?](#)
- [Data sources for the Power BI service](#)

More questions? [Try the Power BI community ↗](#)

# Data refresh in Power BI

Article • 07/10/2023

Power BI enables you to go from data to insight to action quickly, yet you must make sure the data in your Power BI reports and dashboards is recent. Knowing how to refresh the data is often critical in delivering accurate results.

This article describes the data refresh features of Power BI and their dependencies at a conceptual level. It also provides best practices and tips to avoid common refresh issues. The content lays a foundation to help you understand how data refresh works. For targeted step-by-step instructions to configure data refresh, refer to the tutorials and how-to guides listed in the Next steps section at the end of this article.

## Understanding data refresh

### [Embed Power BI content with service principal and an application secret](#)

Whenever you refresh data, Power BI must query the underlying data sources, possibly load the source data into a dataset, and then update any visualizations in your reports or dashboards that rely on the updated dataset. The entire process consists of multiple phases, depending on the storage modes of your datasets, as explained in the following sections.

To understand how Power BI refreshes your datasets, reports, and dashboards, you must be aware of the following concepts:

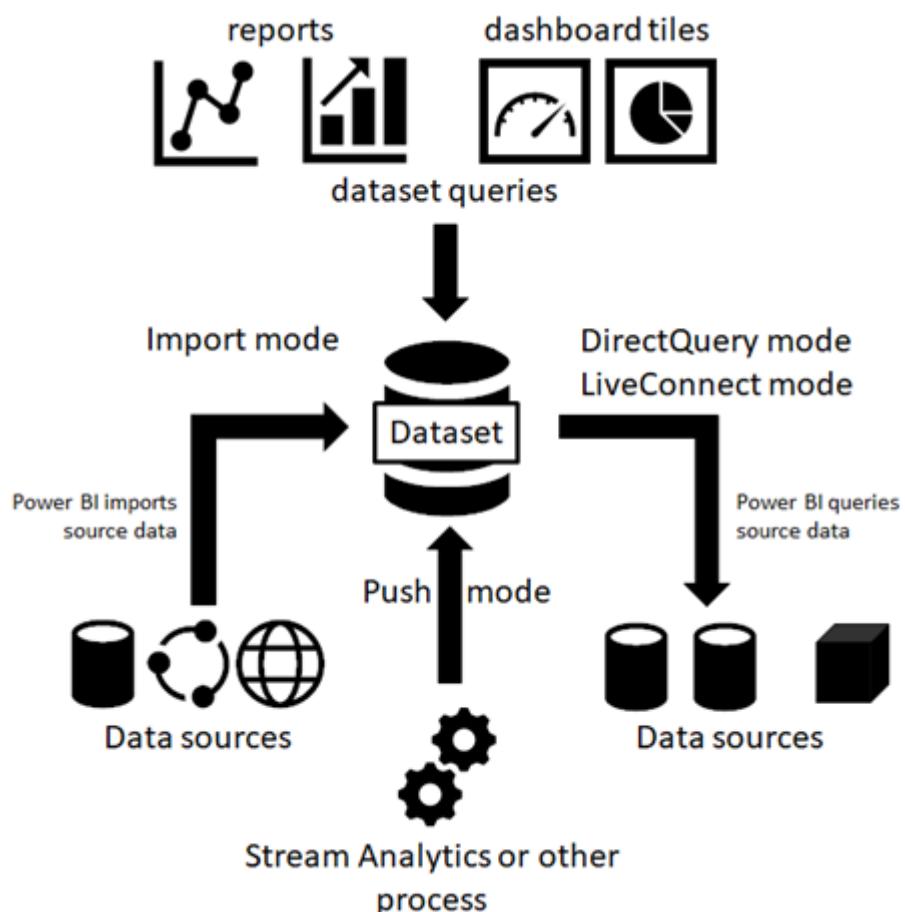
- **Storage modes and dataset types:** The storage modes and dataset types that Power BI supports have different refresh requirements. You can choose between reimporting data into Power BI to see any changes that occurred or querying the data directly at the source.
- **Power BI refresh types:** Regardless of dataset specifics, knowing the various refresh types can help you understand where Power BI might spend its time during a refresh operation. And combining these details with storage mode specifics helps to understand what exactly Power BI performs when you select **Refresh now** for a dataset.

## Storage modes and dataset types

A Power BI dataset can operate in one of the following modes to access data from various data sources. For more information, see [Storage mode in Power BI Desktop](#).

- Import mode
- DirectQuery mode
- LiveConnect mode
- Push mode

The following diagram illustrates the different data flows, based on storage mode. The most significant point is that only Import mode datasets require a source data refresh. They require refresh because only this type of dataset imports data from its data sources, and the imported data might be updated on a regular or ad-hoc basis. DirectQuery datasets and datasets in LiveConnect mode to Analysis Services don't import data; they query the underlying data source with every user interaction. Datasets in push mode don't access any data sources directly but expect you to push the data into Power BI. Dataset refresh requirements vary depending on the storage mode/dataset type.



## Datasets in Import mode

Power BI imports the data from the original data sources into the dataset. Power BI report and dashboard queries submitted to the dataset return results from the imported tables and columns. You might consider such a dataset a point-in-time copy. Because

Power BI copies the data, you must refresh the dataset to fetch changes from the underlying data sources.

When a dataset is refreshed, it's either fully refreshed or partially refreshed. Partial refresh will take place in datasets that have tables with an [incremental refresh](#) policy. In these datasets, only a subset of the table partitions are refreshed. In addition, advanced users can use the [XMLA endpoint](#) to refresh specific partitions in any dataset.

The amount of memory required to refresh a dataset depends on whether you're performing a full or partial refresh. During the refresh, a copy of the dataset is kept to handle queries to the dataset. This means that if you're performing a full refresh, you'll need twice the amount of memory the dataset requires.

We recommend that you plan your capacity usage to ensure that the extra memory needed for dataset refresh, is accounted for. Having enough memory prevents refresh issues that can occur if your datasets require more memory than available, during refresh operations. To find out how much memory is available for each dataset on a Premium capacity, refer to the [Capacities and SKUs](#) table.

For more information about large datasets in Premium capacities, see [large datasets](#).

## Datasets in DirectQuery mode

Power BI doesn't import data over connections that operate in DirectQuery mode. Instead, the dataset returns results from the underlying data source whenever a report or dashboard queries the dataset. Power BI transforms and forwards the queries to the data source.

### Note

Live connection reports submit queries to the capacity or Analysis Services instance that hosts the dataset or the model. When using external analysis services such as SQL Server Analysis Services (SSAS) or Azure Analysis Services (AAS), resources are consumed outside of Power BI.

Because Power BI doesn't import the data, you don't need to run a data refresh. However, Power BI still performs tile refreshes and possibly report refreshes, as the next section on refresh types explains. A tile is a report visual pinned to a dashboard, and dashboard tile refreshes happen about every hour so that the tiles show recent results. You can change the schedule in the dataset settings, as in the screenshot below, or force a dashboard update manually by using the **Refresh now** option.

General Alerts Subscriptions Dashboards Datasets Workbooks Reports Dataflows App

Settings for sampleAutoAggs

[View dataset](#)

This dataset has been configured by [chkt@microsoft.com](mailto:chkt@microsoft.com).

[Refresh history](#)

► Dataset description

► Gateway connection

► Data source credentials

AzureDataExplorer [Edit credentials](#) [Show in lineage view](#)

► Sensitivity label

► Parameters

► Query Caching

► Optimize performance

To improve the performance of exploring reports, enable automatic aggregations and estimate how caching can improve query response times.

**Enable automatic aggregations** ⓘ

To speed up exploring reports, cache some of the data for queries—your reports will run faster, and visuals with cached data will load more quickly. [Learn more](#)

Off

On

Automatic dashboard tile refresh

Adjust the automatic refresh frequency of each dashboard tile to suit your needs, or manually refresh a tile by selecting More. [Learn more](#)

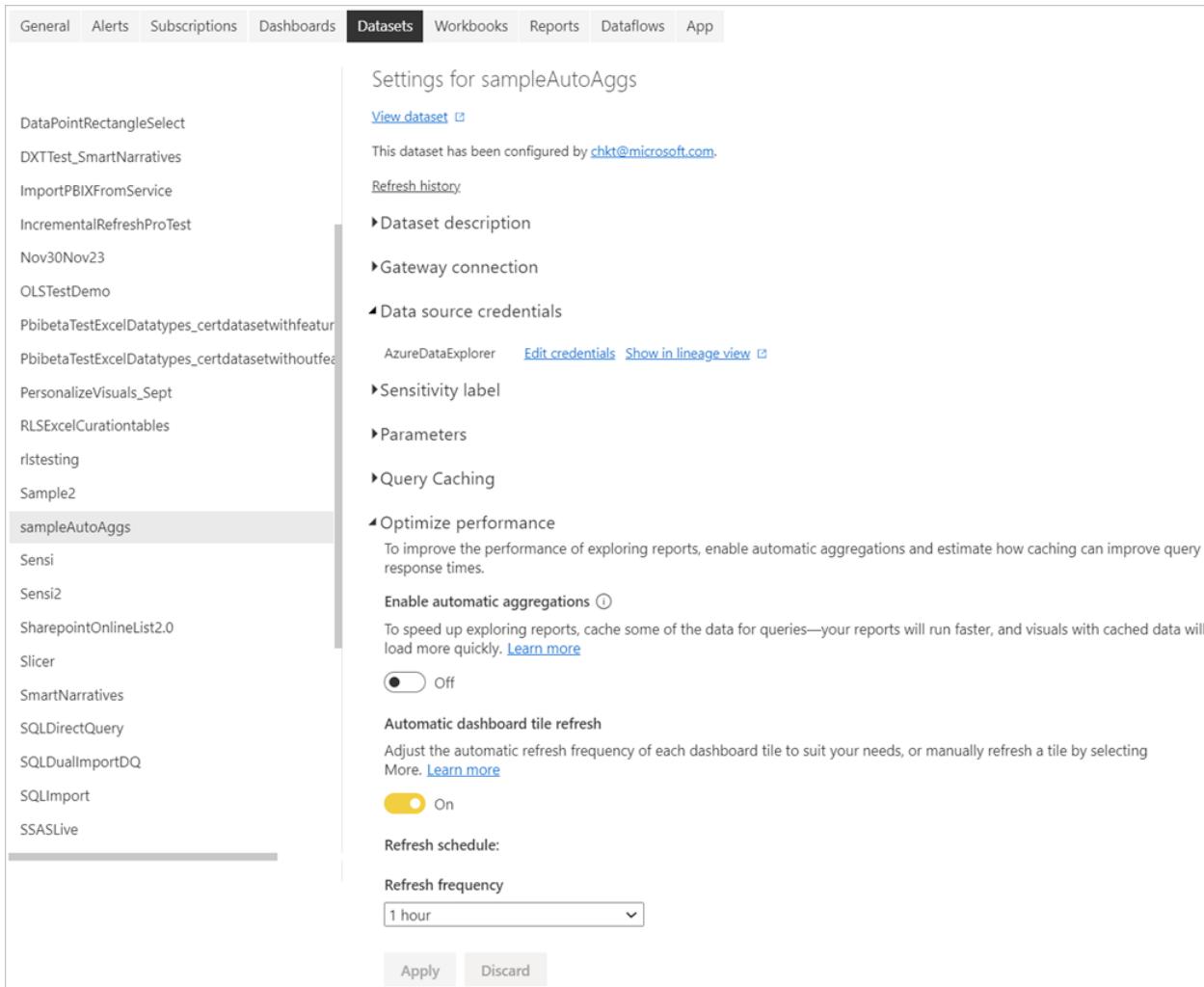
On

Refresh schedule:

Refresh frequency

1 hour

Apply Discard



## (!) Note

- Datasets in import mode and composite datasets that combine import mode and DirectQuery mode don't require a separate tile refresh, because Power BI refreshes the tiles automatically during each scheduled or on-demand data refresh. Datasets that are updated based on the XMLA endpoint will only clear the cached tile data (invalidate cache). The tile caches aren't refreshed until each user accesses the dashboard. For import models, you can find the refresh schedule in the "Scheduled refresh" section of the **Datasets** tab. For composite datasets, the "Scheduled refresh" section is located in the **Optimize Performance** section.
- Power BI does not support cross-border live connections to Azure Analysis Services (AAS) in a sovereign cloud.

## Push datasets

Push datasets don't contain a formal definition of a data source, so they don't require you to perform a data refresh in Power BI. You refresh them by pushing your data into the dataset through an external service or process, such as Azure Stream Analytics. This is a common approach for real-time analytics with Power BI. Power BI still performs cache refreshes for any tiles used on top of a push dataset. For a detailed walkthrough, see [Tutorial: Stream Analytics and Power BI: A real-time analytics dashboard for streaming data](#).

## Power BI refresh types

A Power BI refresh operation can consist of multiple refresh types, including data refresh, OneDrive refresh, refresh of query caches, tile refresh, and refresh of report visuals. While Power BI determines the required refresh steps for a given dataset automatically, you should know how they contribute to the complexity and duration of a refresh operation. For a quick reference, refer to the following table.

Storage mode	Data refresh	OneDrive refresh	Query caches	Tile refresh	Report visuals
Import	Scheduled and on-demand	Yes, for connected datasets	If enabled on Premium capacity	Automatically and on-demand	No
DirectQuery	Not applicable	Yes, for connected datasets	If enabled on Premium capacity	Automatically and on-demand	No
LiveConnect	Not applicable	Yes, for connected datasets	If enabled on Premium capacity	Automatically and on-demand	Yes
Push	Not applicable	Not applicable	Not practical	Automatically and on-demand	No

Another way to consider the different refresh types is what they impact and where you can apply them. Changes in data source table structure, or schema, such as a new, renamed, or removed column can only be applied in Power BI Desktop, and in the Power BI service they can cause the refresh to fail. For a quick reference on what they impact, refer to the following table.

	Refresh of report visuals	Data refresh	Schema refresh
What do the different	Queries used to populate visuals are refreshed.	Data is refreshed from the data source.	Any data source table structure change since previous refresh will

	<b>Refresh of report visuals</b>	<b>Data refresh</b>	<b>Schema refresh</b>
refresh types do?	For visuals using DirectQuery tables the visual will query to get the latest data from the data source.	Doesn't apply to DirectQuery tables as they are at the visual level and rely on refresh of report visuals.	show.  For example: To show a new column added to a Power BI Dataflow or SQL Database view.
	For visuals using imported tables the visual will only query data already imported to the dataset on the last data refresh.	For imported tables the data is refreshed from the source.	Applies to both imported and DirectQuery tables.

In **Power BI Desktop** refresh of report visuals, data refresh, and schema refresh all happen together using

- Home ribbon > **Refresh** button
- Home ribbon > **Transform data** > **Close & Apply** button
- The context menu (right-click or select the ellipsis) on any table then choosing **Refresh data**

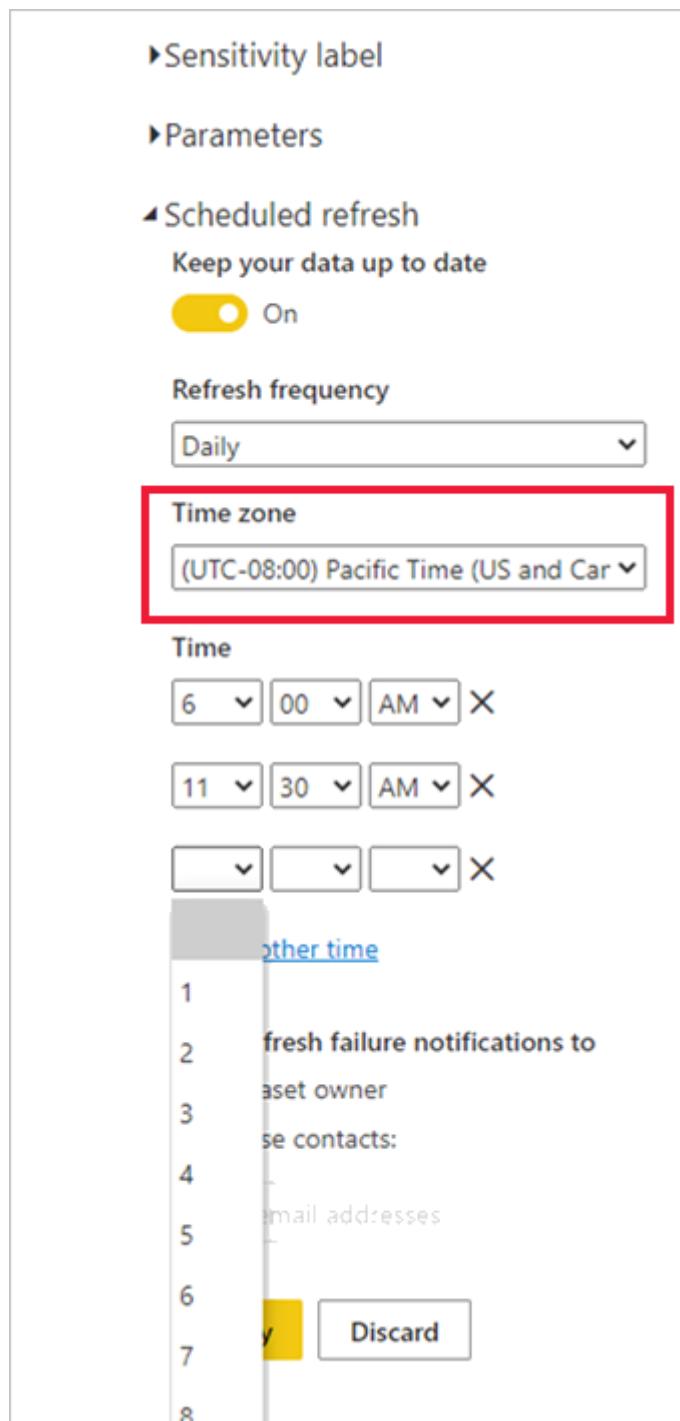
These refresh types cannot always be applied independently, and where you can apply them is different in Power BI Desktop and the Power BI service. For a quick reference, refer to the following table.

	<b>Refresh of report visuals</b>	<b>Data refresh</b>	<b>Schema refresh</b>
In Power BI Desktop	<ul style="list-style-type: none"> <li>• View ribbon &gt; <b>Performance Analyzer</b> button &gt; <b>Refresh visuals</b></li> <li>• Creating and changing visuals causing a DAX query to run</li> <li>• When <b>Page Refresh</b> is turned on (DirectQuery only)</li> <li>• Opening the PBIX file</li> </ul>	Not available independently from other refresh types	Not available independently from other refresh types
In the Power BI service	<ul style="list-style-type: none"> <li>• When the browser loads or reloads the report</li> <li>• Clicking the <b>Refresh Visuals</b> top right</li> </ul>	<ul style="list-style-type: none"> <li>• Scheduled refresh</li> <li>• Refresh now</li> <li>• Refresh a Power BI dataset from Power Automate</li> </ul>	Not available

	<b>Refresh of report visuals</b>	<b>Data refresh</b>	<b>Schema refresh</b>
	<ul style="list-style-type: none"> <li>menu bar button</li> <li>• Clicking the Refresh button in edit mode</li> <li>• When <a href="#">Page Refresh</a> is turned on (DirectQuery only)</li> </ul>	<ul style="list-style-type: none"> <li>• Processing the table from SQL Server Management Studio (Premium)</li> </ul>	
Keep in mind	For example, if you open a report in the browser, then the scheduled refresh performs a data refresh of the imported tables, the report visuals in the open browser won't update until a refresh of report visuals is initiated.	Data refresh on the Power BI service will fail when the source column or table is renamed or removed. It fails because the Power BI service doesn't also include a schema refresh. To correct this error, a schema refresh needs to happen in Power BI Desktop and the dataset republished to the service.	A renamed or removed column or table at the data source will be updated with a schema refresh in Power BI Desktop, but it can break visuals and DAX expressions (measures, calculated columns, row level security, etc.), as well as remove relationships, that are dependent on those columns or tables.

## Data refresh

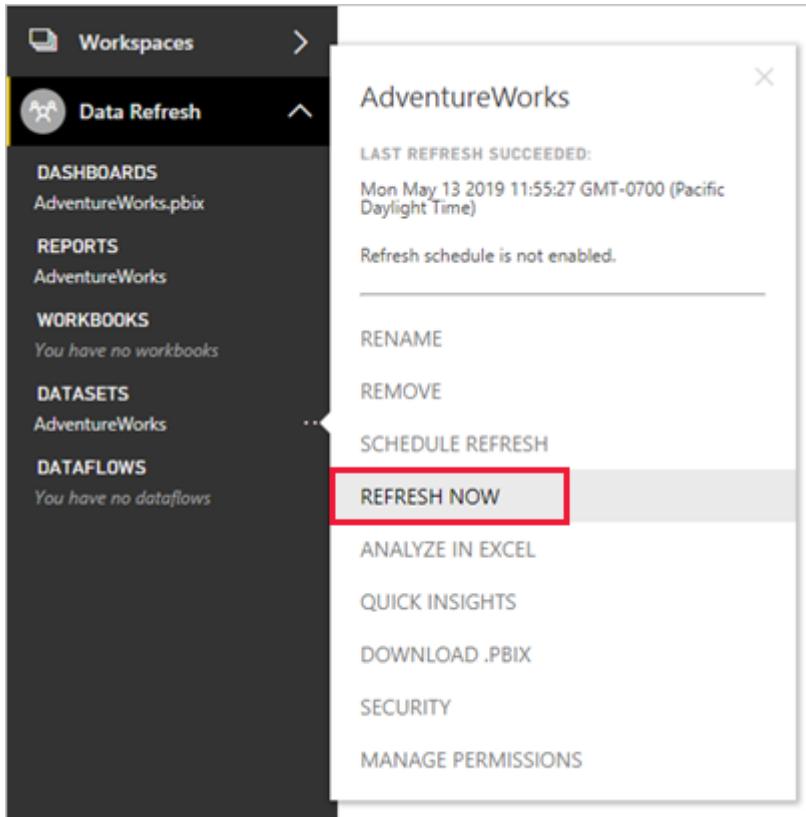
For Power BI users, refreshing data typically means importing data from the original data sources into a dataset, either based on a refresh schedule or on-demand. You can perform multiple dataset refreshes daily, which might be necessary if the underlying source data changes frequently. Power BI limits datasets on shared capacity to eight scheduled daily dataset refreshes. The eight time values are stored in the backend database and are based on the *local time* zone that was selected on the Dataset Settings page. The scheduler checks which model should be refreshed and at what time(s). The quota of eight refreshes resets daily at 12:01 a.m. local time.



If the dataset resides on a Premium capacity, you can schedule up to 48 refreshes per day in the dataset settings. For more information, see [Configure scheduled refresh](#) later in this article. Datasets on a Premium capacity with the [XMLA endpoint](#) enabled for read-write support unlimited refresh operations when configured programmatically with TMSL or PowerShell.

It's also important to call out that the shared-capacity limitation for daily refreshes applies to both scheduled refreshes and API refreshes combined. You can also trigger an on-demand refresh by selecting **Refresh now** in the dataset menu, as the following screenshot depicts. On-demand refreshes aren't included in the refresh limitation. Also note that datasets on a Premium capacity don't impose limitations for API refreshes. If

you're interested in building your own refresh solution by using the Power BI REST API, see [Datasets - Refresh Dataset](#).



### ! Note

Data refreshes must complete in less than 2 hours on shared capacity. If your datasets require longer refresh operations, consider moving the dataset onto a Premium capacity. On Premium, the maximum refresh duration is 5 hours.

## OneDrive refresh

If you created your datasets and reports based on a Power BI Desktop file, Excel workbook, or comma separated value (.csv) file on OneDrive or SharePoint Online, Power BI performs another type of refresh, known as OneDrive refresh. For more information, see [Get data from files for Power BI](#).

Unlike a dataset refresh during which Power BI imports data from a data source into a dataset, OneDrive refresh synchronizes datasets and reports with their source files. By default, Power BI checks about every hour if a dataset connected to a file on OneDrive or SharePoint Online requires synchronization.

Power BI performs refresh based on an item ID in OneDrive, so be thoughtful when considering updates versus replacement. When you set a OneDrive file as the data source, Power BI references the item ID of the file when it performs the refresh. Consider

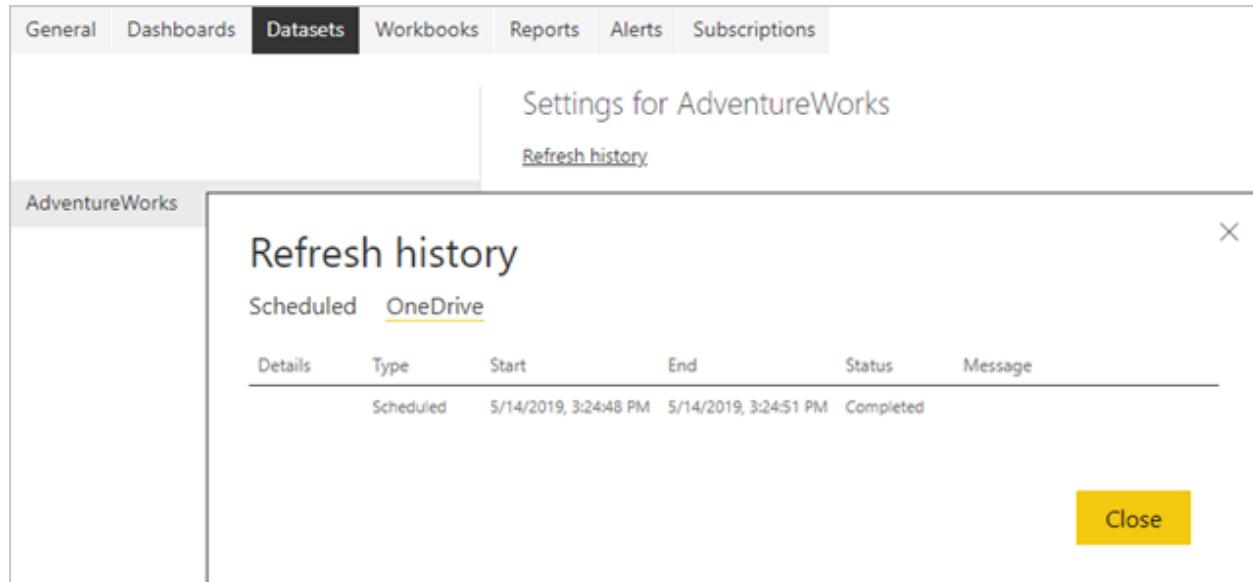
the following scenario: you have a master file *A* and a production copy of that file *B*, and you configure OneDrive refresh for file *B*. If you then *copy* file *A* over file *B*, the copy operation deletes the old file *B* and creates a new file *B* with a different item ID, which breaks OneDrive refresh. To avoid that situation, you can instead upload and replace file *B*, which keeps its same item ID.

You can move the file to another location (using drag and drop, for example) and refresh will continue to work because Power BI still knows the file ID. However, if you copy that file to another location, a new instance of the file and a new fileId is created. Therefore, your Power BI file reference is no longer valid and refresh will fail.

### Note

It can take Power BI up to 60 minutes to refresh a dataset, even once the sync has completed on your local machine and after you've used *Refresh now* in the Power BI service.

To review past synchronization cycles, check the OneDrive tab in the refresh history. The following screenshot shows a completed synchronization cycle for a sample dataset.



The screenshot shows the Power BI interface with the 'Datasets' tab selected. A modal window titled 'Refresh history' is open for the 'AdventureWorks' dataset. The window displays a single row of refresh history:

Details	Type	Start	End	Status	Message
Scheduled	OneDrive	5/14/2019, 3:24:48 PM	5/14/2019, 3:24:51 PM	Completed	

A yellow 'Close' button is visible at the bottom right of the modal.

As the above screenshot shows, Power BI identified this OneDrive refresh as a **Scheduled** refresh, but it isn't possible to configure the refresh interval. You can only deactivate OneDrive refresh in the dataset's settings. Deactivating refresh is useful if you don't want your datasets and reports in Power BI to pick up any changes from the source files automatically.

The dataset settings page only shows the **OneDrive Credentials** and **OneDrive refresh** sections if the dataset is connected to a file in OneDrive or SharePoint Online, as in the

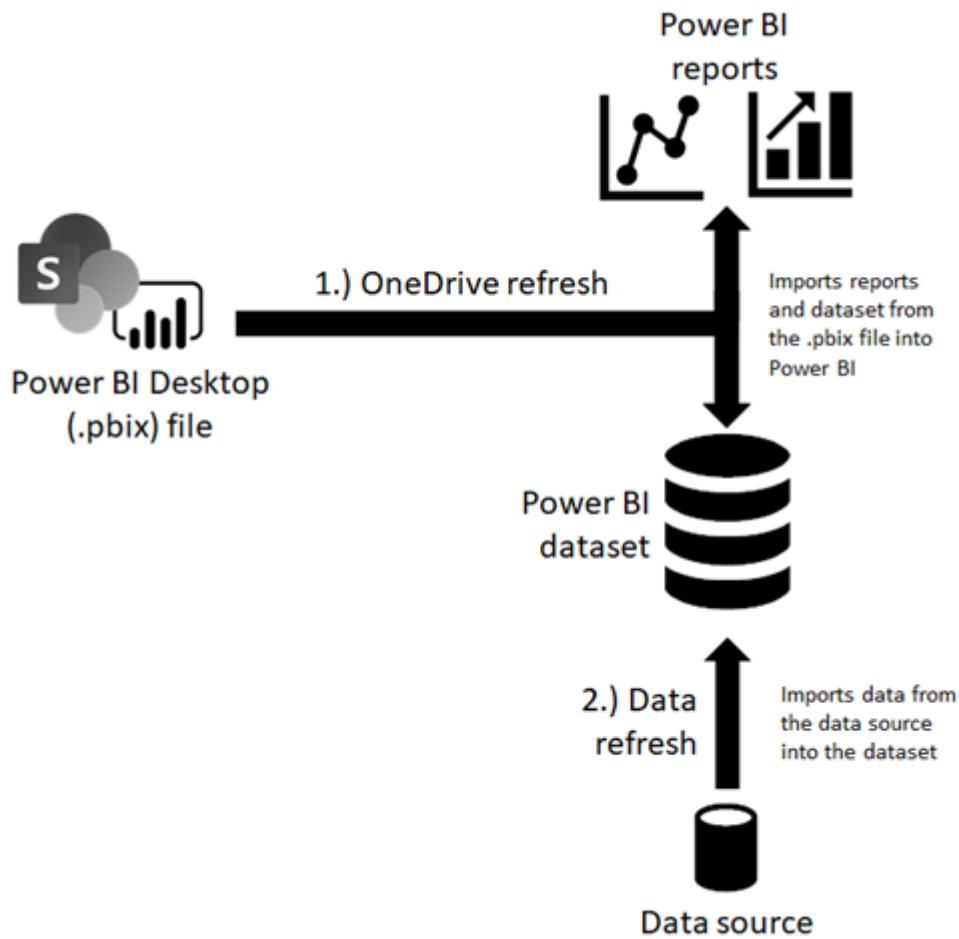
following screenshot. Datasets that aren't connected to sources file in OneDrive or SharePoint Online don't show these sections.

The screenshot shows the 'Datasets' tab selected in the navigation bar. A dataset named 'AdventureWorks' is selected on the left. The main pane displays 'Settings for AdventureWorks' with the following sections:

- [Refresh history](#)
- ▶ Gateway connection
- ▶ Data source credentials
- ▶ Parameters
- ▶ Scheduled refresh
- ▲ OneDrive Credentials
  - OneDrive - Personal [Edit credentials](#)
- ▲ OneDrive refresh
  - By default, OneDrive updates files hourly. Do you want your files to be kept up to date?  
 On
- ▶ Server settings
- ▶ Q&A and Cortana
- ▶ Featured Q&A questions

If you disable OneDrive refresh for a dataset, you can still synchronize your dataset on-demand by selecting **Refresh now** in the dataset menu. As part of the on-demand refresh, Power BI checks if the source file on OneDrive or SharePoint Online is newer than the dataset in Power BI and synchronizes the dataset if so. The **Refresh history** lists these activities as on-demand refreshes on the **OneDrive** tab.

Keep in mind that OneDrive refresh doesn't pull data from the original data sources. OneDrive refresh simply updates the resources in Power BI with the metadata and data from the .pbix, .xlsx, or .csv file, as the following diagram illustrates. To ensure that the dataset has the most recent data from the data sources, Power BI also triggers a data refresh as part of an on-demand refresh. You can verify this in the **Refresh history** if you switch to the **Scheduled** tab.



If you keep OneDrive refresh enabled for a OneDrive or SharePoint Online-connected dataset and you want to perform data refresh on a scheduled basis, make sure you configure the schedule so that Power BI performs the data refresh after the OneDrive refresh. For example, if you created your own service or process to update the source file in OneDrive or SharePoint Online every night at 1 am, you could configure scheduled refresh for 2:30 am to give Power BI enough time to complete the OneDrive refresh before starting the data refresh.

## Refresh of query caches

If your dataset resides on a Premium capacity, you might be able to improve the performance of any associated reports and dashboards by enabling query caching, as in the following screenshot. Query caching instructs the Premium capacity to use its local caching service to maintain query results, avoiding having the underlying data source compute those results. For more information, see [Query caching in Power BI Premium](#).

The screenshot shows the 'Datasets' tab selected in the top navigation bar. On the left, a sidebar lists datasets: 'AdventureWorksCloud' (selected), 'AdventureworksLT', 'AdventureworksDW', and 'AdventureworksLTReport'. The main pane displays 'Settings for AdventureWorksCloud' with a 'Refresh history' link. A tree view on the right shows sections: 'Gateway connection', 'Data source credentials', 'Parameters', and 'Query Caching' (which is expanded). Under 'Query Caching', three radio button options are shown: 'Capacity default: query caching Off' (unselected), 'off: Do not cache query results for this dataset' (unselected), and 'On: Speed up reports by using previously saved query results.' (selected). Below the radio buttons are 'Apply' and 'Discard' buttons.

Following a data refresh, however, previously cached query results are no longer valid. Power BI discards these cached results and must rebuild them. For this reason, query caching might not be as beneficial for reports and dashboards associated with datasets that you refresh often, for example 48 times per day.

## Refresh of report visuals

This refresh process is less important because it's only relevant for live connections to Analysis Services. For these connections, Power BI caches the last state of the report visuals so that when you view the report again, Power BI doesn't have to query the Analysis Services tabular model. When you interact with the report, such as by changing a report filter, Power BI queries the tabular model and updates the report visuals automatically. If you suspect that a report is showing stale data, you can also select the Refresh button of the report to trigger a refresh of all report visuals, as the following screenshot illustrates.

The screenshot shows the Power BI desktop application. On the left is a navigation sidebar with icons for Home, Favorites, Recent, Apps, Shared with me, Workspaces, and Data Refresh. The main area displays a data grid titled 'AdventureWorks' with columns 'LastName' and 'OrderQuantity'. A red box highlights the 'Refresh' button in the top right corner of the main window.

LastName	OrderQuantity
Adams	252
Agbonile	3
Alan	19
Albrecht	2
Alexander	348
Allen	205
Alonso	296
Alvarez	345
Anand	249
Anderesen	320
Anderson	252
Arthur	72
Arun	187
Ashe	100
Total	60398

Only pinned visuals are refreshed, not pinned live pages. To refresh a pinned live page, you can use the browser's Refresh button.

## Review data infrastructure dependencies

Regardless of storage modes, no data refresh can succeed unless the underlying data sources are accessible. There are three main data access scenarios:

- A dataset uses data sources that reside on-premises
- A dataset uses data sources in the cloud
- A dataset uses data from both, on-premises and cloud sources

## Connecting to on-premises data sources

If your dataset uses a data source that Power BI can't access over a direct network connection, you must configure a gateway connection for this dataset before you can enable a refresh schedule or perform an on-demand data refresh. For more information about data gateways and how they work, see [What are on-premises data gateways?](#)

You have the following options:

- Choose an enterprise data gateway with the required data source definition
- Deploy a personal data gateway

### Note

You can find a list of data source types that require a data gateway in the article [Manage your data source - Import/Scheduled Refresh](#).

## Using an enterprise data gateway

Microsoft recommends using an enterprise data gateway instead of a personal gateway to connect a dataset to an on-premises data source. Make sure the gateway is properly configured, which means the gateway must have the latest updates and all required data source definitions. A data source definition provides Power BI with the connection information for a given source, including connection endpoints, authentication mode, and credentials. For more information about managing data sources on a gateway, see [Manage your data source - import/scheduled refresh](#).

Connecting a dataset to an enterprise gateway is relatively straightforward if you're a gateway administrator. With admin permissions, you can promptly update the gateway and add missing data sources, if necessary. In fact, you can add a missing data source to your gateway straight from the dataset settings page. Expand the toggle button to view the data sources and select the **Add to gateway** link, as in the following screenshot. If you aren't a gateway administrator, on the other hand, you must contact a gateway admin to add the required data source definition.

### Note

Only gateway admins can add data sources to a gateway. Also make sure your gateway admin adds your user account to the list of users with permissions to use the data source. The dataset settings page only lets you select an enterprise gateway with a matching data source that you have permission to use.

The screenshot shows the Power BI Settings interface with the 'Datasets' tab selected. On the left, a sidebar lists various datasets. The main area displays the 'Settings for AdventureWorksProducts'. It includes a note about the last refresh and a section for 'Gateway connection'. A table lists the gateway, department, contact information, status, and actions. The 'Actions' column for the 'Enterprise Data Gateway' row has a dropdown menu open, with the 'Maps to:' option also highlighted by a red box. Below the table, there's a note about personal gateways and a 'Install now' button.

Make sure you map the correct data source definition to your data source. As the above screenshot illustrates, gateway admins can create multiple definitions on a single gateway connecting to the same data source, each with different credentials. In the example shown, a dataset owner in the Sales department would choose the AdventureWorksProducts-Sales data source definition while a dataset owner in the Support department would map the dataset to the AdventureWorksProducts-Support data source definition. If the names of the data source definition aren't intuitive, contact your gateway admin to clarify which definition to pick.

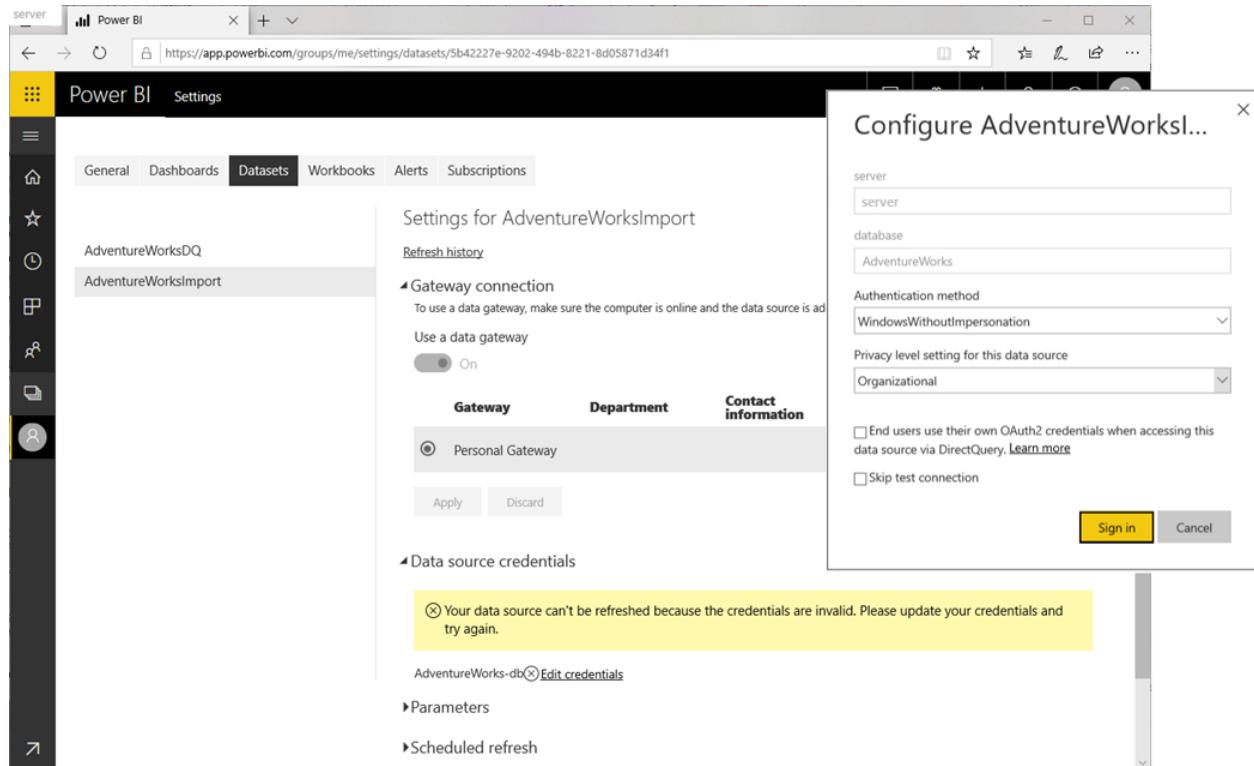
### ! Note

A dataset can only use a single gateway connection. In other words, it is not possible to access on-premises data sources across multiple gateway connections. Accordingly, you must add all required data source definitions to the same gateway.

## Deploying a personal data gateway

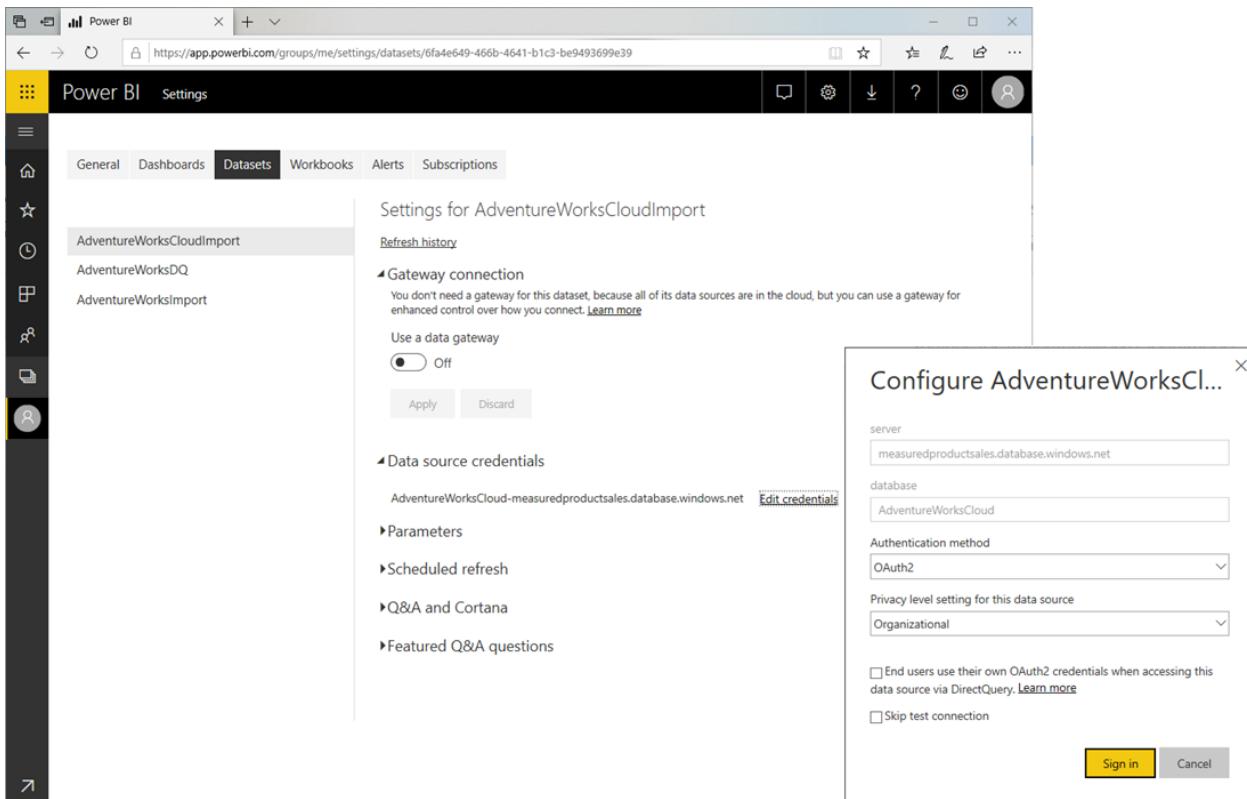
If you have no access to an enterprise data gateway and you're the only person who manages datasets so you don't need to share data sources with others, you can deploy a data gateway in personal mode. In the **Gateway connection** section, under **You have no personal gateways installed**, select **Install now**. The personal data gateway has several limitations as documented in [On-premises data gateway \(personal mode\)](#).

Unlike for an enterprise data gateway, you don't need to add data source definitions to a personal gateway. Instead, you manage the data source configuration by using the **Data source credentials** section in the dataset settings, as the following screenshot illustrates.



## Accessing cloud data sources

Datasets that use cloud data sources, such as Azure SQL DB, don't require a data gateway if Power BI can establish a direct network connection to the source. Accordingly, you can manage the configuration of these data sources by using the **Data source credentials** section in the dataset settings. As the following screenshot shows, you don't need to configure a gateway connection.

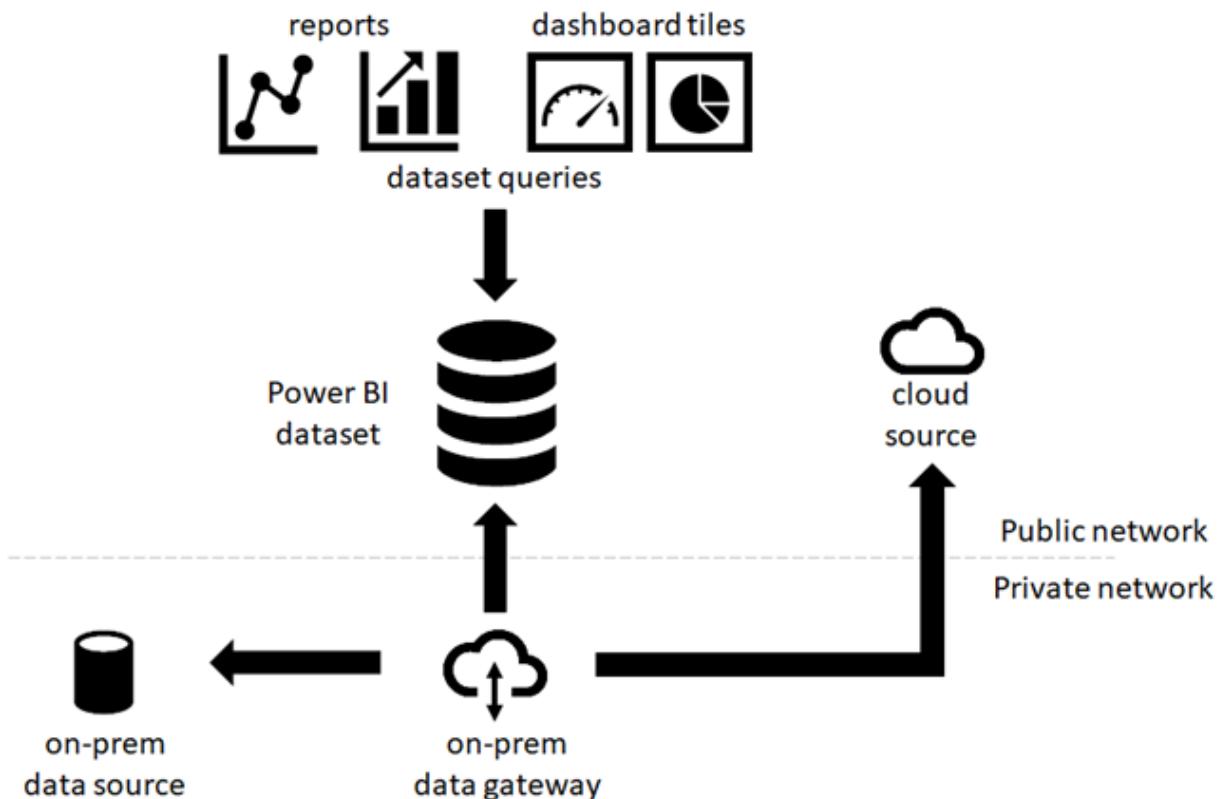


### ⓘ Note

Each user can only have one set of credentials per data source, across all of the datasets they own, regardless of the workspaces where the datasets reside. And each dataset can only have one owner. If you want to update the credentials for a dataset where you are not the dataset owner, you must first take over the dataset by clicking on the Take Over button on the dataset settings page.

## Accessing on-premises and cloud sources in the same source query

A dataset can get data from multiple sources, and these sources can reside on-premises or in the cloud. However, a dataset can only use a single gateway connection, as mentioned earlier. While cloud data sources don't necessarily require a gateway, a gateway is required if a dataset connects to both on-premises and cloud sources in a single mashup query. In this scenario, Power BI must use a gateway for the cloud data sources as well. The following diagram illustrates how such a dataset accesses its data sources.



### ① Note

If a dataset uses separate mashup queries to connect to on-premises and cloud sources, Power BI uses a gateway connection to reach the on-premises sources and a direct network connection to the cloud sources. If a mashup query merges or appends data from on-premises and cloud sources, Power BI switches to the gateway connection even for the cloud sources.

Power BI datasets rely on Power Query to access and retrieve source data. The following mashup listing shows a basic example of a query that merges data from an on-premises source and a cloud source.

Let

```

OnPremSource = Sql.Database("on-premises-db", "AdventureWorks"),

CloudSource = Sql.Databases("cloudsql.database.windows.net",
"AdventureWorks"),

TableData1 = OnPremSource{[Schema="Sales",Item="Customer"]}[Data],

TableData2 = CloudSource {[Schema="Sales",Item="Customer"]}[Data],

MergedData = Table.NestedJoin(TableData1, {"BusinessEntityID"},
TableData2, {"BusinessEntityID"}, "MergedData", JoinKind.Inner)

```

in

## MergedData

There are two options to configure a data gateway to support merging or appending data from on-premises and cloud sources:

- Add a data source definition for the cloud source to the data gateway in addition to the on-premises data sources.
- Enable the checkbox **Allow user's cloud data sources to refresh through this gateway cluster.**

The screenshot shows the 'ADD DATA SOURCE' interface. On the left, under 'GATEWAY CLUSTERS', 'Enterprise Data Gateway' is selected. On the right, the 'Gateway Cluster Settings' tab is active, showing a green checkmark for 'Online: You are good to go.' Below it is a section titled '① Add data sources to use the gateway'. This section contains fields for 'Gateway Cluster Name' (set to 'Enterprise Data Gateway'), 'Department', 'Description', and 'Contact Information'. At the bottom of this section is a checkbox group with three options:

- Allow user's cloud data sources to refresh through this gateway cluster. These cloud data sources do not need to be configured under this gateway cluster. [Learn more](#)
- Allow user's custom data connectors to refresh through this gateway cluster (preview). [Learn more](#)
- Distribute requests across all active gateways in this cluster. [Learn more](#)

A red rectangular box highlights the first checkbox. At the bottom of the page are 'Apply' and 'Discard' buttons.

If you enable the checkbox **Allow user's cloud data sources to refresh through this gateway cluster in the gateway configuration**, as in the screenshot above, Power BI can use the configuration that the user defined for the cloud source under **Data source credentials** in the dataset settings. This can help to lower the gateway configuration overhead. On the other hand, if you want to have greater control over the connections that your gateway establishes, you shouldn't enable this checkbox. In this case, you must add an explicit data source definition for every cloud source that you want to support to your gateway. It's also possible to enable the checkbox and add explicit data source definitions for your cloud sources to a gateway. In this case, the gateway uses the data source definitions for all matching sources.

## Configuring query parameters

The mashup or M queries you create by using Power Query can vary in complexity from trivial steps to parameterized constructs. The following listing shows a small sample mashup query that uses two parameters called *SchemaName* and *TableName* to access a given table in an AdventureWorks database.

```
let
 Source = Sql.Database("SqlServer01", "AdventureWorks"),
 TableData = Source{[Schema=SchemaName,Item=TableName]}[Data]
in
 TableData
```

 **Note**

Query parameters are only supported for Import mode datasets.  
DirectQuery/LiveConnect mode does not support query parameter definitions.

To ensure that a parameterized dataset accesses the correct data, you must configure the mashup query parameters in the dataset settings. You can also update the parameters programmatically by using the [Power BI REST API](#). The following screenshot shows the user interface to configure the query parameters for a dataset that uses the above mashup query.

The screenshot shows the 'Datasets' tab selected in the top navigation bar. On the left, a list of datasets includes 'AdventureWorksCloudImport', 'AdventureWorksDQ', 'AdventureWorksImport', 'AdventureWorksOnPremAndCloudImport', and 'AdventureWorksParameterizedImport', which is currently selected and highlighted in grey. The main pane displays 'Settings for AdventureWorksParameterizedImport'. It includes a 'Refresh history' link, a 'Gateway connection' section, a 'Data source credentials' section, and a 'Parameters' section. Under 'Parameters', there are two fields: 'SchemaName' containing 'Sales' and 'TableName' containing 'vIndividualCustomer'. At the bottom are 'Apply' and 'Discard' buttons. A 'Scheduled refresh' section is also present.

## Refresh and dynamic data sources

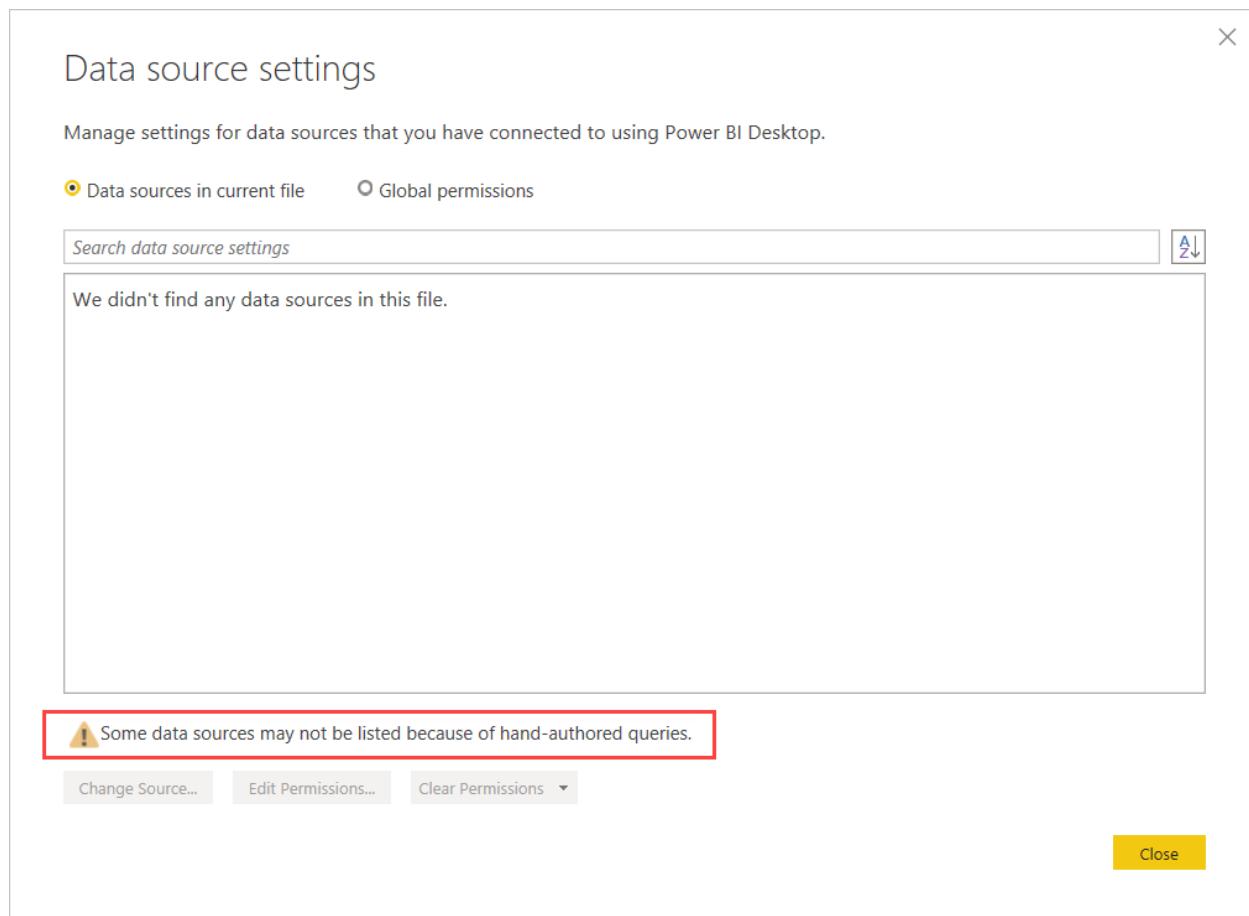
A *dynamic data source* is a data source in which some or all of the information required to connect can't be determined until Power Query runs its query, because the data is generated in code or returned from another data source. Examples include: the instance name and database of a SQL Server database; the path of a CSV file; or the URL of a web service.

In most cases, Power BI datasets that use dynamic data sources can't be refreshed in the Power BI service. There are a few exceptions in which dynamic data sources can be refreshed in the Power BI service, such as when using the `RelativePath` and `Query` options with the `Web.Contents` M function. Queries that reference Power Query parameters can also be refreshed.

To determine whether your dynamic data source can be refreshed, open the **Data Source Settings** dialog in **Power Query Editor**, and then select **Data Sources In Current File**. In the window that appears, look for the following warning message, as shown in the following image:

**Note**

Some data sources may not be listed because of hand-authored queries.



If that warning is present in the **Data Source Settings** dialog that appears, then a dynamic data source that can't be refreshed in the Power BI service is present.

## Configure scheduled refresh

Establishing connectivity between Power BI and your data sources is by far the most challenging task in configuring a data refresh. The remaining steps are relatively straightforward and include setting the refresh schedule and enabling refresh failure notifications. For step-by-step instructions, see the how-to guide [Configuring scheduled refresh](#).

## Setting a refresh schedule

The **Scheduled refresh** section is where you define the frequency and time slots to refresh a dataset. As mentioned earlier, you can configure up to eight daily time slots if your dataset is on shared capacity, or 48 time slots on Power BI Premium. The following screenshot shows a refresh schedule on a twelve-hour interval.

AdventureWorksCloudImport

AdventureWorksDQ

AdventureWorksImport

AdventureWorksOnPremAndCloudImport

AdventureWorksParameterizedImport

### Settings for AdventureWorksImport

Next refresh: Tue May 21 2019 13:30:00 GMT-0700 (Pacific Daylight Time)  
[Refresh history](#)

▶ Gateway connection

▶ Data source credentials

▶ Parameters

◀ Scheduled refresh

Keep your data up to date  
 On

Refresh frequency  
Daily

Time zone  
(UTC-08:00) Pacific Time (US and Canada)

Time  
1 ▾ 30 ▾ AM ▾ X  
1 ▾ 30 ▾ PM ▾ X

[Add another time](#)

Send refresh failure notification emails to me

**Apply** **Discard**

Having configured a refresh schedule, the dataset settings page informs you about the next refresh time, as in the screenshot above. If you want to refresh the data sooner, such as to test your gateway and data source configuration, perform an on-demand refresh by using the **Refresh Now** option in the dataset menu in the nav pane. On-demand refreshes don't affect the next scheduled refresh time.

#### Tip

Power BI does not have a monthly refresh interval option. However, you can use Power Automate to create a custom refresh interval that occurs monthly, as described in the following [Power BI blog post](#).

Note also that the configured refresh time might not be the exact time when Power BI starts the next scheduled process. Power BI starts scheduled refreshes on a best effort basis. The target is to initiate the refresh within 15 minutes of the scheduled time slot,

but a delay of up to one hour can occur if the service can't allocate the required resources sooner.

 **Note**

Power BI deactivates your refresh schedule after four consecutive failures or when the service detects an unrecoverable error that requires a configuration update, such as invalid or expired credentials. It is not possible to change the consecutive failures threshold.

## Getting refresh failure notifications

By default, Power BI sends refresh failure notifications to the dataset owner through email, so that they can act in a timely manner should refresh issues occur. If the owner has the Power BI app on their mobile device, they will also get the failure notification there. Power BI also sends an email notification when the service disables a scheduled refresh due to consecutive failures. Microsoft recommends that you leave the checkbox **Send refresh failure notification emails dataset owner** enabled.

It's also a good idea to specify additional recipients for scheduled refresh failure notifications by using the **Email these contacts when the refresh fails** textbox. Specified recipients receive refresh failure notifications via email and push notifications to the mobile app, just like the dataset owner does. Specified recipients might include a colleague taking care of your datasets while you are on vacation, or the email alias of your support team taking care of refresh issues for your department or organization. Sending refresh failure notifications to others in addition to the dataset owner helps ensure that issues get noticed and addressed in a timely manner.

 **Note**

Push notifications to the mobile apps do not support group aliases.

Note that Power BI not only sends notifications on refresh failures but also when the service pauses a scheduled refresh due to inactivity. After two months, when no user has visited any dashboard or report built on the dataset, Power BI considers the dataset inactive. In this situation, Power BI sends an email message to the dataset owner indicating that the service paused the refresh schedule for the dataset. See the following screenshot for an example of such a notification.

The screenshot shows a Microsoft Outlook window with the following details:

- File**, **Message** (selected), **Help**
- Tell me what you want to do** search bar
- Icons:** Delete, Archive, Reply, Reply All, Forward, Protect, Quick Steps, Move, Tags, Editing, Speech, Zoom, Report Message, Protection.
- Date:** Fri 2/1/2019 4:35 PM
- From:** Microsoft Power BI (No Reply)
- Subject:** Your scheduled refresh has been paused.
- To:** Email recipient
- Message Content:**
  - Microsoft** logo
  - Power BI** logo
  - Your scheduled refresh has been paused.**
  - We've paused scheduled refresh for **AdventureWorksDQ** in My Workspace due to inactivity.
  - No one has viewed dashboards or reports built using this dataset for two months. To resume it, just visit a report or dashboard built using this dataset or manually refresh the dataset using the Refresh Now option.
  - Learn more >** button
  - Activity Id:** fcb45050-2c2f-4299-90af-7d3535b5230d
  - Request Id:** e38f134c-8d50-45d1-8e12-06c3a3bd7adb
  - Cluster Name:** DUS-redirect.analysis.windows.net
- Privacy Statement** link
- Microsoft** logo

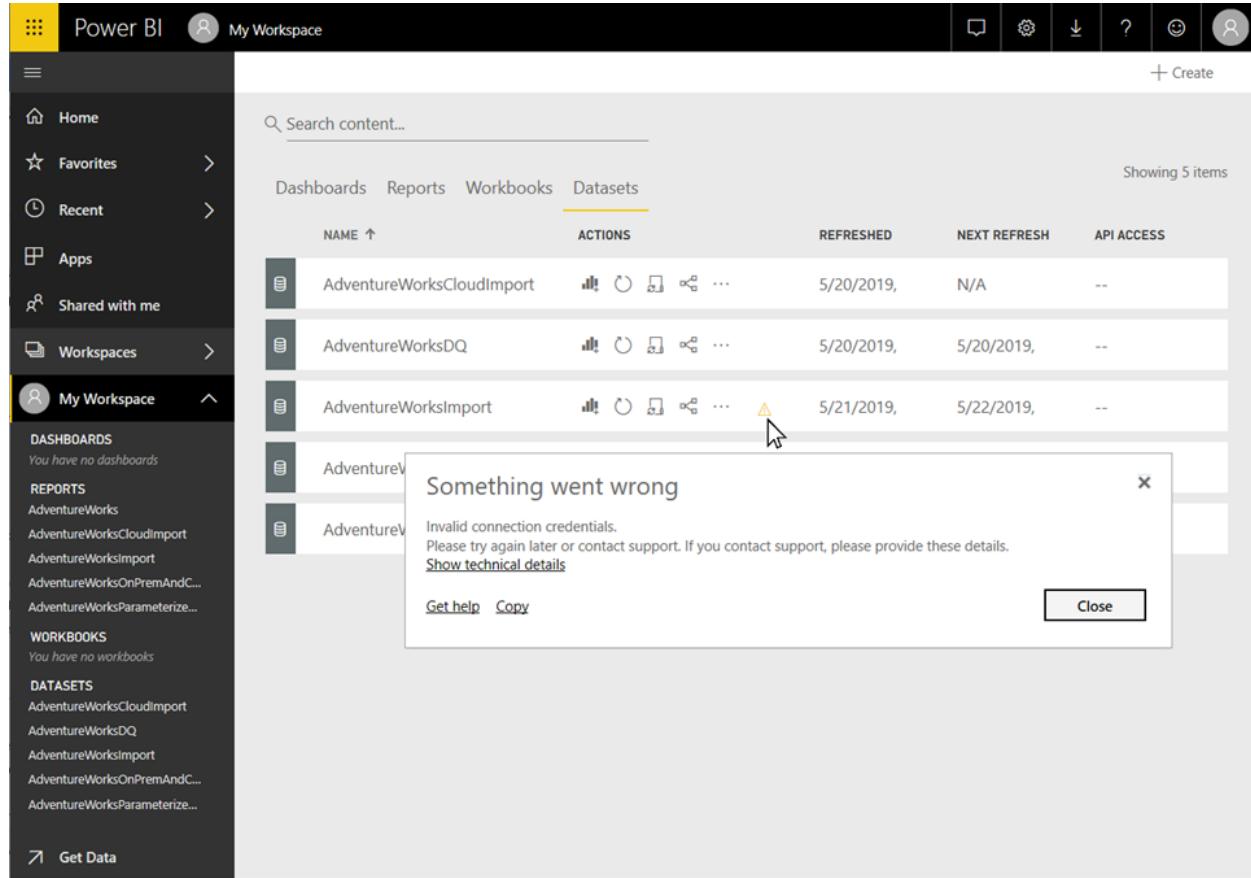
To resume scheduled refresh, visit a report or dashboard built using this dataset or manually refresh the dataset using the **Refresh Now** option.

### **Note**

Sending refresh notifications to external users is not supported. The recipients you specify in the **Email these users when the refresh fails** textbox must have accounts in your Azure Active Directory tenant. This limitation applies to both dataset refresh and dataflow refresh.

## Checking refresh status and history

In addition to failure notifications, it's a good idea to check your datasets periodically for refresh errors. A quick way is to view the list of datasets in a workspace. Datasets with errors show a small warning icon. Select the warning icon to obtain additional information, as in the following screenshot. For more information about troubleshooting specific refresh errors, see [Troubleshooting refresh scenarios](#).



The screenshot shows the Power BI 'My Workspace' interface. On the left is a navigation sidebar with sections for Home, Favorites, Recent, Apps, Shared with me, Workspaces, and My Workspace (which is currently selected). The main area displays a search bar and tabs for Dashboards, Reports, Workbooks, and Datasets. Under the Datasets tab, a table lists five items. The third item, 'AdventureWorksImport', has a yellow warning icon next to its name. A tooltip box is overlaid on this icon with the message: 'Something went wrong' followed by 'Invalid connection credentials. Please try again later or contact support. If you contact support, please provide these details.' Below the table are links for 'Get help' and 'Copy', and a 'Close' button.

NAME ↑	ACTIONS	REFRESHED	NEXT REFRESH	API ACCESS
AdventureWorksCloudImport	refresh icon	5/20/2019,	N/A	--
AdventureWorksDQ	refresh icon	5/20/2019,	5/20/2019,	--
AdventureWorksImport	refresh icon (yellow)	5/21/2019,	5/22/2019,	--
AdventureWorksOnPremAndCloud	refresh icon			
AdventureWorksParameterize	refresh icon			

The warning icon helps to indicate current dataset issues, but it's also a good idea to check the refresh history occasionally. As the name implies, the refresh history enables you to review the success or failure status of past synchronization cycles. For example, a gateway administrator might have updated an expired set of database credentials. As you can see in the following screenshot, the refresh history shows when an affected refresh started working again.

The screenshot shows the Power BI Settings interface. On the left, there's a navigation sidebar with options like Home, Favorites, Recent, Apps, Shared with me, Workspaces, and My Workspace. Under My Workspace, there are sections for Dashboards, Reports, Workbooks, Datasets, and Get Data. The main area is titled 'Datasets' and shows three datasets: AdventureWorksCloudImport, AdventureWorksDQ, and AdventureWorksImport. The AdventureWorksImport dataset is selected. A sub-section titled 'Settings for AdventureWorksImport' displays a message: 'Refresh in progress...' and 'Next refresh: Wed May 22 2019 01:30:00 GMT-0700 (Pacific Daylight Time)'. Below this is a link 'Refresh history'. A modal window titled 'Refresh history' is open, showing a table of refresh attempts:

Details	Type	Start	End	Status	Message
	On demand	5/21/2019, 1:52:54 PM	5/21/2019, 1:53:14	M	Completed
Show	On demand	5/21/2019, 1:51:07 PM	5/21/2019, 1:51:28	M	Failed Invalid connection credentials.
Show	Scheduled	5/21/2019, 1:30:08 PM	5/21/2019, 1:32:32	M	Failed Invalid connection credentials.
Show	On demand	5/21/2019, 1:20:04 PM	5/21/2019, 1:20:27	M	Failed Invalid connection credentials.

A red box highlights the last three failed attempts. A yellow 'Close' button is at the bottom right of the modal.

### ⓘ Note

You can find a link to display the refresh history in the dataset settings. You can also retrieve the refresh history programmatically by using the [Power BI REST API](#). By using a custom solution, you can monitor the refresh history of multiple datasets in a centralized way.

## Automatic page refresh

Automatic page refresh works at a report page level, and allows report authors to set a refresh interval for visuals in a page that is only active when the page is being consumed. Automatic page refresh is only available for DirectQuery data sources. The minimum refresh interval depends on which type of workspace the report is published in, and the capacity admin settings for Premium workspaces and [embedded workspaces](#).

Learn more about automatic page refresh in the [automatic page refresh](#) article.

## Refresh cancellation

Stopping a dataset refresh is useful when you want to stop a refresh of a large dataset during peak time. Use the refresh cancellation feature to stop refreshing datasets that reside on [Premium](#), [Premium Per User \(PPU\)](#) or [Power BI Embedded](#) capacities.

To cancel a dataset refresh, you need to be a contributor, member or an admin of the dataset's workspace. Dataset refresh cancellation only works with datasets that use [import mode](#) or [composite mode](#).

 **Note**

Datasets created as part of datamarts aren't supported.

To start a refresh go to the dataset you want to refresh, and select **Refresh now**.

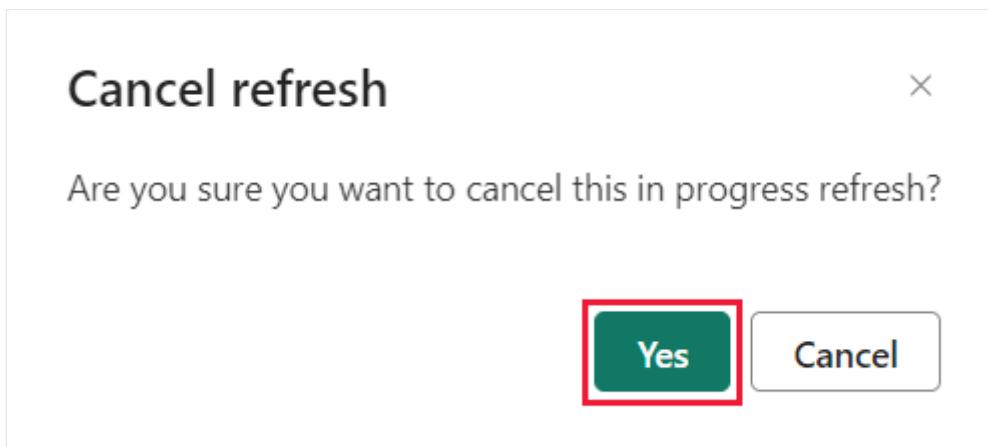
	Name	Type
	Profitability	Dashboard
	Profitability	Report
	Profitability	Workbook
	Profitability	  ... Dataset

To stop a refresh follow these steps:

1. Go to the dataset that's refreshing and select **Cancel refresh**.

	Name	Type
⌚	Profitability	Dashboard
⌚⌚	Profitability	Report
📅	Profitability	Workbook
🕒	Profitability	Dataset

2. In the *Cancel refresh* pop-up window, select **Yes**.



## Best practices

Checking the refresh history of your datasets regularly is one of the most important best practices you can adopt to ensure that your reports and dashboards use current data. If you discover issues, address them promptly and follow up with data source owners and gateway administrators if necessary.

In addition, consider the following recommendations to establish and maintain reliable data refresh processes for your datasets:

- Schedule your refreshes for less busy times, especially if your datasets are on Power BI Premium. If you distribute the refresh cycles for your datasets across a broader time window, you can help to avoid peaks that might otherwise overtax available resources. Delays starting a refresh cycle are an indicator of resource overload. If a Premium capacity is exhausted, Power BI might even skip a refresh cycle.

- Keep refresh limits in mind. If the source data changes frequently or the data volume is substantial, consider using DirectQuery/LiveConnect mode instead of Import mode if the increased load at the source and the impact on query performance are acceptable. Avoid constantly refreshing an Import mode dataset. However, DirectQuery/LiveConnect mode has several limitations, such as a one-million-row limit for returning data and a 225-seconds response time limit for running queries, as documented in [Use DirectQuery in Power BI Desktop](#). These limitations might require you to use Import mode nonetheless. For large data volumes, consider the use of [aggregations in Power BI](#).
- Verify that your dataset refresh time doesn't exceed the maximum refresh duration. Use Power BI Desktop to check the refresh duration. If it takes more than 2 hours, consider moving your dataset to Power BI Premium. Your dataset might not be refreshable on shared capacity. Also consider using [Incremental refresh](#) for datasets that are larger than 1 GB or take several hours to refresh.
- Optimize your datasets to include only those tables and columns that your reports and dashboards use. Optimize your mashup queries and, if possible, avoid dynamic data source definitions and expensive DAX calculations. Specifically avoid DAX functions that test every row in a table because of the high memory consumption and processing overhead.
- Apply the same privacy settings as in Power BI Desktop to ensure that Power BI can generate efficient source queries. Keep in mind that Power BI Desktop does not publish privacy settings. You must manually reapply the settings in the data source definitions after publishing your dataset.
- Limit the number of visuals on your dashboards, especially if you use [row-level security \(RLS\)](#). As explained earlier in this article, an excessive number of dashboard tiles can significantly increase the refresh duration.
- Use a reliable enterprise data gateway deployment to connect your datasets to on-premises data sources. If you notice gateway-related refresh failures, such as gateway unavailable or overloaded, follow up with gateway administrators to either add additional gateways to an existing cluster or deploy a new cluster (scale up versus scale out).
- Use separate data gateways for Import datasets and DirectQuery/LiveConnect datasets so that the data imports during scheduled refresh don't impact the performance of reports and dashboards on top of DirectQuery/LiveConnect datasets, which query the data sources with each user interaction.
- Ensure that Power BI can send refresh failure notifications to your mailbox. Spam filters might block the email messages or move them into a separate folder where you might not notice them immediately.

## Next steps

[Configuring scheduled refresh](#)

[Tools for troubleshooting refresh issues](#)

[Troubleshooting refresh scenarios](#)

More questions? [Try asking the Power BI Community](#) ↗

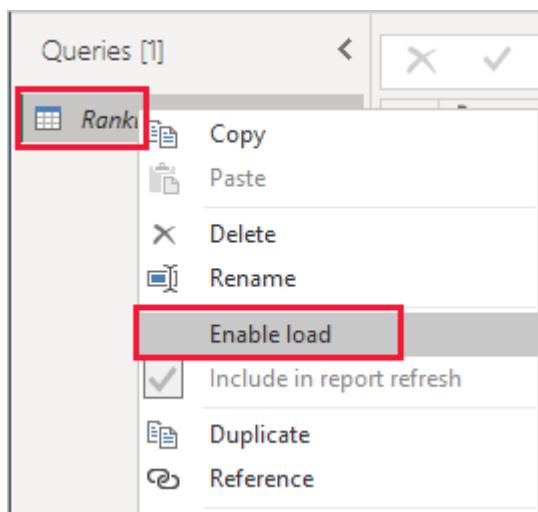
# Managing query refresh in Power BI

Article • 02/02/2023

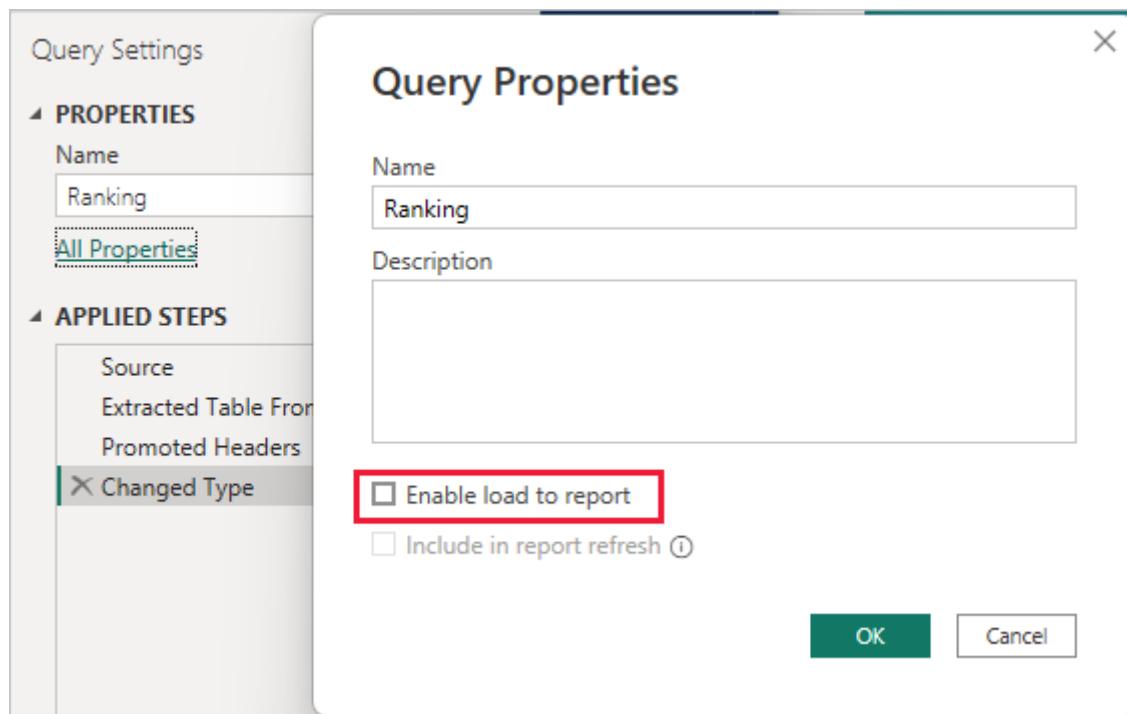
With Power BI, you can connect to [many different types of data sources](#) and shape the [data to meet your needs](#). The connections and transformations are stored in queries, which are by default refreshed either by manual or automatic refresh of the report in the Service.

## Managing loading of queries

In many situations, it makes sense to break down your data transformations in multiple queries. One popular example is merging where you merge two queries into one to essentially do a join. In this situation, some queries aren't relevant to load into Power BI Desktop because they're intermediate steps, while they're still required for your data transformations to work correctly. For these queries, you can make sure they aren't loaded in Power BI Desktop. Unselect **Enable load** in the context menu of the query in Power Query Editor:

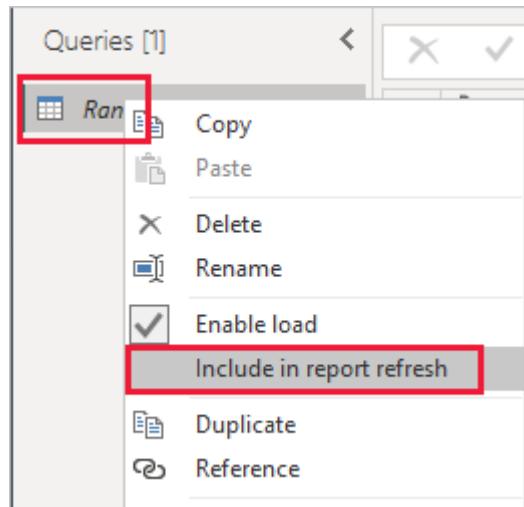


You can also make this change in the **Properties** screen:

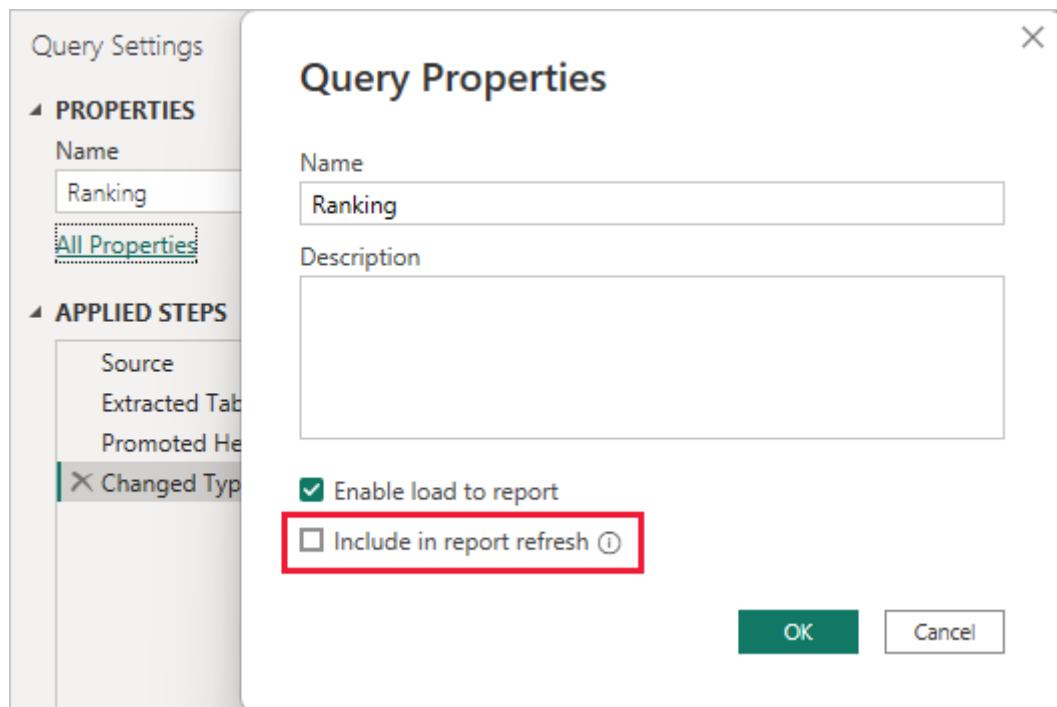


## Excluding queries from refresh

For queries for which the source data isn't updated often or at all, it makes sense to not have the queries included in the refresh of the report. In this scenario, you can exclude queries from being refreshed when the report is refreshed. Unselect **Include in report refresh** in the context menu of the query in Power Query Editor:



You can also make this change in the **Properties** screen:



### i Note

Any queries excluded from refresh are also excluded in automatic refresh in the Power BI service.

## Next steps

- Shape and combine data
- Configuring scheduled refresh
- Tools for troubleshooting refresh issues
- Troubleshooting refresh scenarios

More questions? [Try asking the Power BI Community](#) ↗

# Configure scheduled refresh

Article • 12/28/2022

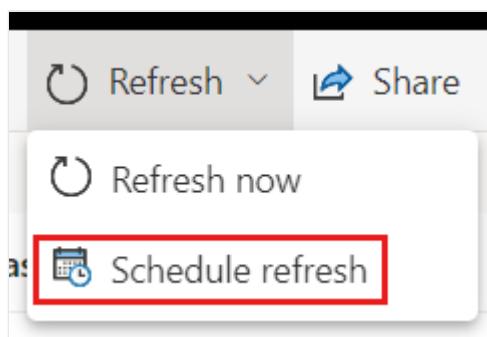
## ⓘ Note

After two months of inactivity, scheduled refresh on your dataset is paused. For more information, see *Scheduled refresh* later in this article.

This article describes the options available for scheduled refresh for the [On-premises data gateway \(personal mode\)](#) and the [On-premises data gateway](#). You specify refresh options in the following areas of the Power BI service: **Gateway connection**, **Data source credentials**, and **Schedule refresh**. We'll look at each in turn. For more information about data refresh, including limitations on refresh schedules, see [Data refresh](#).

To get to the **Schedule refresh** screen:

1. In the navigation pane, under **Datasets**, select a dataset.
2. Select **Refresh > Schedule refresh**.



## Gateway connection

You'll see different options here depending on whether you have a personal gateway or enterprise gateway online and available.

If no gateway is available, you'll see **Gateway connection** disabled. You'll also see a message indicating how to install the personal gateway.

If you have a personal gateway configured and it's online, it's available to select. It shows offline if it's not available.

#### ⚠ Gateway connection

You don't need a gateway for this dataset, because all of its data sources are in the cloud, but you can use a gateway for enhanced control over how you connect. [Learn more](#)

Use an On-premises or VNet data gateway



You can also select the enterprise gateway if one is available for you. You only see an enterprise gateway available if your account is listed in the **Users** tab of the data source configured for a given gateway.

## Data source credentials

### Power BI Gateway - Personal

If you're using the personal gateway to refresh data, you must supply the credentials to connect to the back-end data source. If you connected to an app from an online service, the credentials you entered to connect are carried over for scheduled refresh.

#### ⚠ Data source credentials

ContosoRetailDW-10.0.0.10    [Edit credentials](#)

You're only required to sign in to a data source the first time you use refresh on that dataset. Once entered, those credentials are retained with the dataset.

#### ⓘ Note

For some authentication methods, if the password you use to sign into a data source expires or is changed, you need to change it for the data source in **Data source credentials** too.

If there's a problem, typically it's either the gateway is offline because it couldn't sign in to Windows and start the service, or Power BI couldn't sign in to the data sources to query for updated data. If refresh fails, check the dataset settings. If the gateway service is offline, **Status** is where you see the error. If Power BI can't sign into the data sources, you see an error in Data Source Credentials.

## On-premises data gateway

If you're using the on-premises data gateway to refresh data, you don't need to supply credentials, as they're defined for the data source by the gateway administrator.

- ▶ Data source credentials
- ▶ Sensitivity label
- ▶ Parameters
- ▶ Scheduled refresh

① Note

When connecting to on-premises SharePoint for data refresh, Power BI supports only *Anonymous*, *Basic*, and *Windows (NTLM/Kerberos)* authentication mechanisms. Power BI does not support *ADFS* or any *Forms-Based Authentication* mechanisms for data refresh of on-premises SharePoint data sources.

## Scheduled refresh

The **Scheduled refresh** section is where you define the frequency and time slots to refresh the dataset. Some data sources don't require a gateway to be configurable for refresh, while other data sources require a gateway.

In a Direct Query scenario, when a dataset qualifies for performance optimization, **Refresh schedule** will be moved to the **Optimize performance** section.

Set the **Keep your data up to date** slider to **On** to configure the settings.

① Note

The target is to initiate the refresh within 15 minutes of the scheduled time slot, but a delay of up to one hour can occur if the service can't allocate the required resources sooner. Refresh can begin as early as five minutes before the scheduled refresh time.

## Scheduled refresh

### Keep your data up to date

Configure a data refresh schedule to import data from the data source into the dataset. [Learn more](#)



On

### Refresh frequency

▼

### Time zone

▼

### Time

[Add another time](#)

### Send refresh failure notifications to

Dataset owner

These contacts:

### Note

After two months of inactivity, scheduled refresh on your dataset is paused. A dataset is considered inactive when no user has visited any dashboard or report built on the dataset. When scheduled refresh is paused, the dataset owner is sent an email. The refresh schedule for the dataset is then displayed as **disabled**. To resume scheduled refresh, revisit any dashboard or report built on the dataset.

## What's supported?

### Note

Scheduled refresh will also get disabled automatically after four consecutive errors.

### Tip

Power BI does not have a monthly refresh interval option. However, you can use Power Automate to create a custom refresh interval that occurs monthly, as

described in the following [Power BI blog post](#).

Certain datasets are supported against different gateways for scheduled refresh.

## Power BI Gateway - Personal

### Power BI Desktop

- All online data sources shown in Power BI Desktop's **Get data** and Power Query Editor.
- All on-premises data sources shown in Power BI Desktop's **Get data** and Power Query Editor except for Hadoop file (HDFS) and Microsoft Exchange.

### Excel

- All online data sources shown in Power Query.
- All on-premises data sources shown in Power Query except for Hadoop file (HDFS) and Microsoft Exchange.
- All online data sources shown in Power Pivot.
- All on-premises data sources shown in Power Pivot except for Hadoop file (HDFS) and Microsoft Exchange.

#### ⓘ Note

In Excel 2016 and later, **Launch Power Query Editor** is available from **Get Data** in the **Data** ribbon.

## Power BI Gateway

For information about supported data sources, see [Power BI data sources](#).

## Troubleshooting

Sometimes refreshing data may not go as expected, typically due to an issue connected with a gateway. See these gateway troubleshooting articles for tools and known issues.

- [Troubleshoot the On-premises data gateway](#)
- [Troubleshoot the Power BI Gateway - Personal](#)

## Next steps

- Data refresh in Power BI
- Power BI Gateway - Personal
- On-premises data gateway (personal mode)
- Troubleshoot the On-premises data gateway
- Troubleshoot the Power BI Gateway - Personal

More questions? [Try asking the Power BI Community ↗](#)

# Refresh summaries for Power BI

Article • 04/17/2023

The Power BI **Refresh summary** page, found in the Power BI Admin portal, provides control and insight into your refresh schedules, capacities, and potential refresh schedule overlaps for your Power BI Premium capacities. You can use the refresh summary page to determine whether you should adjust refresh schedules, learn error codes associated with refresh issues, and properly manage your data refresh scheduling.

The refresh summaries page has two views:

- **History.** Displays the refresh summary history for Power BI Premium capacities for which you're an administrator.
- **Schedule.** Shows the schedule view for scheduled refresh, which also can uncover issues with time slots that are oversubscribed.

You can also export information about a refresh event to a .csv file, which can provide significant information and insight into refresh events or errors that can be impacting the performance or completion of scheduled refresh events.

The following sections look at each of these views in turn.

## Refresh history

You can select the **History** view by clicking on **History** in the refresh summaries page.

The screenshot shows the Power BI Admin portal interface. In the top navigation bar, 'Microsoft' and 'Power BI Admin portal' are visible. On the left, there's a sidebar with links like Home, Favorites, Recent, Apps, Shared with me, Deployment pipelines, Learn, Workspaces, and My workspace. The main content area is titled 'Admin portal' and contains a 'Capacity settings' section with a 'Refresh summary' button. Below this is the 'Refresh summary' section, which has tabs for 'Schedule' and 'History'. The 'History' tab is selected and highlighted with a red box. Underneath are buttons for 'Refresh' and 'Export'. A table lists refresh history items with columns for Name, Type, Workspace, and Capacity. The first item is 'legacySQLDQv3' (Dataset, AS Azure, msitAPCap). The second is 'Report Usage Metrics Model' (Dataset, DVE PBI Ribar). The third is 'SSASLiveConnOnPrem' (Dataset, ALM-BetaTestPhase2). A circular icon with a magnifying glass and a plus sign is in the bottom right corner of the table area.

Name	Type	Workspace	Capacity
legacySQLDQv3	Dataset	AS Azure	msitAPCap
Report Usage Metrics Model	Dataset	DVE PBI Ribar	
SSASLiveConnOnPrem	Dataset	ALM-BetaTestPhase2	

The **History** provides an overview of the outcomes of recently scheduled refreshes on the capacities for which you have admin privilege. You can sort the view by any column

by selecting the column. You can choose to sort the view by the column selected by ascending order, descending, or by using text filters.

The screenshot shows a table of refresh history with columns: Capacity, Start time, Duration, Average duration, and Refreshes per day. A red box highlights the sorting and filtering dropdown menu. The menu includes 'Sort ascending' (with an arrow pointing up), 'Sort descending' (with an arrow pointing down), 'Text filters' (with a magnifying glass icon), and 'Clear all filters' (with a cross icon).

In history view, the data associated with a given refresh is based on up to 60 most recent records for each scheduled refresh.

You can also export information for any scheduled refresh to a .csv file, which includes detailed information including error messages for each refresh event. Exporting to a .csv file lets you sort the file based on any of the columns, search for words, sort based on error codes or owners, and so on. The following image shows an example exported .csv file.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Type	Workspace	Capacity	St	Duration	Average duration	Refreshes per day	Outcome	Owner	Error Details				
2	Dataset AS Azure	Power BI Te11/2	0:00:01					Failed		{"error":{"code":"ADGraphGroupOperationFailedGroupOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the group."}}				
3	Dataset AS Azure	Power BI Te11/2	0:00:01					Failed		{"error":{"code":"ADGraphGroupOperationFailedGroupOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the group."}}				
4	Dataset BigDataKusto	Power BI Te10/2	0:00:03					Failed		{"error":{"code":"DM_GWPipeline_Gateway_InvalidConnectionString","message": "The connection string is invalid or missing required parameters."}}				
5	Dataset FTC EMEA CRS	Power BI Te7/4/	0:41:18					Failed		{"error":{"code":"DM_GWPipeline_Gateway_MashupDatasetOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the dataset."}}				
6	Dataset DEV CTQ	Power BI Te9/27	0:28:13					Failed		{"error":{"code":"DM_GWPipeline_Gateway_MashupDatasetOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the dataset."}}				
7	Dataset alm-xmla-test	Power BI Te3/2/	0:00:22					Failed		{"error":{"code":"DM_GWPipeline_Gateway_MashupDatasetOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the dataset."}}				
8	Dataset DEV CTQ	Power BI Te7/30	0:24:31					Failed		{"error":{"code":"DM_GWPipeline_Gateway_MashupDatasetOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the dataset."}}				
9	Dataset RAYJACTestBe	Power BI Te9/4/	0:00:08					Failed		{"error":{"code":"DM_GWPipeline_Gateway_MashupDatasetOwnerNotInGroupOrADGraphError","message": "The user is not a member of the group or has no permission to access the dataset."}}				
10	Dataset ALM-BetaTest	Power BI Te4/10	0:00:00					Failed		{"error":{"code":"DMTS_MonikerWithUnboundDatasetOwnerNotInGroupOrADGraphError","message": "The moniker is bound to a dataset that does not exist or the user is not a member of the group or has no permission to access the dataset."}}				
11	Dataset ProjectVelocity	Power BI Te7/16	0:07:01					Failed		{"error":{"code":"DMTS_MonikerWithUnboundDatasetOwnerNotInGroupOrADGraphError","message": "The moniker is bound to a dataset that does not exist or the user is not a member of the group or has no permission to access the dataset."}}				
12	Dataset DidierS Test Gi	Power BI Te1/28	0:00:02					Failed		{"error":{"code":"DMTS_OAuthTokenRefreshFailedError","message": "The OAuth token refresh failed."}}				
13	Dataset AS Azure	Power BI Te4/10	0:00:56					Failed		{"error":{"code":"DMTS_OAuthTokenRefreshFailedError","message": "The OAuth token refresh failed."}}				
14	Dataset IPG UAT	Power BI Te9/25	0:00:00					Failed		{"error":{"code":"DMTS_UserNotFoundInADGraphError","message": "The user was not found in the Active Directory Graph."}}				
15	Dataset Compliance Cc	Power BI Te10/2	0:00:00					Failed		{"error":{"code":"DMTS_UserNotFoundInADGraphError","message": "The user was not found in the Active Directory Graph."}}				
16	Dataset Compliance Cc	Power BI Te4/10	0:00:00					Failed		{"error":{"code":"DMTS_UserNotFoundInADGraphError","message": "The user was not found in the Active Directory Graph."}}				
17	Dataset Compliance Cc	Power BI Te4/10	0:00:01					Failed		{"error":{"code":"DMTS_UserNotFoundInADGraphError","message": "The user was not found in the Active Directory Graph."}}				
18	Dataset AbbottRpt	PBI Premium 4/9/	0:00:03					Failed		{"error":{"code":"EnterpriseGateway_ShortMessage_Error","message": "An error occurred while processing the request."}}				
19	Dataset alm-xmla-test	Power BI Te3/13	0:00:01					Failed		{"error":{"code":"EnterpriseGateway_ShortMessage_Error","message": "An error occurred while processing the request."}}				
20	Dataset mvv_TestSite	Power BI Te11/1	0:00:02					Failed		{"error":{"code":"EnterpriseGateway_ShortMessage_Error","message": "An error occurred while processing the request."}}				
21	Dataset Retail Investig	Power BI Te2/23	0:00:45					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
22	Dataset AS Azure	Power BI Te2/12	1:47:22					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
23	Dataset AS Azure	Power BI Te2/21	0:51:15					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
24	Dataset AS Azure	Power BI Te2/6/	0:08:04					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
25	Dataset Microsoft New	Power BI Te4/23	0:30:02					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
26	Dataset PBI SaaS Devs	Power BI Te4/7/	0:00:04					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
27	Dataset P&G Performa	Power BI TeToda	0:00:03					1 Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				
28	Dataset AS Azure	Power BI Te3/12	0:00:04					Failed		{"error":{"code":"ModelRefresh_ShortMessage_Proces","message": "The model refresh process failed."}}				

With the information in the exported file, you can review the capacity, duration, and any error messages recorded for the instance of refresh.

# Refresh schedule

You can select the **Schedule** view by selecting **Schedule** in refresh summaries. The **Schedule** view displays scheduling information for the week, broken down into 30-minute time slots.

The screenshot shows the Admin portal interface. On the left, there's a sidebar with various settings like Tenant settings, Usage metrics, Users, Premium Per User, Audit logs, Domains (preview), Capacity settings, and Refresh summary. The Refresh summary item is highlighted with a red box. The main area is titled "Refresh summary" and has tabs for "Schedule" (which is also highlighted with a red box) and "History". Below the tabs, there's a dropdown for "Choose a capacity:" set to "Power BI Team - P2" and a toggle for "Use UTC timezone:". A "Refresh" button and an "Export" link are also present. The main content area shows a table of scheduled refreshes:

Day of the week	Time	Refresh time booked (minutes)	Refresh time available (minutes)
Sunday	12:00 AM		Unlimited
Sunday	12:30 AM		Unlimited
Sunday	01:00 AM		Unlimited
Sunday	01:30 AM		Unlimited
Sunday	02:00 AM		Unlimited

A magnifying glass icon with a plus sign is located in the bottom right corner of the table area.

The **Schedule** view is very useful in determining whether the refresh events scheduled are properly spaced, allowing for all refreshes to complete without overlap, or whether you have scheduled refresh events that are taking too long and creating resource contention. If you find such resource contention, you should adjust your refresh schedules to avoid the conflicts or overlap, so your scheduled refreshes can complete successfully.

The **Refresh time booked (minutes)** column is a calculation of the average of up to 60 records for each associated dataset. The numeric value for each 30-minute time slot is the sum of minutes calculated for all scheduled refreshes scheduled to start on the time slot *and* any scheduled refreshes set to start on the *previous* time slot, but whose average duration overflows into the time slot that's selected.

The **Refresh time available (minutes)** column is a calculation of the minutes available for refresh in each time slot, minus whatever refresh is already scheduled for that time slot. For example, if your P2 subscription provides 80 concurrently running refreshes, you have 80 30-minute slots, so  $80 \text{ refreshes} \times 30 \text{ minutes each} = 2,400 \text{ minutes}$  available for refresh in that time slot. If you have one refresh booked in that slot that takes 20

minutes, your **Refresh time available (minutes)** in that slot is 2,380 minutes (2,400 total minutes available, minus 20 minutes already booked = 2,380 minutes still available).

You can select a time slot and then select the associated **details** button to see which scheduled refresh events contribute to the refresh time booked, their owners, and how long they take to complete.

Let's look at an example, to see how this works. The following dialog is displayed when we select the 8:30 PM time slot for Sunday, and select **details**.

Refresh details for Sunday at 8:30 PM							
See a breakdown of refreshes occurring in your selected time slot and find the owner.							
Name	Type	Workspace	Owner	Scheduled time slot	Average duration	Refreshes per day	
1 Unilever Gov Review	Dataset	TGHCompliance	admin@contoso.com	08:30 PM	00:04:39	1	
2 Unilever Performance Overview	Dataset	Unilever Performance	admin@contoso.com	08:00 PM	00:48:04	6	
3 API Telemetry	Dataset	Compliance Central	admin@contoso.com	08:30 PM	00:00:06	60	

There are three scheduled refresh events occurring in this time slot.

Scheduled refresh #1 and #3 are both scheduled for this 8:30 PM time slot, which we can determine by looking at the value in the **Scheduled time slot** column. Their average durations are 4:39 and six seconds (0:06) respectively. All is good there.

However, scheduled refresh #2 is scheduled for the 8:00 PM time slot, but because it takes an average of over 48 minutes to complete (seen in the **Average duration** column), that refresh event overflows into the next 30-minute time slot.

That's not good. The administrator in this case should contact the owners of that scheduled refresh instance and suggest they find a different time slot for that scheduled refresh, or reschedule the other refreshes so there's no overlap, or find some other solution to prevent such overlap.

## Next steps

- [Data refresh in Power BI](#)
- [Power BI Gateway - Personal](#)
- [On-premises data gateway \(personal mode\)](#)
- [Troubleshooting the On-premises data gateway](#)

- Troubleshooting the Power BI Gateway - Personal

More questions? [Try asking the Power BI Community ↗](#)

# Enhanced refresh with the Power BI REST API

Article • 05/09/2023

You can use any programming language that supports REST calls to do dataset refresh operations by using the Power BI Refresh Dataset REST API.

Optimized refresh for large and complex partitioned datasets is traditionally invoked with programming methods that use TOM (Tabular Object Model), PowerShell cmdlets, or TMSL (Tabular Model Scripting Language). However, these methods require long-running HTTP connections that can be unreliable.

The Power BI Refresh Dataset REST API can carry out dataset refresh operations asynchronously, so long-running HTTP connections from client applications aren't necessary. Compared to standard refresh operations, *enhanced refresh* with the REST API provides more customization options and the following features that are helpful for large models:

- Batched commits
- Table and partition-level refresh
- Applying incremental refresh policies
- `GET` refresh details
- Refresh cancellation

## ⓘ Note

- Previously, enhanced refresh was called *asynchronous refresh with REST API*. However, a standard refresh that uses the Refresh Dataset REST API also runs asynchronously by its inherent nature.
- Enhanced Power BI REST API refresh operations don't automatically refresh tile caches. Tile caches refresh only when a user accesses a report.

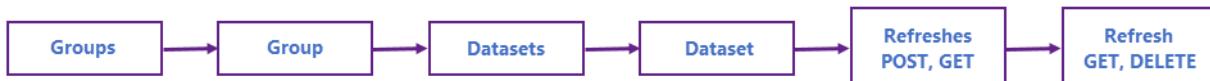
## Base URL

The base URL is in the following format:

HTTP

```
https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets/{datasetId}/refreshes
```

You can append resources and operations to the base URL based on parameters. In the following diagram, **Groups**, **Datasets**, and **Refreshes** are *collections*. **Group**, **Dataset**, and **Refresh** are *objects*.



## Requirements

You need the following requirements to use the REST API:

- A dataset in Power BI Premium, Premium per user, or Power BI Embedded.
- A group ID and dataset ID to use in the request URL.
- `Dataset.ReadWrite.All` permission scope.

The number of refreshes is limited per the general limitations for API-based refreshes for Pro and Premium datasets.

## Authentication

All calls must authenticate with a valid Azure Active Directory (Azure AD) OAuth 2 token in the Authorization header. The token must meet the following requirements:

- Be either a user token or an application service principal.
- Have the audience correctly set to <https://api.powerbi.com>.
- Be used by a user or application that has sufficient permissions on the dataset.

### ⓘ Note

REST API modifications don't change currently defined permissions for dataset refreshes.

## POST /refreshes

To do a refresh, use the POST verb on the /refreshes collection to add a new refresh object to the collection. The Location header in the response includes the `requestId`.

Because the operation is asynchronous, a client application can disconnect and use the `requestId` to check the status later if necessary.

The following code shows a sample request:

HTTP

```
POST https://api.powerbi.com/v1.0/myorg/groups/f089354e-8366-4e18-aea3-4cb4a3a50b48/datasets/cfafbeb1-8037-4d0c-896e-a46fb27ff229/refreshes
```

The request body might resemble the following example:

JSON

```
{
 "type": "Full",
 "commitMode": "transactional",
 "maxParallelism": 2,
 "retryCount": 2,
 "objects": [
 {
 "table": "DimCustomer",
 "partition": "DimCustomer"
 },
 {
 "table": "DimDate"
 }
]
}
```

### ⓘ Note

The service accepts only one refresh operation at a time for a dataset. If there's a current running refresh and another request is submitted, a `400 Bad Request` HTTP status code returns.

## Parameters

To do an enhanced refresh operation, you must specify one or more parameters in the request body. Specified parameters can specify the default or an optional value. When the request specifies parameters, all other parameters apply to the operation with their default values. If the request specifies no parameters, all parameters use their default values, and a standard refresh operation occurs.

Name	Type	Default	Description
<code>type</code>	Enum	<code>automatic</code>	The type of processing to perform. Types align with the TMSL refresh command types: <code>full</code> , <code>clearValues</code> , <code>calculate</code> , <code>dataOnly</code> , <code>automatic</code> , and <code>defragment</code> . The <code>add</code> type isn't supported.
<code>commitMode</code>	Enum	<code>transactional</code>	Determines whether to commit objects in batches or only when complete. Modes are <code>transactional</code> and <code>partialBatch</code> .
<code>maxParallelism</code>	Int	<code>10</code>	Determines the maximum number of threads that can run the processing commands in parallel. This value aligns with the <code>MaxParallelism</code> property that can be set in the TMSL <code>Sequence</code> command or by using other methods.
<code>retryCount</code>	Int	<code>0</code>	Number of times the operation retries before failing.
<code>objects</code>	Array	Entire dataset	An array of objects to process. Each object includes <code>table</code> when processing an entire table, or <code>table</code> and <code>partition</code> when processing a partition. If no objects are specified, the entire dataset refreshes.
<code>applyRefreshPolicy</code>	Boolean	<code>true</code>	If an incremental refresh policy is defined, determines whether to apply the policy. Modes are <code>true</code> or <code>false</code> . If the policy isn't applied, the full process leaves partition definitions unchanged, and fully refreshes all partitions in the table.  If <code>commitMode</code> is <code>transactional</code> , <code>applyRefreshPolicy</code> can be <code>true</code> or <code>false</code> . If <code>commitMode</code> is <code>partialBatch</code> , <code>applyRefreshPolicy</code> of <code>true</code> isn't supported, and <code>applyRefreshPolicy</code> must be set to <code>false</code> .
<code>effectiveDate</code>	Date	Current date	If an incremental refresh policy is applied, the <code>effectiveDate</code> parameter overrides the current date.

## Response

JSON

202 Accepted

The response also includes a `Location` response-header field to point the caller to the refresh operation that was created and accepted. The `Location` is the location of the new resource the request created, which includes the `requestId` that some enhanced refresh operations require. For example, in the following response, `requestId` is the last identifier in the response `87f31ef7-1e3a-4006-9b0b-191693e79e9e`.

JSON

```
x-ms-request-id: 87f31ef7-1e3a-4006-9b0b-191693e79e9e
Location: https://api.powerbi.com/v1.0/myorg/groups/f089354e-8366-4e18-aea3-4cb4a3a50b48/datasets/cfafbeb1-8037-4d0c-896e-a46fb27ff229/refreshes/87f31ef7-1e3a-4006-9b0b-191693e79e9e
```

## GET /refreshes

Use the GET verb on the `/refreshes` collection to list historical, current, and pending refresh operations.

The response body might look like the following example:

JSON

```
[
 {
 "requestId": "1344a272-7893-4afa-a4b3-3fb87222fdac",
 "refreshType": "ViaEnhancedApi",
 "startTime": "2020-12-07T02:06:57.1838734Z",
 "endTime": "2020-12-07T02:07:00.4929675Z",
 "status": "Completed",
 "extendedStatus": "Completed"
 },
 {
 "requestId": "474fc5a0-3d69-4c5d-adb4-8a846fa5580b",
 "startTime": "2020-12-07T01:05:54.157324Z",
 "refreshType": "ViaEnhancedApi",
 "endTime": "2020-12-07T01:05:57.353371Z",
 "status": "Unknown",
 "extendedStatus": "InProgress"
 },
 {
 "requestId": "85a82498-2209-428c-b273-f87b3a1eb905",
 "refreshType": "ViaEnhancedApi",
 "startTime": "2020-12-07T01:05:54.157324Z",
 "endTime": "2020-12-07T01:05:57.353371Z",
 "status": "Unknown",
 "extendedStatus": "InProgress"
 }
]
```

```
 "extendedStatus": "NotStarted"
 }
]
```

### ⓘ Note

Power BI might drop requests if there are too many requests in a short period of time. Power BI does a refresh, queues the next request, and drops all others. By design, you can't query status on dropped requests.

## Response properties

Name	Type	Description
<code>requestId</code>	Guid	The identifier of the refresh request. You need <code>requestId</code> to query for individual refresh operation status or cancel an in-progress refresh operation.
<code>refreshType</code>	String	<code>OnDemand</code> indicates the refresh was triggered interactively through the Power BI portal. <code>Scheduled</code> indicates that a dataset refresh schedule triggered the refresh. <code>ViaApi</code> indicates that an API call triggered the refresh. <code>ViaEnhancedApi</code> indicates that an API call triggered an enhanced refresh.
<code>startTime</code>	String	Date and time of refresh start.
<code>endTime</code>	String	Date and time of refresh end.
<code>status</code>	String	<code>Completed</code> indicates the refresh operation completed successfully. <code>Failed</code> indicates the refresh operation failed. <code>Unknown</code> indicates that the completion state can't be determined. With this status, <code>endTime</code> is empty. <code>Disabled</code> indicates that the refresh was disabled by selective refresh. <code>Cancelled</code> indicates the refresh was canceled successfully.
<code>extendedStatus</code>	String	Augments the <code>status</code> property to provide more information.

### ⓘ Note

In Azure Analysis Services, the completed `status` result is `succeeded`. If you migrate an Azure Analysis Services solution to Power BI, you might have to modify your solutions.

## Limit the number of refresh operations returned

The Power BI REST API supports limiting the requested number of entries in the refresh history by using the optional `$top` parameter. If not specified, the default is all available entries.

HTTP

```
GET https://api.powerbi.com/v1.0/myorg/datasets/{datasetId}/refreshes?$top={$top}
```

## GET /refreshes/<requestId>

To check the status of a refresh operation, use the GET verb on the refresh object by specifying the `requestId`. If the operation is in progress, `status` returns `InProgress`, as in the following example response body:

JSON

```
{
 "startTime": "2020-12-07T02:06:57.1838734Z",
 "endTime": "2020-12-07T02:07:00.4929675Z",
 "type": "Full",
 "status": "InProgress",
 "currentRefreshType": "Full",
 "objects": [
 {
 "table": "DimCustomer",
 "partition": "DimCustomer",
 "status": "InProgress"
 },
 {
 "table": "DimDate",
 "partition": "DimDate",
 "status": "InProgress"
 }
]
}
```

## DELETE /refreshes/<requestId>

To cancel an in-progress enhanced refresh operation, use the `DELETE` verb on the refresh object by specifying the `requestId`.

For example,

HTTP

```
DELETE https://api.powerbi.com/v1.0/myorg/groups/f089354e-8366-4e18-aea3-4cb4a3a50b48/datasets/cfafbeb1-8037-4d0c-896e-a46fb27ff229/refreshes/1344a272-7893-4afa-a4b3-3fb87222fdac
```

## Considerations and limitations

The refresh operation has the following considerations and limitations:

### Standard refresh operations

- You can't cancel scheduled or on-demand manual dataset refreshes by using `DELETE /refreshes/<requestId>`.
- Scheduled and on-demand manual dataset refreshes don't support getting refresh operation details by using `GET /refreshes/<requestId>`.
- Get details and Cancel are new operations for enhanced refresh only. Standard refresh doesn't support these operations.

### Power BI Embedded

If capacity is paused manually in the Power BI portal or by using PowerShell, or a system outage occurs, the status of any ongoing enhanced refresh operation remains `InProgress` for a maximum of six hours. If the capacity resumes within six hours, the refresh operation resumes automatically. If the capacity resumes after longer than six hours, the refresh operation might return a timeout error. You must then restart the refresh operation.

### Dataset eviction

Power BI uses dynamic memory management to optimize capacity memory. If the dataset is evicted from memory during a refresh operation, the following error might return:

JSON

```
{
 "messages": [
 {
 "code": "0xC11C0020",
 "message": "Session cancelled because it is connected to a
database that has been evicted to free up memory for other operations",
 "type": "Error"
 }
]
}
```

The solution is to rerun the refresh operation. To learn more about dynamic memory management and dataset eviction, see [Dataset eviction](#).

## Refresh operation time limits

The maximum amount of time for a single refresh operation is five hours. If the refresh operation doesn't successfully complete within the five-hour limit, and `retryCount` isn't specified or is specified as `0` (the default) in the request body, a timeout error returns.

If `retryCount` specifies `1` or another number, a new refresh operation with a five-hour limit starts. If this retry operation fails, the service continues to retry the refresh operation up to the greatest number of retries that `retryCount` specifies, or the enhanced refresh processing time limit of 24 hours from the beginning of the first refresh request.

When you plan your enhanced dataset refresh solution with the Refresh Dataset REST API, it's important to consider these time limits and the `retryCount` parameter. A successful refresh completion can exceed five hours if an initial refresh operation fails and `retryCount` specifies `1` or more.

For example, if you request a refresh operation with `"retryCount": 1`, and the initial retry operation fails four hours from the start time, a second refresh operation for that request begins. If that second refresh operation succeeds in three hours, the total time for successful execution of the refresh request is seven hours.

If refresh operations regularly fail, exceed the five-hour time limit, or exceed your desired successful refresh operation time, consider reducing the amount of data being refreshed from the data source. You can split refresh into multiple requests, for example a request for each table. You can also specify `partialBatch` in the `commitMode` parameter.

# Code sample

For a C# code sample to get you started, see [RestApiSample](#) on GitHub.

To use the code sample:

1. Clone or download the repo.
2. Open the RestApiSample solution.
3. Find the line `client.BaseAddress = ...` and provide your [base URL](#).

The code sample uses service principal authentication.

## See also

- [Power BI Refresh Dataset REST API](#)
- [Use the Power BI REST APIs](#)

# Incremental refresh and real-time data for datasets

Article • 01/26/2023

Incremental refresh extends scheduled refresh operations by providing automated partition creation and management for dataset tables that frequently load new and updated data. For most datasets, one or more tables contain transaction data that changes often and can grow exponentially, like a fact table in a relational or star database schema. An incremental refresh policy to partition the table, refreshing only the most recent import partitions, and optionally using another DirectQuery partition for real-time data can significantly reduce the amount of data that has to be refreshed. At the same time, this policy ensures that the latest changes at the data source are included in the query results.

With incremental refresh and real-time data:

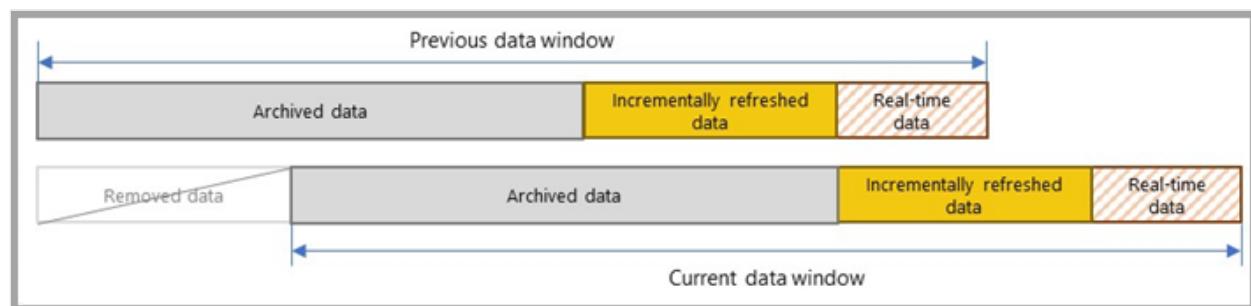
- **Fewer refresh cycles for fast-changing data are needed.** DirectQuery mode gets the latest data updates as queries are processed, without requiring a high refresh cadence.
- **Refreshes are faster.** Only the most recent data that has changed needs to be refreshed.
- **Refreshes are more reliable.** Long-running connections to volatile data sources aren't necessary. Queries to source data run faster, reducing potential for network problems to interfere.
- **Resource consumption is reduced.** Less data to refresh reduces overall consumption of memory and other resources in both Power BI and data source systems.
- **Large datasets are enabled.** Datasets with potentially billions of rows can grow without the need to fully refresh the entire dataset with each refresh operation.
- **Setup is easy.** Incremental refresh *policies* are defined in Power BI Desktop with just a few tasks. When Power BI Desktop publishes the report, the service automatically applies those policies with each refresh.

When you publish a Power BI Desktop model to the service, each table in the new dataset has a single partition. That single partition contains all rows for that table. If the table is large, say with tens of millions of rows or more, a refresh for that table can take a long time and consume an excessive amount of resources.

With incremental refresh, the service dynamically partitions and separates data that needs to be refreshed frequently from data that can be refreshed less frequently. Table

data is filtered by using Power Query date/time parameters with the reserved, case-sensitive names `RangeStart` and `RangeEnd`. When you configure incremental refresh in Power BI Desktop, these parameters are used to filter only a small period of data that's loaded into the model. When Power BI Desktop publishes the report to the Power BI service, with the first refresh operation the service creates incremental refresh and historical partitions, and optionally a real-time DirectQuery partition based on the incremental refresh policy settings. The service then overrides the parameter values to filter and query data for each partition based on date/time values for each row.

With each subsequent refresh, the query filters return only those rows within the refresh period dynamically defined by the parameters. Those rows with a date/time within the refresh period are refreshed. Rows with a date/time no longer within the refresh period then become part of the historical period, which isn't refreshed. If a real-time DirectQuery partition is included in the incremental refresh policy, its filter is also updated so that it picks up any changes that occur after the refresh period. Both the refresh and historical periods are rolled forward. As new incremental refresh partitions are created, refresh partitions no longer in the refresh period become historical partitions. Over time, historical partitions become less granular as they're merged together. When a historical partition is no longer in the historical period defined by the policy, it's removed from the dataset entirely. This behavior is known as a *rolling window pattern*.



The beauty of incremental refresh is that the service handles all of it for you based on the incremental refresh policies you define. In fact, the process and partitions created from it aren't visible in the service. In most cases, a well-defined incremental refresh policy is all that's necessary to significantly improve dataset refresh performance. However, the real-time DirectQuery partition is only supported for datasets in Premium capacities. Power BI Premium also enables more advanced partition and refresh scenarios through the [XML for Analysis \(XMLA\) endpoint](#).

## Requirements

The next sections describe the supported plans and data sources.

## Supported plans

Incremental refresh is supported for Power BI Premium, Premium per user, Power BI Pro, and Power BI Embedded datasets.

Getting the latest data in real time with DirectQuery is only supported for Power BI Premium, Premium per user, and Power BI Embedded datasets.

## Supported data sources

Incremental refresh and real-time data works best for structured, relational data sources like SQL Database and Azure Synapse, but can also work for other data sources. In any case, your data source must support the following:

**Date filtering** - The data source must support some mechanism to filter data by date. For a relational source this is typically a date column of date/time or integer data type on the target table. The RangeStart and RangeEnd parameters, which must be date/time data type, filter table data based on the date column. For date columns of integer surrogate keys in the form of `yyyymmdd`, you can create a function that converts the date/time value in the RangeStart and RangeEnd parameters to match the integer surrogate keys of the date column. To learn more, see [Configure incremental refresh - Convert DateTime to integer](#).

For other data sources, the RangeStart and RangeEnd parameters must be passed to the data source in some way that enables filtering. For file-based data sources where files and folders are organized by date, the RangeStart and RangeEnd parameters can be used to filter the files and folders to select which files to load. For web-based data sources the RangeStart and RangeEnd parameters can be integrated into the HTTP request. For example, the following query can be used for incremental refresh of the traces from an AppInsights instance:

```
Power Query M

let
 strRangeStart = DateTime.ToString(RangeStart, [Format="yyyy-MM-
dd'T'HH:mm:ss'Z'", Culture="en-US"]),
 strRangeEnd = DateTime.ToString(RangeEnd, [Format="yyyy-MM-
dd'T'HH:mm:ss'Z'", Culture="en-US"]),
 Source =
Json.Document(Web.Contents("https://api.applicationinsights.io/v1/apps/<app-
guid>/query",
[Query=[#"query"="traces
| where timestamp >= datetime(" & strRangeStart &")
| where timestamp < datetime("& strRangeEnd &")
",#"x-ms-app"="AAPBI",#"prefer"="ai.response-
```

```

thinning=true"],Timeout=#duration(0,0,4,0)])),
 TypeMap = #table(
 { "AnalyticsTypes", "Type" },
 {
 { "string", Text.Type },
 { "int", Int32.Type },
 { "long", Int64.Type },
 { "real", Double.Type },
 { "timespan", Duration.Type },
 { "datetime", DateTimeZone.Type },
 { "bool", Logical.Type },
 { "guid", Text.Type },
 { "dynamic", Text.Type }
 }),
 DataTable = Source[tables]{0},
 Columns = Table.FromRecords(DataTable[columns]),
 ColumnsWithType = Table.Join(Columns, {"type"}, TypeMap ,
 {"AnalyticsTypes"}),
 Rows = Table.FromRows(DataTable[rows], Columns[name]),
 Table = Table.TransformColumnTypes(Rows, Table.ToList(ColumnsWithType,
 (c) => { c{0}, c{3}}))
in
Table

```

When incremental refresh is configured, a Power Query expression that includes a date/time filter based on the RangeStart and RangeEnd parameters is executed against the data source. If the filter is specified in a query step after the initial source query, it's important that query folding combines the initial query step with the steps reference the RangeStart and RangeEnd parameters. For example, in the following query expression, the `Table.SelectRows` will fold because it immediately follows the `Sql.Database` step, and SQL Server supports folding:

```

Power Query M

let
 Source = Sql.Database("dwdev02", "AdventureWorksDW2017"),
 Data = Source{[Schema="dbo",Item="FactInternetSales"]}[Data],
 "#Filtered Rows" = Table.SelectRows(Data, each [OrderDateKey] >=
 Int32.From(DateTime.ToDateTime(RangeStart,[Format="yyyyMMdd"]))),
 "#Filtered Rows1" = Table.SelectRows(#"Filtered Rows", each [OrderDateKey]
 < Int32.From(DateTime.ToDateTime(RangeEnd,[Format="yyyyMMdd"])))
in
 "#Filtered Rows1"

```

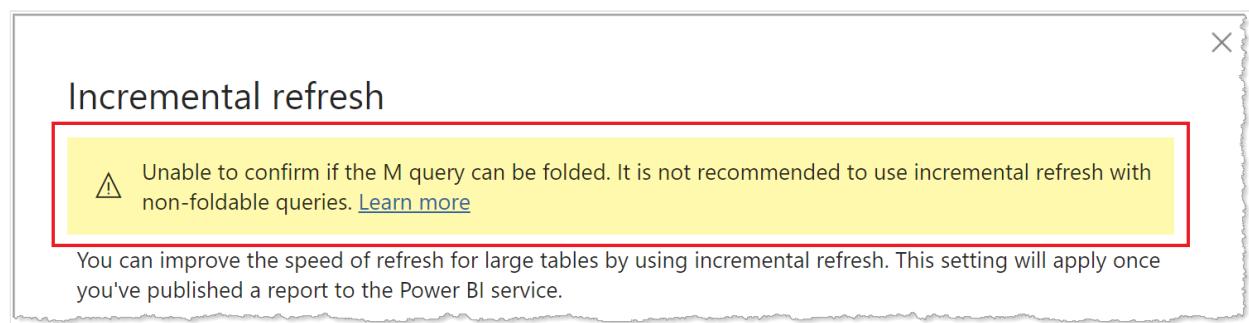
There's no requirement the *final query* support folding. For example in the following expression, we use a non-folding NativeQuery but integrate the RangeStart and RangeEnd parameters directly into SQL:

## Power Query M

```
let
 Query = "select * from dbo.FactInternetSales where OrderDateKey >= '"&
Text.From(Int32.From(DateTime.ToString(RangeStart,"yyyyMMdd"))) &"' and
OrderDateKey < '"& Text.From(Int32.From(
DateTime.ToString(RangeEnd,"yyyyMMdd"))) &"' ",
 Source = Sql.Database("dwdev02","AdventureWorksDW2017"),
 Data = Value.NativeQuery(Source, Query, null, [EnableFolding=false])
in
 Data
```

However, if the incremental refresh policy includes getting real-time data with DirectQuery, non-folding transformations can't be used. If it's a pure import mode policy without real-time data, the query mashup engine might compensate and apply the filter locally, which requires retrieving all rows for the table from the data source. This can cause incremental refresh to be slow, and the process can run out of resources either in the Power BI service or in an On-premises Data Gateway - effectively defeating the purpose of incremental refresh.

Because support for query folding is different for different types of data sources, verification should be performed to ensure the filter logic is included in the queries being run against the data source. In most cases, Power BI Desktop attempts to perform this verification for you when defining the incremental refresh policy. For SQL-based data sources such as SQL Database, Azure Synapse, Oracle, and Teradata, this verification is reliable. However, other data sources may be unable to verify without tracing the queries. If Power BI Desktop is unable to confirm the queries, a warning is shown in the Incremental refresh policy configuration dialog.



If you see this warning and want to verify the necessary query folding is occurring, use the Power Query Diagnostics feature, or trace queries by using a tool supported by the data source, such as SQL Profiler. If query folding isn't occurring, verify the filter logic is included in the query being passed to the data source. If not, it's likely the query includes a transformation that prevents folding.

Before configuring your incremental refresh solution, be sure to thoroughly read and understand [Query folding guidance in Power BI Desktop](#) and [Power Query query](#)

[folding](#). These articles can help you determine whether your data source and queries support query folding.

## Single data source

When you configure incremental refresh and real-time data by using Power BI Desktop, or configure an advanced solution by using Tabular Model Scripting Language (TMSL) or Tabular Object Model (TOM) through the XMLA endpoint, *all partitions*, whether import or DirectQuery, must query data from a single source.

## Other data source types

By using more custom query functions and query logic, incremental refresh can be used with other types of data sources if filters based on `RangeStart` and `RangeEnd` can be passed in a single query, like with data sources such as Excel workbook files stored in a folder, files in SharePoint, and RSS feeds. Keep in mind these are advanced scenarios that require further customization and testing beyond what is described here. Be sure to check the [Community](#) section later in this article for suggestions on how you can find more information about using incremental refresh for unique scenarios.

## Time limits

Regardless of incremental refresh, Power BI Pro datasets have a refresh time limit of **two hours** and don't support getting real-time data with DirectQuery. For datasets in a Premium capacity, the time limit is **five hours**. Refresh operations are process and memory intensive. A full refresh operation can use as much as double the amount of memory required by the dataset alone, because the service maintains a snapshot of the dataset in memory until the refresh operation is complete. Refresh operations can also be process intensive, consuming a significant amount of available CPU resources. Refresh operations must also rely on volatile connections to data sources, and the ability of those data source systems to quickly return query output. The time limit is a safeguard to limit over-consumption of your available resources.

### Note

With Premium capacities, refresh operations performed through the XMLA endpoint have no time limit. To learn more, see [Advanced incremental refresh with the XMLA endpoint](#).

Because incremental refresh optimizes refresh operations at the partition level in the dataset, resource consumption can be significantly reduced. At the same time, even with incremental refresh, unless they go through the XMLA endpoint, refresh operations are bound by those same two-hour and five-hour limits. An effective incremental refresh policy not only reduces the amount of data processed with a refresh operation, but also reduces the amount of unnecessary historical data stored in your dataset.

Queries can also be limited by a default time limit for the data source. Most relational data sources allow overriding time limits in the Power Query M expression. For example, the expression below uses the [SQL Server data-access function](#) to set CommandTimeout to 2 hours. Each period defined by the policy ranges submits a query observing the command timeout setting:

```
Power Query M

let
 Source = Sql.Database("myserver.database.windows.net", "AdventureWorks",
 [CommandTimeout=#duration(0, 2, 0, 0)]),
 dbo_Fact = Source{[Schema="dbo",Item="FactInternetSales"]}[Data],
 #"Filtered Rows" = Table.SelectRows(dbo_Fact, each [OrderDate] >=
RangeStart and [OrderDate] < RangeEnd)
in
 #"Filtered Rows"
```

For *very large* datasets in Premium capacities that likely contain billions of rows, the initial refresh operation can be bootstrapped. Bootstrapping allows the service to create table and partition objects for the dataset, but doesn't load and process data into any of the partitions. By using SQL Server Management Studio, you can set partitions to be processed individually, sequentially, or in parallel, to both reduce the amount of data returned in a single query, and also bypass the five-hour time limit. To learn more, see [Advanced incremental refresh - Prevent timeouts on initial full refresh](#).

## Current date and time

The current date and time is based on the system date at the time of refresh. If scheduled refresh is enabled for the dataset in the service, the specified time zone is taken into account when determining the current date and time. Both individual and scheduled refreshes through the service observe the time zone if available. For example, a refresh that occurs at 8:00 PM Pacific Time (US and Canada) with a time zone specified determines the current date and time based on Pacific Time, not Coordinated Universal Time (UTC), which would return the next day. Refresh operations not invoked through the Power BI service, such as the [TMSL refresh command](#), don't consider the scheduled refresh time zone.

## Scheduled refresh

Keep your data up to date

Configure a data refresh schedule to import data from the data source into the dataset. [Learn more](#)



On

Refresh frequency

Daily 

Time zone

(UTC-08:00) Pacific Time (US and Can)

Time

8  00  PM  

[Add another time](#)

Send refresh failure notifications to

Dataset owner

These contacts:

Enter email addresses

**Apply**

**Discard**

# Configure incremental refresh and real-time data

This section describes important concepts of configuring incremental refresh and real-time data. When you're ready for more detailed step-by-step instructions, see [Configure incremental refresh and real-time data for datasets](#).

Configuring incremental refresh is done in Power BI Desktop. For most models, only a few tasks are required. However, keep the following points in mind:

- After publishing to the Power BI service, you can't publish the same model again from Power BI Desktop. Republishing removes any existing partitions and data already in the dataset. If you're publishing to a Premium capacity, subsequent metadata schema changes can be made with tools such as the open-source ALM Toolkit, or by using TMSL. To learn more, see [Advanced incremental refresh - Metadata-only deployment](#).
- After publishing to the Power BI service, you can't download the dataset back as a .pbix to Power BI Desktop. Because datasets in the service can grow so large, it's

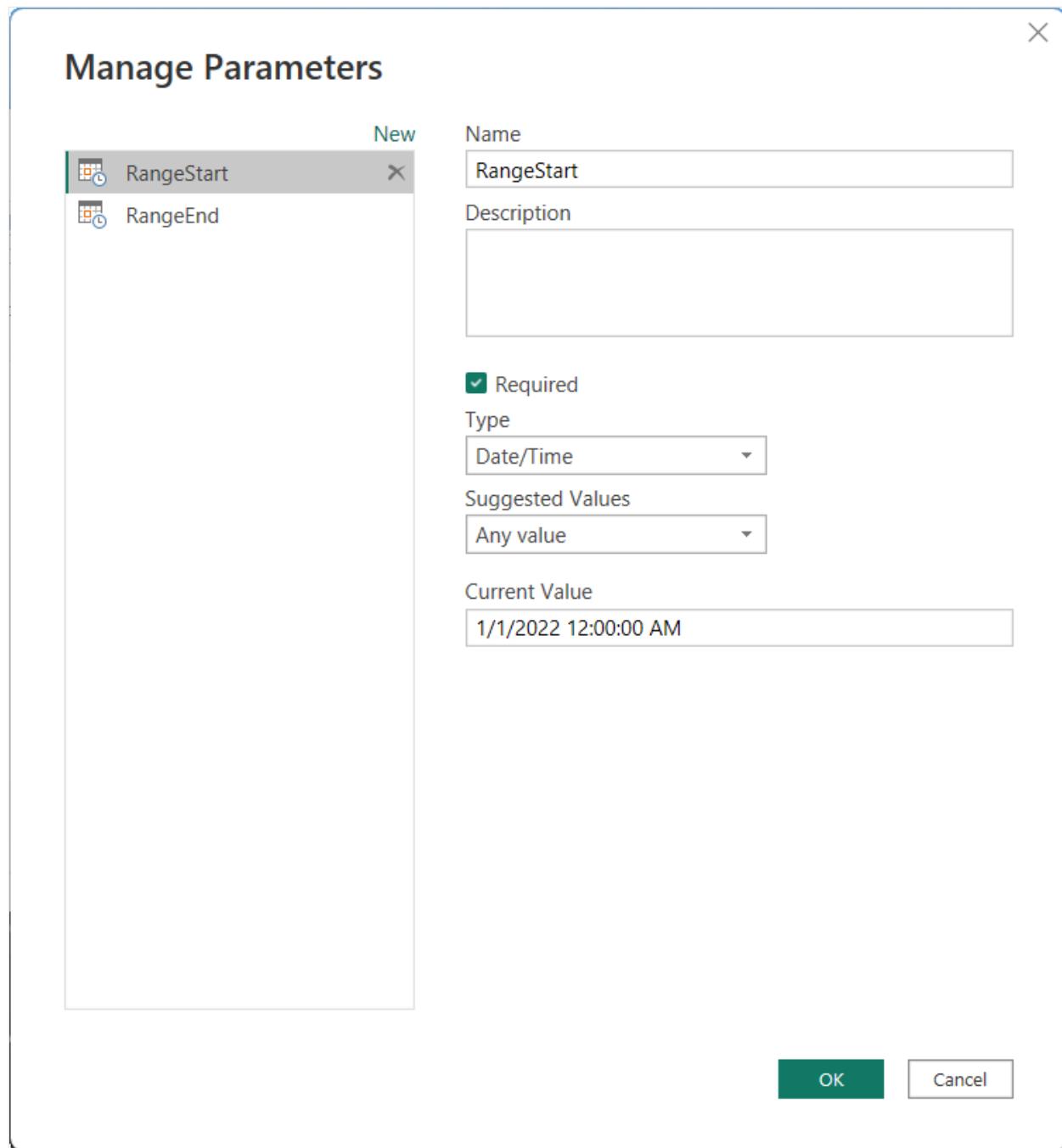
impractical to download and open them on a typical desktop computer.

- When getting real-time data with DirectQuery, you can't publish the dataset to a non-Premium workspace. Incremental refresh with real-time data is only supported with Power BI Premium.

## Create parameters

To configure incremental refresh in Power BI Desktop, you first create two Power Query date/time parameters with the reserved, case-sensitive names `RangeStart` and `RangeEnd`. These parameters, defined in the Manage Parameters dialog in Power Query Editor, are initially used to filter the data loaded into the Power BI Desktop model table to include only those rows with a date/time within that period. `RangeStart` represents the oldest, or earliest date/time, and `RangeEnd` represents the newest, or latest date/time. After the model is published to the service, `RangeStart` and `RangeEnd` are overridden automatically by the service to query data defined by the refresh period specified in the incremental refresh policy settings.

For example, the FactInternetSales data source table averages 10,000 new rows per day. To limit the number of rows initially loaded into the model in Power BI Desktop, specify a two-day period between `RangeStart` and `RangeEnd`.



## Filter data

With the `RangeStart` and `RangeEnd` parameters defined, you apply custom date filters on your table's date column. The filters you apply select a subset of data that's loaded into the model when you select **Apply**.

The screenshot shows the 'Products' table in the Power BI Data Filter dialog. The 'OrderDate' column is selected. The filter pane on the left lists various date filters like 'Equals...', 'Before...', and 'Between...'. The main pane shows a list of dates from July 4, 1996, to July 25, 1996. A red box highlights the 'Custom Filter...' button at the bottom right.

With our FactInternetSales example, after creating filters based on the parameters and applying steps, two days of data (roughly 20,000 rows) are loaded into the model.

## Define policy

After filters have been applied and a subset of data has been loaded into the model, you define an incremental refresh policy for the table. After the model is published to the service, the policy is used by the service to create and manage table partitions and perform refresh operations. To define the policy, you use the **Incremental refresh and real-time data** dialog box to specify both required and optional settings.

## Incremental refresh and real-time data

Refresh large tables faster with incremental refresh. Plus, get the latest data in real time with DirectQuery (Premium only). [Learn more](#)

i These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. [Learn more](#)

### 1. Select table

Orders

### 2. Set import and refresh ranges

Incrementally refresh this table

Archive data starting  Years ▼ before refresh date

Data imported from 1/1/2017 to 11/17/2022 (inclusive)

Incrementally refresh data starting  Days ▼ before refresh date

Data will be incrementally refreshed from 11/18/2022 to 11/20/2022 (inclusive)

### 3. Choose optional settings

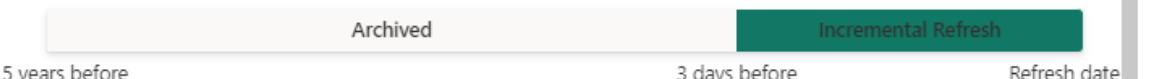
Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)

Selected table cannot be folded for DirectQuery.

Only refresh complete days [Learn more](#)

Detect data changes [Learn more](#)

### 4. Review and apply



## Table

The **Select table** listbox defaults to the table you selected in Data view. Enable incremental refresh for the table with the slider. If the Power Query expression for the table doesn't include a filter based on the `RangeStart` and `RangeEnd` parameters, the toggle isn't available.

## Required settings

The **Archive data starting before refresh date** setting determines the historical period in which rows with a date/time in that period are included in the dataset, plus rows for the

current incomplete historical period, plus rows in the refresh period up to the current date and time.

For example, if you specify five years, the table stores the last five whole years of historical data in year partitions. The table will also include rows for the current year in quarter, month, or day partitions, up to and including the refresh period.

For datasets in Premium capacities, backdated historical partitions can be selectively refreshed at a granularity determined by this setting. To learn more, see [Advanced incremental refresh - Partitions](#).

The **Incrementally refresh data starting before refresh date** setting determines the incremental refresh period in which all rows with a date/time in that period are included in the refresh partitions and refreshed with each refresh operation.

For example, if you specify a refresh period of three days, with each refresh operation, the service overrides the `RangeStart` and `RangeEnd` parameters to create a query for rows with a date/time within a three-day period, with the beginning and ending dependent on the current date and time. Rows with a date/time in the last three days up to the current refresh operation time are refreshed. With this type of policy, you can expect our FactInternetSales dataset table in the service, which averages 10,000 new rows per day, to refresh roughly 30,000 rows with each refresh operation.

Specify a period that includes only the minimum number of rows required to ensure accurate reporting. When you define policies for more than one table, the same `RangeStart` and `RangeEnd` parameters must be used even if different store and refresh periods are defined for each table.

## Optional settings

The **Get the latest data in real time with DirectQuery (Premium only)** setting enables fetching the latest changes from the selected table at the data source beyond the incremental refresh period by using DirectQuery. All rows with a date/time later than the incremental refresh period are included in a DirectQuery partition and fetched from the data source with every dataset query.

For example, if this setting is enabled, with each refresh operation, the service still overrides the `RangeStart` and `RangeEnd` parameters to create a query for rows with a date/time after the refresh period, with the beginning dependent on the current date and time. Rows with a date/time after the current refresh operation time are also included. With this type of policy, the FactInternetSales dataset table in the service includes the latest data updates.

The **Only refresh complete days** setting ensures all rows for the entire day are included in the refresh operation. This setting is optional *unless* you enable the **Get the latest data in real time with DirectQuery (Premium only)** setting. For example, say your refresh is scheduled to run at 4:00 AM every morning. If new rows of data appear in the data source table during those four hours between midnight and 4:00 AM, you don't want to account for them. Some business metrics like barrels per day in the oil and gas industry make no sense with partial days. Another example is refreshing data from a financial system where data for the previous month is approved on the twelfth calendar day of the month. You could set the refresh period to one month and schedule the refresh to run on the twelfth day of the month. With this option selected, it would, for example, refresh January data on February 12.

Keep in mind, unless scheduled refresh is configured for a non-UTC time zone, refresh operations in the service run under UTC time, which can determine the effective date and complete periods.

The **Detect data changes** setting enables even more selective refresh. You can select a date/time column used to identify and refresh only those days where the data has changed. This setting assumes such a column exists in the data source, which is typically for auditing purposes. This column *shouldn't* be the same column used to partition the data with the `RangeStart` and `RangeEnd` parameters. The maximum value of this column is evaluated for each of the periods in the incremental range. If it hasn't changed since the last refresh, there's no need to refresh the period, which could potentially further reduce the days incrementally refreshed from three to one.

The current design requires that the column to detect data changes is persisted and cached into memory. The following techniques can be used to reduce cardinality and memory consumption:

- Persist only the maximum value of the column at time of refresh, perhaps by using a Power Query function.
- Reduce the precision to an acceptable level, given your refresh-frequency requirements.
- Define a custom query for detecting data changes by using the XMLA endpoint, and avoid persisting the column value altogether.

In some cases, enabling the **Detect data changes\*** option can be further enhanced. For example, you may want to avoid persisting a last-update column in the in-memory cache, or enable scenarios where a configuration/instruction table is prepared by extract-transform-load (ETL) processes for flagging only those partitions that need to be refreshed. In cases like these, for Premium capacities, use TMSL and/or the TOM to

override the detect data changes behavior. To learn more, see [Advanced incremental refresh - Custom queries for detect data changes](#).

## Publish

After configuring the incremental refresh policy, you publish the model to the service. When publishing is complete, you can perform the initial refresh operation on the *dataset*.

 **Note**

Datasets with an incremental refresh policy to get the latest data in real time with DirectQuery can only be published to a Premium workspace.

For datasets published to workspaces assigned to Premium capacities, if you think the dataset will grow beyond 1 GB, you can improve refresh operation performance and ensure the dataset doesn't max out size limits by enabling Large dataset storage format *before* performing the first refresh operation in the service. To learn more, see [Large datasets in Power BI Premium](#).

 **Important**

After Power BI Desktop publishes the model to the service, you can't download that .pbix back.

## Refresh

After publishing to the service, you perform an initial refresh operation on the dataset. This refresh should be an individual (manual) refresh so you can monitor progress. The initial refresh operation can take quite a while to complete. Partitions must be created, historical data loaded, objects such as relationships and hierarchies built or rebuilt, and calculated objects recalculated.

Subsequent refresh operations, either individual or scheduled, are much faster because only the incremental refresh partitions are refreshed. Other processing operations must still occur, like merging partitions and recalculation, but it usually takes much less time than the initial refresh.

## Automatic report refresh

For reports that use a dataset with an incremental refresh policy to get the latest data in real time with DirectQuery, it's a good idea to enable automatic page refresh at a fixed interval or based on change detection so that the reports include the latest data without delay. To learn more, see [Automatic page refresh in Power BI](#).

## Advanced incremental refresh

If your dataset is on a Premium capacity with an XMLA endpoint enabled, incremental refresh can be further extended for advanced scenarios. For example, you can use SQL Server Management Studio to view and manage partitions, bootstrap the initial refresh operation, or refresh backdated historical partitions. To learn more, see [Advanced incremental refresh with the XMLA endpoint](#).

## Community

Power BI has a vibrant community where MVPs, BI pros, and peers share expertise in discussion groups, videos, blogs, and more. When learning about incremental refresh, see these resources:

- [Power BI Community](#) ↗
- [Search "Power BI incremental refresh" on Bing](#) ↗
- [Search "Incremental refresh for files" on Bing](#) ↗
- [Search "Keep existing data using incremental refresh" on Bing](#) ↗

## Next steps

- [Configure incremental refresh for datasets](#)
- [Advanced incremental refresh with the XMLA endpoint](#)
- [Troubleshoot incremental refresh](#)
- [Incremental refresh for dataflows](#)

# Configure incremental refresh and real-time data

Article • 02/23/2023

This article describes how to configure incremental refresh and real-time data for [datasets](#). To learn about configuring incremental refresh for dataflows, see [Premium features of dataflows - Incremental refresh](#).

Configuring incremental refresh includes creating RangeStart and RangeEnd parameters, applying filters, and defining an incremental refresh policy. After publishing to the Power BI service, you'll perform an initial refresh operation on the dataset. The initial refresh operation, and subsequent refresh operations apply the incremental refresh policy you defined. Before completing these steps, be sure you fully understand the functionality described in [Incremental refresh and real-time data for datasets](#).

## Create parameters

In this task, you'll use Power Query Editor to create RangeStart and RangeEnd parameters with default values. The default values apply only when filtering the data to be loaded into the model in Power BI Desktop. The values you enter should include only a small amount of the most recent data from your data source. When published to the service, these time range values are overridden by the incremental refresh policy. That is, the policy creates windows of incoming data, one after another.

1. In Power BI Desktop, select **Transform data** on the **Home** ribbon to open Power Query Editor.
2. Select the **Manage Parameters** dropdown and then choose **New Parameter**.
3. In the **Name** field, enter *RangeStart* (case-sensitive). In the **Type** field, select **Date/Time** from the dropdown. In the **Current Value** field, enter a start date and time value.

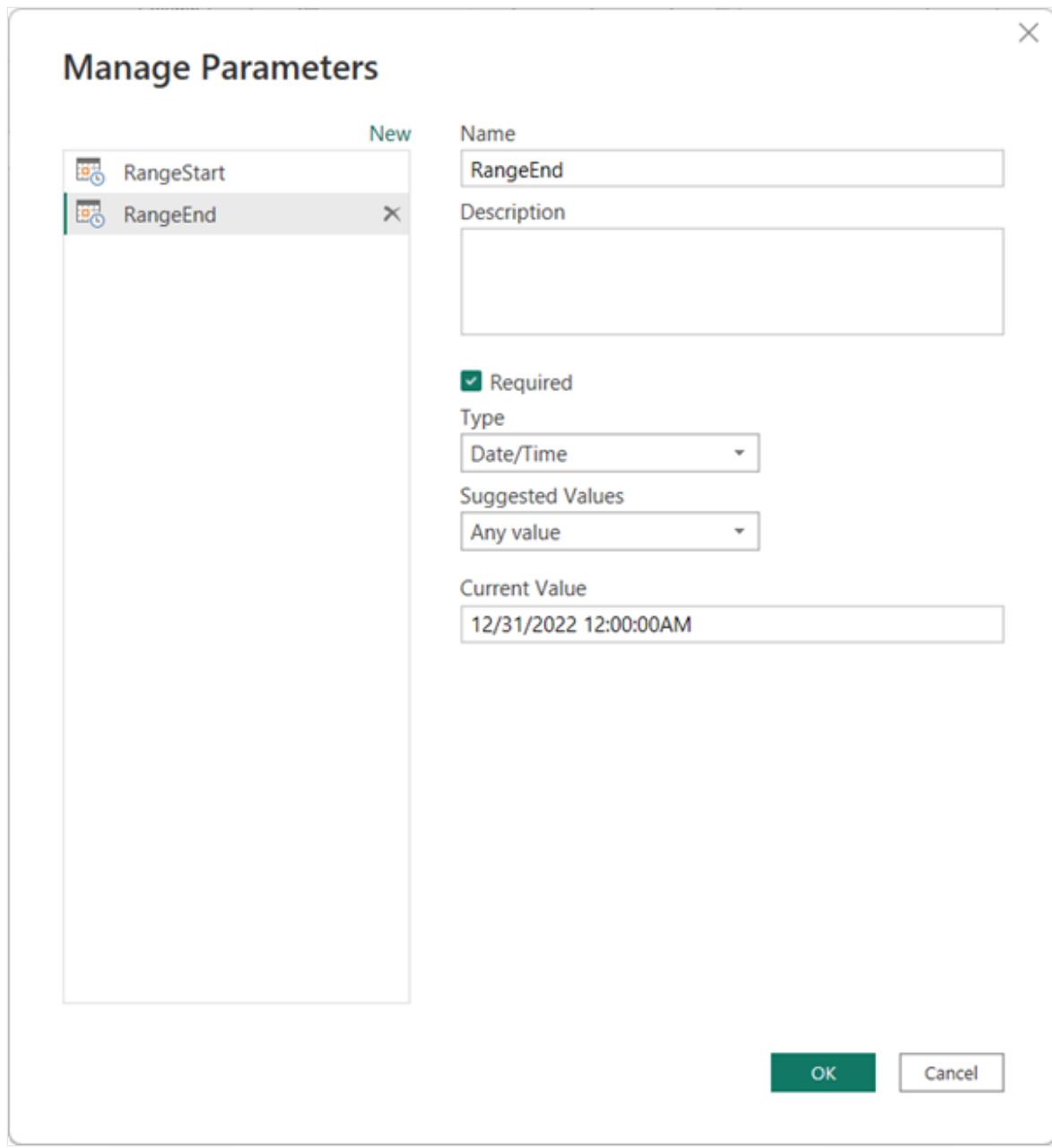
## Manage Parameters

New

 RangeStart	X
Name	
RangeStart	
Description	
<input checked="" type="checkbox"/> Required	
Type	Date/Time
Suggested Values	Any value
Current Value	1/1/2022 12:00:00AM

OK Cancel

4. Select **New** to create a second parameter named *RangeEnd*. In the **Type** field, select **Date/Time**, and then in the **Current Value** field enter an end date and time value. Select **OK**.



Now that you've defined the RangeStart and RangeEnd parameters, you'll filter the data to be loaded into the model based on those parameters.

## Filter data

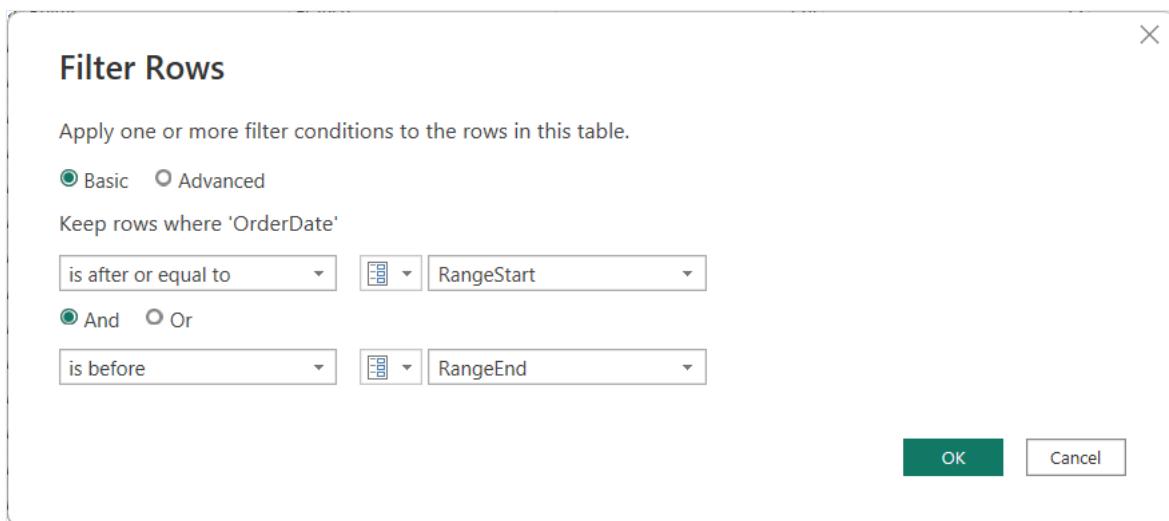
### ! Note

Before continuing with this task, verify your source table has a date column of Date/Time data type. If it doesn't have a Date/Time column, but it has a date column of integer surrogate keys in the form of `yyyymmdd`, follow the steps in [Convert DateTime to integer](#) later in this article to create a function that converts the date/time value in the parameters to match the integer surrogate key of the source table.

You'll now apply a filter based on *conditions* in the RangeStart and RangeEnd parameters.

1. In Power Query Editor, select the date column you want to filter on, and then choose the dropdown arrow > Date Filters > Custom Filter.
2. In **Filter Rows**, to specify the first condition, select **is after** or **is after or equal to**, then choose **Parameter**, and then choose **RangeStart**.

To specify the second condition, if you selected **is after** in the first condition, then choose **is before or equal to**, or if you selected **is after or equal to** in the first condition, then choose **is before** for the second condition, then choose **Parameter**, and then choose **RangeEnd**.



**Important:** Verify queries have an equal to (=) on either RangeStart or RangeEnd, but not both. If the equal to (=) exists on both parameters, a row could satisfy the conditions for two partitions, which could lead to duplicate data in the model. For example, `= Table.SelectRows(#"Changed Type", each [OrderDate] >= RangeStart and [OrderDate] <= RangeEnd)` could result in duplicate data if there's an OrderDate that equals both RangeStart and RangeEnd.

Select **OK** to close.

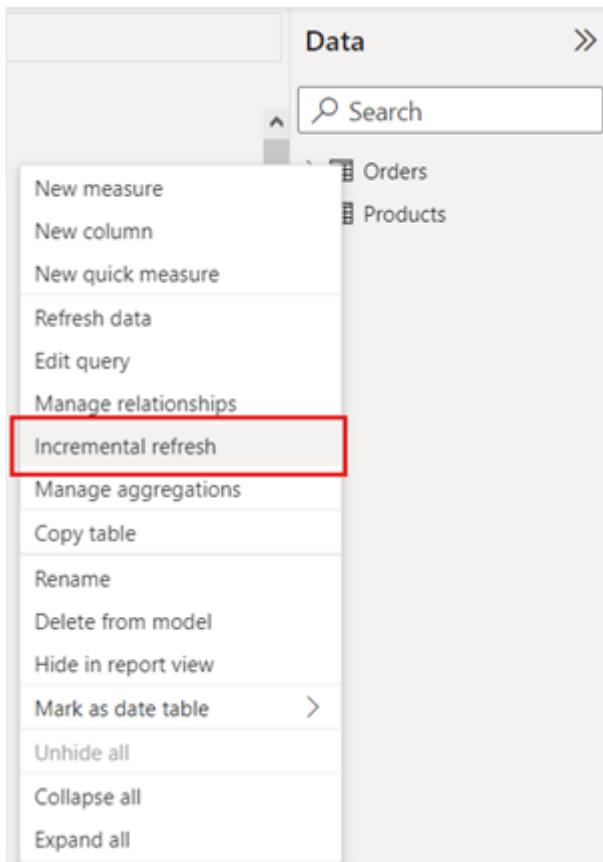
3. On the **Home** ribbon in Power Query Editor, select **Close & Apply**. Power Query loads data based on the filters defined by the RangeStart and RangeEnd parameters, and any other filters you've defined.

Power Query loads only data specified between the RangeStart and RangeEnd parameters. Depending on the amount of data in that period, the table should load quickly. If it seems slow and process-intensive, it's likely [the query isn't folding](#).

# Define policy

After you've defined RangeStart and RangeEnd parameters, and filtered data based on those parameters, you'll define an incremental refresh policy. This policy is applied only after the model is published to the service, and a manual or scheduled refresh operation is performed.

1. In Data view, right-click a table in the **Data pane** and select **Incremental refresh**.



2. In **Incremental refresh and real-time data > Select table**, verify or select the table.

The default value of the **Select table** listbox is the table you selected in Data view.

3. Specify required settings:

In **Set import and refresh ranges > Incrementally refresh this table** move the slider to **On**. If the slider is disabled, it means the Power Query expression for the table doesn't include a filter based on the RangeStart and RangeEnd parameters.

In **Archive data starting**, specify the historical *store* period you want to include in the dataset. All rows with dates in this period will be loaded into the dataset in the service, unless other filters apply.

In **Incrementally refresh data starting**, specify the *refresh* period. All rows with dates in this period will be refreshed in the dataset each time a manual or scheduled refresh operation is performed by the Power BI service.

#### 4. Specify optional settings:

In **Choose optional settings**, select **Get the latest data in real time with DirectQuery (Premium only)** to include the latest data changes that occurred at the data source after the last refresh period. This setting causes the incremental refresh policy to add a DirectQuery partition to the table.

Select **Only refresh complete days** to refresh only whole days. If the refresh operation detects a day isn't complete, rows for that whole day aren't refreshed. This option is automatically enabled when you select **Get the latest data in real time with DirectQuery (Premium only)**.

Select **Detect data changes** to specify a date/time column used to identify and refresh only the days where the data has changed. A date/time column must exist, usually for auditing purposes, at the data source. This column **should not be the same column** used to partition the data with the RangeStart and RangeEnd parameters. The maximum value of this column is evaluated for each of the periods in the incremental range. If it hasn't changed since the last refresh, the current period isn't refreshed. For datasets published to Premium capacities, you can also specify a custom query. To learn more, see [Advanced incremental refresh - Custom queries for detect data changes](#).

Depending on your settings, your policy should look something like this:

## Incremental refresh and real-time data

Refresh large tables faster with incremental refresh. Plus, get the latest data in real time with DirectQuery (Premium only). [Learn more](#)

 These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. [Learn more](#)

### 1. Select table

Orders

### 2. Set import and refresh ranges

 Incrementally refresh this table

Archive data starting  Years  before refresh date

Data imported from 1/1/2017 to 11/17/2022 (inclusive)

Incrementally refresh data starting  Days  before refresh date

Data will be incrementally refreshed from 11/18/2022 to 11/20/2022 (inclusive)

### 3. Choose optional settings

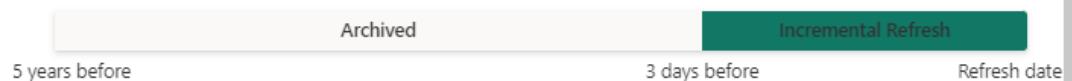
Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)

Selected table cannot be folded for DirectQuery.

Only refresh complete days [Learn more](#)

Detect data changes [Learn more](#)

### 4. Review and apply



5. Review your settings and then select **Apply** to complete the refresh policy. This step doesn't load data.

## Save and publish to the service

Now that your RangeStart and RangeEnd parameters, filtering, and refresh policy settings are complete, save your model, and then publish to the service. If your dataset will become large, be sure to enable [Large dataset storage format](#) before invoking the first refresh in the service.

## Refresh dataset

In the service, refresh the dataset. The first refresh will load both new and updated data in the refresh period as well as historical data for the entire store period. Depending on the amount of data, this refresh can take quite a while. Subsequent refreshes, whether manual or scheduled, are typically much faster because the incremental refresh policy is applied and only data for the period specified in the refresh policy setting is refreshed.

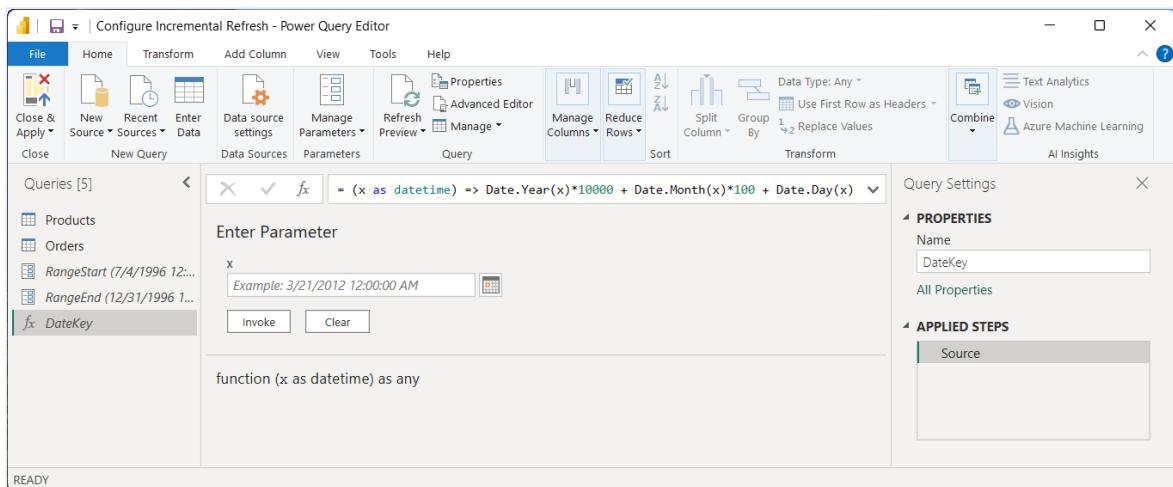
## Convert DateTime to integer

This task is only required if your table uses integer surrogate keys instead of Date/Time values in the date column you use for the RangeStart and RangeEnd filter definition.

The data type of the RangeStart and RangeEnd parameters must be of date/time data type regardless of the data type of the date column. However, for many data sources, tables don't have a column of date/time data type but instead have a date column of integer surrogate keys in the form of `yyyymmdd`. You typically can't convert these integer surrogate keys to the Date/Time data type because the result would be a non-folding query expression, but you can create a function that converts the date/time value in the parameters to match the integer surrogate key of the data source table without losing foldability. The function is then called in a filter step. This conversion step is required if the data source table contains *only* a surrogate key as integer data type.

1. On the **Home** ribbon in Power Query Editor, select the **New Source** dropdown and then choose **Blank Query**.
2. In **Query Settings**, enter a name, for example, DateKey, and then in the formula editor, enter the following formula:

```
= (x as datetime) => Date.Year(x)*10000 + Date.Month(x)*100 + Date.Day(x)
```



3. To test the formula, in **Enter Parameter**, enter a date/time value, and then select **Invoke**. If the formula is correct, an integer value for the date is returned. After

verifying, delete this new **Invoked Function** query.

4. In **Queries**, select the table, and then edit the query formula to call the function with the RangeStart and RangeEnd parameters.

```
= Table.SelectRows(#"Reordered Column OrderDateKey", each [OrderDateKey] > DateKey(RangeStart) and [OrderDateKey] <= DateKey(RangeEnd))
```

The screenshot shows the Power Query Editor interface with the following details:

- File** tab selected.
- Home** tab selected.
- Transform** tab selected.
- Add Column**, **View**, **Tools**, and **Help** tabs are visible.
- General** ribbon tab selected.
- Queries** list: Products, Orders (selected), RangeStart (7/4/1996 12:00:00 AM), RangeEnd (12/31/1996 11:59:59 PM), DateKey.
- Formula Bar**: `= Table.SelectRows(#"Reordered Column OrderDateKey", each [OrderDateKey] > DateKey(RangeStart) and [OrderDateKey] <= DateKey(RangeEnd))`
- Preview Area**: Shows a table with columns: OrderDate, OrderDateKey, OrderID. The data consists of 14 rows from July 5, 1996, to July 19, 1996.
- Query Settings** pane: Name set to Orders.
- Applied Steps** pane: Shows the steps taken: Source, Navigation, Reordered Columns, Added Custom, Reordered Column OrderDate..., and Filtered Rows (selected).
- Bottom Status**: 19 COLUMNS, 151 ROWS, Column profiling based on top 1000 rows, PREVIEW DOWNLOADED AT 2:43 PM.

## See also

- [Troubleshoot configuring incremental refresh](#)
- [Advanced incremental refresh with the XMLA endpoint](#)
- [Configure scheduled refresh](#)

# Advanced incremental refresh and real-time data with the XMLA endpoint

Article • 01/23/2023

Datasets in a Premium capacity with the [XMLA endpoint](#) enabled for read/write operations allow more advanced dataset refresh, partition management, and metadata only deployments through tool, scripting, and API support. In addition, refresh operations through the XMLA endpoint aren't limited to [48 refreshes per day](#), and the [scheduled refresh time limit](#) isn't imposed.

## Partitions

Dataset table partitions aren't visible and can't be managed by using Power BI Desktop or the Power BI service. For datasets in a workspace assigned to a Premium capacity, partitions can be managed through the XMLA endpoint by using tools like SQL Server Management Studio (SSMS), the open-source Tabular Editor, scripted with Tabular Model Scripting Language (TMSL), and programmatically with the Tabular Object Model (TOM).

When you first publish a model to the Power BI service, each table in the new dataset has one partition. For tables with no incremental refresh policy, that one partition contains all rows of data for that table, unless filters have been applied. For tables with an incremental refresh policy, that one initial partition only exists because Power BI hasn't yet applied the policy. You configure the initial partition in Power BI Desktop when you define the date/time range filter for your table based on the `RangeStart` and `RangeEnd` parameters, and any other filters applied in Power Query Editor. This initial partition contains only those rows of data that meet your filter criteria.

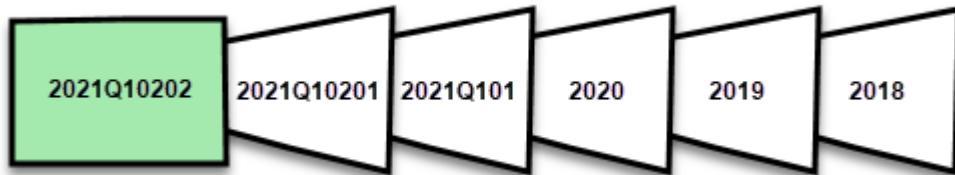
When you perform the *first* dataset refresh operation, tables with no incremental refresh policy refresh all rows contained in that table's default single partition. For tables with an incremental refresh policy, refresh and historical partitions are automatically created and rows are loaded into them according to the date/time for each row. If the incremental refresh policy includes getting data in real time, Power BI also adds a DirectQuery partition to the table.

This first refresh operation can take quite some time depending on the amount of data that needs to be loaded from the data source. The complexity of the model can also be a significant factor because refresh operations must do more processing and

recalculation. This operation can be bootstrapped. For more information, see [Prevent timeouts on initial full refresh](#).

Partitions are created for and named by period granularity: Years, quarters, months, and days. The most recent partitions, the *refresh* partitions, contains rows in the refresh period you specify in the policy. Historical partitions contain rows by complete period up to the refresh period. If real time is enabled, a DirectQuery partition picks up any data changes that occurred after the end date of the refresh period. Granularity for refresh and historical partitions is dependent on the refresh and historical (store) periods you choose when defining the policy.

For example, if today's date is February 2, 2021 and our **FactInternetSales** table at the data source contains rows up through today, if our policy specifies to include real-time changes, refresh rows in the last one day refresh period, and store rows in the last three years historical period. Then with the first refresh operation, a DirectQuery partition is created for changes in the future, a new import partition is created for today's rows, a historical partition is created for yesterday, a whole day period, February 1, 2021. A historical partition is created for the previous whole month period (January 2021), a historical partition is created for the previous whole year period (2020), and historical partitions for 2019 and 2018 whole year periods are created. No whole quarter partitions are created because we haven't yet completed the first full quarter of 2021.



With each refresh operation, only the refresh period partitions are refreshed and the date filter of the DirectQuery partition is updated to include only those changes that occur after the current refresh period. A new refresh partition is created for new rows with a new date/time within the updated refresh period, and existing rows with a date/time already within existing partitions in the refresh period are refreshed with updates. Rows with a date/time older than the refresh period are no longer refreshed.

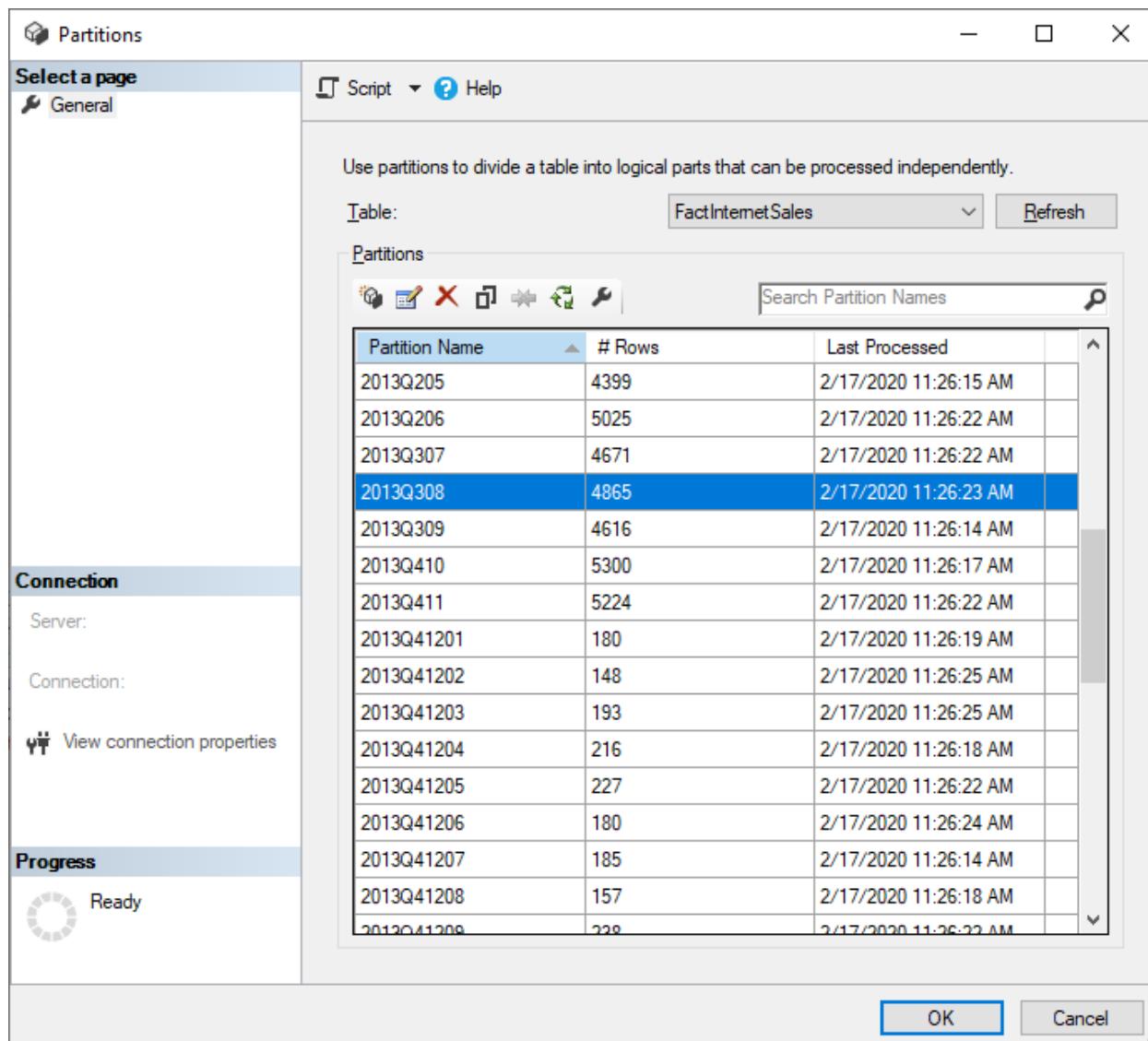
As whole periods close, partitions are merged. For example, if a one-day refresh period and three year historical store period is specified in the policy, on the first day of the month, all day partitions for the previous month are merged into a month partition. On the first day of a new quarter, all three previous month partitions are merged into a quarter partition. On the first day of a new year, all four previous quarter partitions are merged into a year partition.

A dataset always retains partitions for the entire historical store period plus whole period partitions up through the current refresh period. In the example, a full three years of historical data are retained in partitions for 2018, 2019, 2020, and also partitions for the 2021Q101 month period, the 2021Q10201 day period, and the current day refresh period partition. Because the example retains historical data for three *years*, the 2018 partition is retained until the first refresh on January 1, 2022.

With Power BI incremental refresh and real-time data, the service handles the partition management for you based on the policy. While the service can handle all of the partition management for you, by using tools through the XMLA endpoint, you can selectively refresh partitions individually, sequentially, or in parallel.

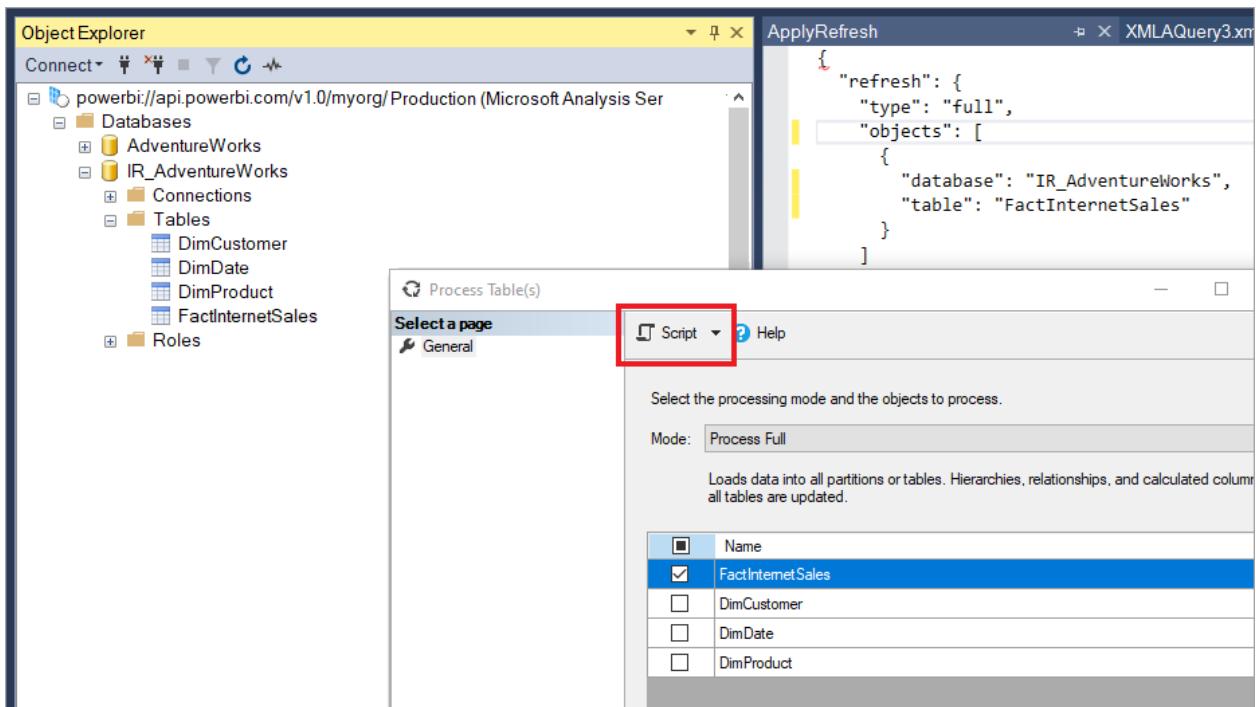
## Refresh management with SQL Server Management Studio

SQL Server Management Studio (SSMS) can be used to view and manage partitions created by the application of incremental refresh policies. By using SSMS you can, for example, refresh a specific historical partition not in the incremental refresh period to perform a back-dated update without having to refresh all historical data. SSMS can also be used when bootstrapping to load historical data for large datasets by incrementally adding/refreshing historical partitions in batches.



## Override incremental refresh behavior

With SSMS, you also have more control over how to invoke refreshes by using [Tabular Model Scripting Language](#) and the [Tabular Object Model](#). For example, in SSMS, in Object Explorer, right-click a table and then select the **Process Table** menu option, and then select the **Script** button to generate a TMSL refresh command.



These parameters can be used with the TMSL refresh command to override the default incremental refresh behavior:

- **applyRefreshPolicy**. If a table has an incremental refresh policy defined, `applyRefreshPolicy` determines if the policy is applied or not. If the policy isn't applied, a process full operation leaves partition definitions unchanged and all partitions in the table are fully refreshed. Default value is true.
- **effectiveDate**. If an incremental refresh policy is being applied, it needs to know the current date to determine rolling window ranges for the incremental refresh and historical periods. The `effectiveDate` parameter allows you to override the current date. This parameter is useful for testing, demos, and business scenarios where data is incrementally refreshed up to a date in the past or the future, for example, budgets in the future. The default value is the current date.

JSON

```
{
 "refresh": {
 "type": "full",

 "applyRefreshPolicy": true,
 "effectiveDate": "12/31/2013",

 "objects": [
 {
 "database": "IR_AdventureWorks",
 "table": "FactInternetSales"
 }
]
}
```

```
}
```

To learn more about overriding default incremental refresh behavior with TMSL, see [Refresh command](#).

## Ensuring optimal performance

With each refresh operation, the Power BI service might send initialization queries to the data source for each incremental refresh partition. You might be able to improve incremental refresh performance by reducing the number of initialization queries by ensuring the following configuration:

- The table you configure incremental refresh for should get data from a single data source. If the table gets data from more than one data source, the number of queries sent by the service for each refresh operation is multiplied by the number of data sources, potentially reducing refresh performance. Ensure the query for the incremental refresh table is for a single data source.
- For solutions with both incremental refresh of import partitions and real-time data with Direct Query, *all partitions* must query data from a single data source.
- If your security requirements allow, set the Data source privacy level setting to *Organizational* or *Public*. By default, the privacy level is *Private*, however this level can prevent data from being exchanged with other cloud sources. To set the privacy level, select the **More options** menu and then choose **Settings > Data source credentials > Edit credentials > Privacy level setting for this data source**. If Privacy level is set in the Power BI Desktop model before publishing to the service, it isn't transferred to the service when you publish. You must still set it in dataset settings in the service. To learn more, see [Privacy levels](#).
- If using an On-premises Data Gateway, be sure you're using version 3000.77.3 or higher.

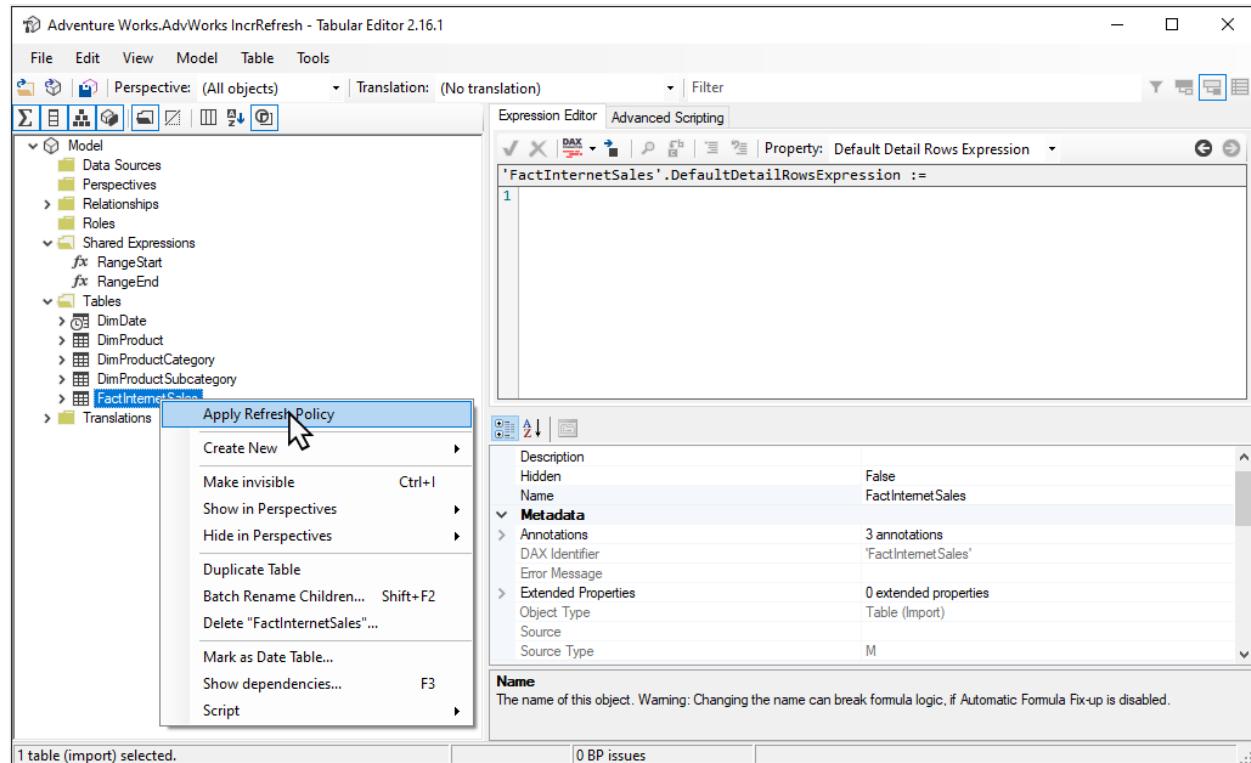
## Prevent timeouts on initial full refresh

After you publish to the Power BI service, the initial full refresh operation for the dataset creates partitions for the incremental refresh table, loads, and processes historical data for the entire period defined in the incremental refresh policy. For some datasets that load and process large amounts of data, the amount of time the initial refresh operation takes can exceed the refresh time limit imposed by the service or a query time limit imposed by the data source.

Bootstrapping the initial refresh operation allows the service to create partition objects for the incremental refresh table, but not load and process historical data into any of the partitions. SSMS is then used to selectively process partitions. Depending on the amount of data to be loaded for each partition, you can process each partition sequentially or in small batches to reduce the potential for one or more of those partitions to cause a timeout. The following methods work for any data source.

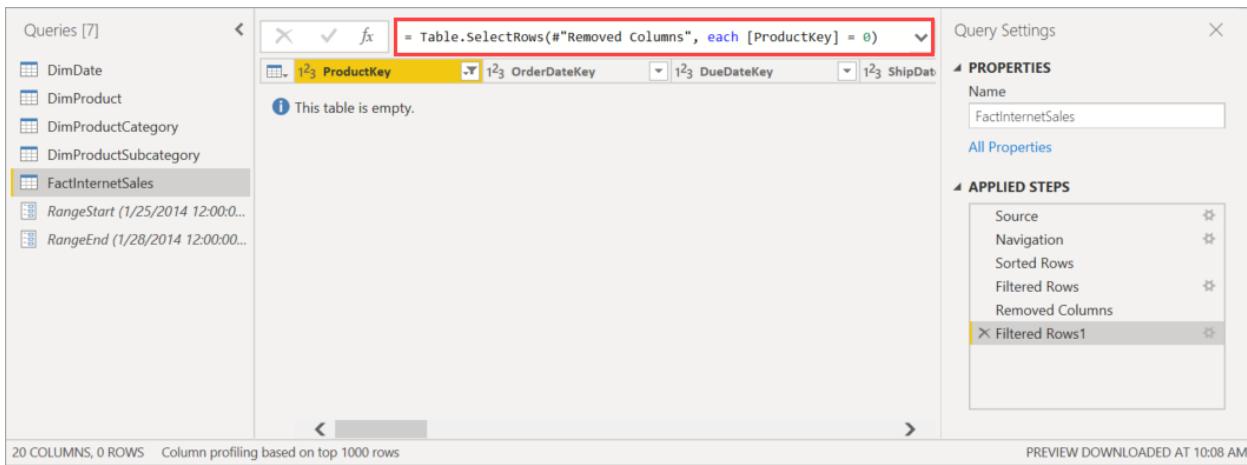
## Apply Refresh Policy

The open-source [Tabular Editor 2](#) tool provides an easy way to bootstrap an initial refresh operation. After publishing a model with an incremental refresh policy defined for it from Power BI Desktop to the service, connect to the dataset by using the XMLA endpoint in Read/Write mode. Run **Apply Refresh Policy** on the incremental refresh table. With only the policy applied, partitions are created but no data is loaded into them. Then connect with SSMS to refresh the partitions sequentially or in batches to load and process the data. For more information, see [Incremental refresh](#) in the Tabular editor documentation.



## Power Query filter for empty partitions

Prior to publishing the model to the service, in Power Query Editor, add another filter to the `ProductKey` column that filters out any value other than 0, effectively filtering out *all* data from the `FactInternetSales` table.



After selecting **Close & Apply** in Power Query Editor, defining the incremental refresh policy, and saving the model, the model is published to the service. From the service, the initial refresh operation is run on the dataset. Partitions for the **FactInternetSales** table are created according to the policy, but no data is loaded and processed because all data is filtered out.

After the initial refresh operation is complete, back in Power Query Editor, the other filter on the **ProductKey** column is removed. After selecting **Close & Apply** in Power Query Editor and saving the model, the model *is not published again*. If the model is published again, it overwrites the incremental refresh policy settings and forces a full refresh on the dataset when a subsequent refresh operation is performed from the service. Instead, perform a [metadata only deployment](#) by using the Application Lifecycle Management (ALM) Toolkit that removes the filter on the **ProductKey** column from the *dataset*. SSMS can then be used to selectively process partitions. When all partitions have been fully processed, which must include a process recalculation on all partitions, from SSMS, subsequent refresh operations on the dataset from the service refresh only the incremental refresh partitions.

### Tip

Be sure to check out videos, blogs, and more provided by Power BI's community of BI experts.

- Search for "Prevent timeouts with incremental refresh" on [Bing](#).

To learn more about processing tables and partitions from SSMS, see [Process database, table, or partitions \(Analysis Services\)](#). To learn more about processing datasets, tables, and partitions by using TMSL, see [Refresh command \(TMSL\)](#).

## Custom queries for detect data changes

TMSL and TOM can be used to override the detected data changes behavior. Not only can this method be used to avoid persisting the last-update column in the in-memory cache, it can enable scenarios where a configuration or instruction table is prepared by extract, transform, and load (ETL) processes for flagging only the partitions that need to be refreshed. This method can create a more efficient incremental refresh process where only the required periods are refreshed, no matter how long ago data updates took place.

The `pollingExpression` is intended to be a lightweight M expression or name of another M query. It must return a scalar value and will be executed for each partition. If the value returned is different to what it was the last time an incremental refresh occurred, the partition is flagged for full processing.

The following example covers all 120 months in the historical period for backdated changes. Specifying 120 months instead of 10 years means data compression might not be quite as efficient, but avoids having to refresh a whole historical year, which would be more expensive when a month would be sufficient for a backdated change.

#### JSON

```
"refreshPolicy": {
 "policyType": "basic",
 "rollingWindowGranularity": "month",
 "rollingWindowPeriods": 120,
 "incrementalGranularity": "month",
 "incrementalPeriods": 120,
 "pollingExpression": "<M expression or name of custom polling query>",
 "sourceExpression": [
 "let ..."
]
}
```

#### Tip

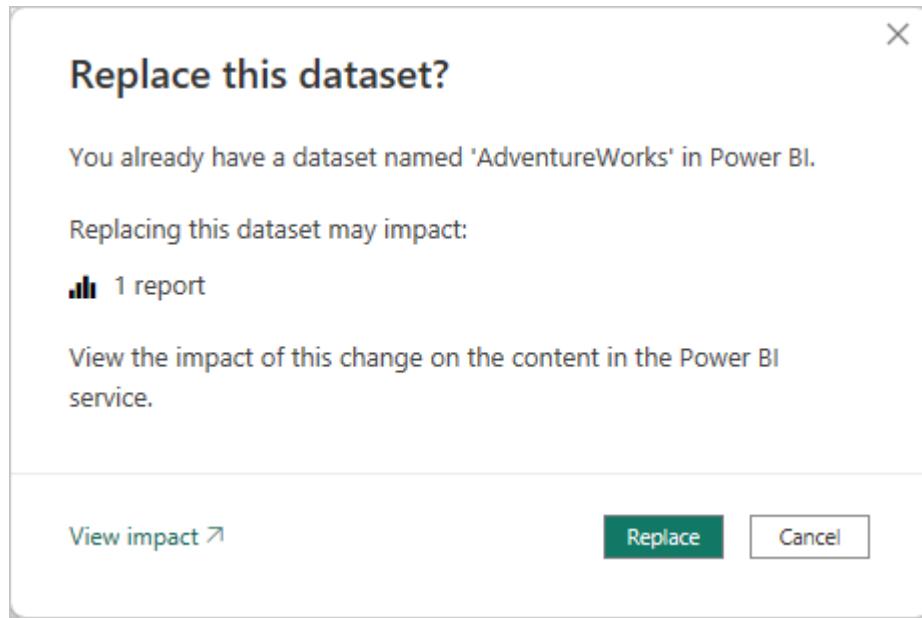
Be sure to check out videos, blogs, and more provided by Power BI's community of BI experts.

- Search for "Power BI Incremental refresh detect data changes" on [Bing](#).

## Metadata only deployment

When publishing a new version of a `.pbix` file from Power BI Desktop to a workspace, if a dataset with the same name already exists, you're prompted to replace the existing

dataset.



In some cases, you might not want to replace the dataset, especially with incremental refresh. The dataset in Power BI Desktop could be much smaller than the one in the Power BI service. If the dataset in the Power BI service has an incremental refresh policy applied, it might have several years of historical data that will be lost if the dataset is replaced. Refreshing all the historical data could take hours and result in system downtime for users.

Instead, it's better to perform a metadata only deployment, which allows deployment of new objects without losing the historical data. For example, if you've added a few measures you can deploy only the new measures without needing to refresh the data, saving time.

For workspaces assigned to a Premium capacity configured for XMLA endpoint read/write, compatible tools enable metadata only deployment. For example, the ALM Toolkit is a schema diff tool for Power BI datasets and can be used to perform deployment of metadata only.

Download and install the latest version of the ALM Toolkit from the [Analysis Services Git repo](#). Step-by-step guidance on using ALM Toolkit isn't included in Microsoft documentation. ALM Toolkit documentation links and information on supportability are available on the **Help** ribbon. To perform a metadata only deployment, perform a comparison and select the running Power BI Desktop instance as the source, and the existing dataset in the Power BI service as the target. Consider the differences displayed and skip the update of the table with incremental refresh partitions or use the **Options** dialog to retain partitions for table updates. Validate the selection to ensure the integrity of the target model and then update.

The screenshot shows the ALM Toolkit application window. At the top, there are tabs for File, Home, and Help, along with icons for Compare, Select Actions, Validate Selection, Update, Generate Script, Options, and Report Differences. To the right, there are links for MAQ Software and Power BI. Below the tabs, there are two tabs: Source (PBI Desktop/SSDT: localhost:63185;IR\_AdventureWorks) and Target (Dataset: powerbi://api.powerbi.com/v1.0/myorg/Production;IR\_AdventureWorks). The main content area is a table comparing objects between the source and target. The table has columns for Type, Source Name, Status, Target Name, and Action. The Action column contains icons for Skip (grey circle), Create (green plus), and Update (blue checkmark). The table lists various objects: Model, DateKey, RangeEnd, RangeStart, DimCustomer, DimDate, DimProduct, and FactInternetSales. Under FactInternetSales, it details relationships (CustomerKey, DueDateKey, OrderDateKey, ProductKey, ShipDateKey) and measures (Discount Amount, Product Cost, Sales Amount, Tax Amount). The status for most objects is 'Same Definition' except for DateKey, RangeEnd, RangeStart, and FactInternetSales which are 'Missing in Target'. The Action column for FactInternetSales is set to 'Update'. Below the table, a code editor shows TMSL script for the FactInternetSales table, comparing source and target definitions across lines 199 to 216.

## Adding an incremental refresh policy and real-time data programmatically

You can also use the TMSL and TOM to add an incremental refresh policy to an existing dataset through the XMLA endpoint.

### ① Note

To avoid compatibility issues, make sure you use the latest version of the Analysis Services client libraries. For example, to work with Hybrid policies, the version must be 19.27.1.8 or higher.

The process includes of the following steps:

1. Ensure the target dataset has the required minimum compatibility level. In SSMS, right-click the [dataset name] > **Properties** > **Compatibility Level**. To increase the compatibility level, either use a createOrReplace TMSL script or check the following TOM sample code for an example.

text

- a. Import policy - 1550
- b. Hybrid policy - 1565

2. Add the `RangeStart` and `RangeEnd` parameters to the dataset expressions. If necessary, also add a function to convert Date/Time values to date keys.
3. Define a `RefreshPolicy` object with the desired archiving (rolling window) and incremental refresh periods as well as a source expression that filters the target table based on the `RangeStart` and `RangeEnd` parameters. Set the refresh policy mode to *Import* or *Hybrid* depending on your real-time data requirements. Hybrid causes Power BI to add a DirectQuery partition to the table to fetch the latest changes from the data source that occurred after the last refresh time.
4. Add the refresh policy to the table and perform a full refresh so that Power BI partitions the table according to your requirements.

The following code sample demonstrates how to perform the previous steps by using TOM. If you want to use this sample as is, you must have a copy for the AdventureWorksDW database and import the **FactInternetSales** table into a dataset. The code sample assumes that the `RangeStart` and `RangeEnd` parameters and the `DateKey` function don't exist in the dataset. Just import the **FactInternetSales** table and publish the dataset to a workspace on Power BI Premium. Then update the `workspaceUrl` so that the code sample can connect to your dataset. Update any more code lines as necessary.

C#

```
using System;
using TOM = Microsoft.AnalysisServices.Tabular;
namespace Hybrid_Tables
{
 class Program
 {
 static string workspaceUrl = "<Enter your Workspace URL here>";
 static string databaseName = "AdventureWorks";
 static string tableName = "FactInternetSales";
 static void Main(string[] args)
 {
 using (var server = new TOM.Server())
 {
 // Connect to the dataset.
 server.Connect(workspaceUrl);
 TOM.Database database =
server.Databases.FindByName(databaseName);
 if (database == null)
 {
 throw new ApplicationException("Database cannot be
found!");
 }
 }
 }
 }
}
```

```

 }
 if(database.CompatibilityLevel < 1565)
 {
 database.CompatibilityLevel = 1565;
 database.Update();
 }
 TOM.Model model = database.Model;
 // Add RangeStart, RangeEnd, and DateKey function.
 model.Expressions.Add(new TOM.NamedExpression {
 Name = "RangeStart",
 Kind = TOM.ExpressionKind.M,
 Expression = "#datetime(2021, 12, 30, 0, 0, 0) meta
[IsParameterQuery=true, Type=\\"DateTime\\", IsParameterQueryRequired=true]"
 });
 model.Expressions.Add(new TOM.NamedExpression
 {
 Name = "RangeEnd",
 Kind = TOM.ExpressionKind.M,
 Expression = "#datetime(2021, 12, 31, 0, 0, 0) meta
[IsParameterQuery=true, Type=\\"DateTime\\", IsParameterQueryRequired=true]"
 });
 model.Expressions.Add(new TOM.NamedExpression
 {
 Name = "DateKey",
 Kind = TOM.ExpressionKind.M,
 Expression =
 "let\n" +
 " Source = (x as datetime) => Date.Year(x)*10000
+ Date.Month(x)*100 + Date.Day(x)\n" +
 "in\n" +
 " Source"
 });
 // Apply a RefreshPolicy with Real-Time to the target table.
 TOM.Table salesTable = model.Tables[tableName];
 TOM.RefreshPolicy hybridPolicy = new TOM.BasicRefreshPolicy
 {
 Mode = TOM.RefreshPolicyMode.Hybrid,
 IncrementalPeriodsOffset = -1,
 RollingWindowPeriods = 1,
 RollingWindowGranularity =
TOM.RefreshGranularityType.Year,
 IncrementalPeriods = 1,
 IncrementalGranularity = TOM.RefreshGranularityType.Day,
 SourceExpression =
 "let\n" +
 " Source =
Sql.Database(\"demopm.database.windows.net\", \"AdventureWorksDW\"),\n" +
 " dbo_FactInternetSales =
Source{[Schema=\"dbo\", Item=\"FactInternetSales\"]}[Data],\n" +
 " #\"Filtered Rows\" =
Table.SelectRows(dbo_FactInternetSales, each [OrderDateKey] >=
DateKey(RangeStart) and [OrderDateKey] < DateKey(RangeEnd))\n" +
 "in\n" +
 " #\"Filtered Rows\""
 };
 }
}

```

```
 salesTable.RefreshPolicy = hybridPolicy;
 model.RequestRefresh(TOM.RefreshType.Full);
 model.SaveChanges();
}
Console.WriteLine("{0}{1}", Environment.NewLine, "Press [Enter]
to exit...");
Console.ReadLine();
}
}
```

## Next steps

- Partitions in tabular models
- External tools in Power BI Desktop
- Configure scheduled refresh
- Troubleshoot incremental refresh and real-time data

# Troubleshoot incremental refresh and real-time data

Article • 01/26/2023

There are two phases when implementing an incremental refresh and real-time data solution, the first being configuring parameters, filtering, and defining a policy in Power BI Desktop, and the second being the initial dataset refresh operation and subsequent refreshes in the service. This article discusses troubleshooting separately for each of these phases.

Having partitioned the table in the Power BI service, it's important to keep in mind that incrementally refreshed tables that are also getting real-time data with DirectQuery are now operating in hybrid mode, meaning they operate in both import and DirectQuery mode. Any tables with relationships to such an incrementally refreshed hybrid table must use Dual mode so that they can be used in import and DirectQuery mode without performance penalties. Moreover, report visuals might cache results to avoid sending queries back to the data source, which would prevent the table from picking up the latest data updates in real time. The final troubleshooting section covers these hybrid-mode issues.

Before troubleshooting incremental refresh and real-time data, be sure to review [Incremental refresh for datasets and real-time data](#) and step-by-step information in [Configure incremental refresh and real-time data](#).

## Configuring in Power BI Desktop

Most problems that occur when configuring incremental refresh and real-time data have to do with query folding. As described in [Incremental refresh for datasets overview - Supported data sources](#), your data source must support query folding.

### Problem: Loading data takes too long

In Power Query Editor, after selecting **Apply**, loading data takes an excessive amount of time and computer resources. There are several potential causes.

### Cause: Data type mismatch

This issue can be caused by a data type mismatch where `Date/Time` is the required data type for the `RangeStart` and `RangeEnd` parameters, but the table date column on which

the filters are applied aren't `Date/Time` data type, or vice-versa. Both the parameters data type and the filtered data column must be `Date/Time` data type and the format must be the same. If not, the query can't be folded.

## Solution: Verify data type

Verify the date/time column for the incremental refresh table is of `Date/Time` data type. If your table doesn't contain a column of `Date/Time` data type, but instead uses an integer data type, you can create a function that converts the date/time value in the `RangeStart` and `RangeEnd` parameters to match the integer surrogate key of the data source table. To learn more, see [Configure incremental refresh - Convert DateTime to integer](#).

## Cause: The data source doesn't support query folding

As described in [Incremental refresh and real-time data for datasets - Requirements](#), incremental refresh is designed for data sources that support query folding. Make sure data source queries are being folded in Power BI Desktop before publishing to the service, where query folding issues can be significantly compounded. This approach is especially important when including real-time data in an incremental refresh policy because the real-time DirectQuery partition requires query folding.

## Solution: Verify and test queries

In most cases, a warning is shown in the Incremental refresh policy dialog indicating if the query to be executed against the data source doesn't support query folding. However, in some cases it might be necessary to further ensure query folding is possible. If possible, monitor the query being passed to the data source by using a tool like SQL Profiler. A query with filters based on `RangeStart` and `RangeEnd` must be executed in a single query.

You can also specify a short date/time period in the `RangeStart` and `RangeEnd` parameters that include no more than a few thousand rows. If the load of filtered data from the data source to the model takes a long time and is process intensive, it likely means the query isn't being folded.

If you determine the query isn't being folded, refer to [Query folding guidance in Power BI Desktop](#) and [Power Query query folding](#) for help with identifying what might be preventing query folding and how, or if, the data source can even support query folding.

# Dataset refresh in the service

Troubleshooting incremental refresh issues in the service differ depending on the type of capacity your dataset has been published to. Datasets on Premium capacities support using tools like SQL Server Management Studio (SSMS) to view and selectively refresh individual partitions. Power BI Pro datasets on the other hand don't provide tool access through the XMLA endpoint, so troubleshooting incremental refresh issues might require a little more trial and error.

## Problem: Initial refresh times out

Scheduled refresh for Power BI Pro datasets on a shared capacity have a time limit of two hours. This time limit is increased to five hours for datasets in a Premium capacity. Data source systems might also impose a query return size limit or query timeout.

## Cause: Data source queries aren't being folded

While problems with query folding can usually be determined in Power BI Desktop before publishing to the service, it's possible that dataset refresh queries aren't being folded, leading to excessive refresh times and query mashup engine resource utilization. This situation happens because a query is created for every partition in the dataset. If the queries aren't being folded, and data isn't being filtered at the data source, the engine then attempts to filter the data.

## Solution: Verify query folding

Use a tracing tool at the data source to determine the query being passed for each partition is a single query that includes a filter based on the RangeStart and RangeEnd parameters. If not, verify query folding is occurring in the Power BI Desktop model when loading a small filtered amount of data into the model. If not, get it fixed in the model first, perform a metadata only update to the dataset (by using XMLA endpoint), or if a Power BI Pro dataset on a shared capacity, delete the incomplete dataset in the service, republish, and try an initial refresh operation again.

If you determine queries aren't being folded, refer to [Query folding guidance in Power BI Desktop](#) and [Power Query query folding](#) for help with identifying what might be preventing query folding.

## Cause: Data loaded into partitions is too large

## Solution: Reduce dataset size

In many cases, the timeout is caused by the amount of data that must be queried and loaded into the dataset partitions exceeds the time limits imposed by the capacity. Reduce the size or complexity of your dataset, or consider breaking the dataset into smaller pieces.

## Solution: Enable Large dataset storage format

For datasets published to Premium capacities, if the dataset grows beyond 1 GB or more, you can improve refresh operation performance and ensure the dataset doesn't max out size limits by enabling Large dataset storage format *before* performing the first refresh operation in the service. To learn more, see [Large datasets in Power BI Premium](#).

## Solution: Bootstrap initial refresh

For datasets published to Premium capacities, you can bootstrap the initial refresh operation. Bootstrapping allows the service to create table and partition objects for the dataset, but not load and process historical data into any of the partitions. To learn more, see [Advanced incremental refresh - Prevent timeouts on initial full refresh](#).

## Cause: Data source query timeout

Queries can be limited by a default time limit for the data source.

## Solution: Override the time limit in the query expression

Many data sources allow overriding time limit in the query expression. To learn more, see [Incremental refresh for datasets - Time limits](#).

## Problem: Refresh fails because of duplicate values

### Cause: Post dates have changed

With a refresh operation, only data that has changed at the data source is refreshed in the dataset. As the data is divided by a date, it's recommended that post (transaction) dates aren't changed.

If a date is changed accidentally, then two issues can occur: Users notice some totals changed in the historical data (that isn't supposed to happen), or during a refresh an

error is returned indicating a unique value isn't in fact unique. For the latter, this situation can happen when the table with incremental refresh configured is used in a 1:N relationship with another table as the 1 side and should have unique values. When the data is changed for a specific ID, that ID then appears in another partition and the engine detects the value isn't unique.

## Solution: Refresh specific partitions

Where there's a business need to change some past data from the dates, a possible solution is to use SSMS to refresh all partitions from the point where the change is located up to the current refresh partition, thus keeping the 1 side of the relationship unique.

## Problem: Data is truncated

### Cause: Data source query limit has been exceeded

Some data sources, like Azure Data Explorer, Log Analytics, and Application Insights, have a limit of 64 MB (compressed) on data that can be returned for an external query. Azure Data Explorer might return an explicit error, but for others like Log Analytics and Application Insights, the data returned is truncated.

### Solution: Specify smaller refresh and store periods

Specify smaller refresh and store periods in the policy. For example, if you specified a refresh period of one year, and a query error is returned or data returned is truncated, try a refresh period of 12 months. You want to ensure queries for the current refresh partition or any historical partitions based on the Refresh and Store periods don't return more than 64 MB of data.

## Problem: Refresh fails because of partition-key conflicts

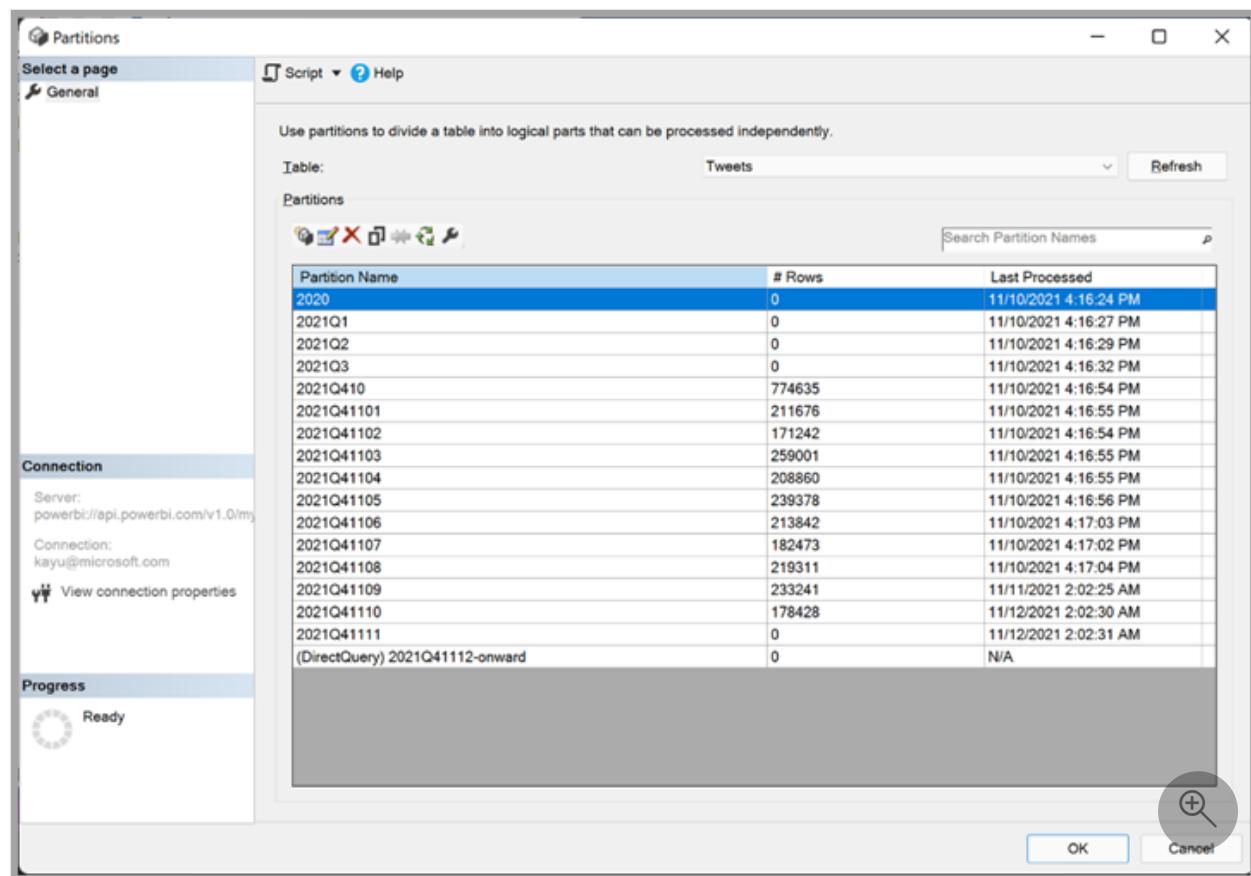
### Cause: Date in the date column at the data source is updated

The filter on the date column is used to dynamically partition the data into period ranges in the Power BI service. Incremental refresh isn't designed to support cases where the filtered date column is updated in the source system. An update is interpreted as an insertion and a deletion, not an actual update. If the deletion occurs in the historical

range and not the incremental range, it isn't picked up, which can cause data refresh failures due to partition-key conflicts.

## Hybrid mode in the service (Preview)

When Power BI applies an incremental refresh policy with real-time data, it turns the incrementally refreshed table into a hybrid table that operates in both import and DirectQuery mode. Notice the DirectQuery partition at the end of the following partitions list of a sample table. The presence of a DirectQuery partition has implications for related tables and report visuals that query this table.



The screenshot shows the 'Partitions' dialog box in Power BI. On the left, there's a sidebar with 'Select a page' (General), 'Connection' (Server: powerbi://api.powerbi.com/v1.0/myorg, Connection: kayu@microsoft.com, View connection properties), and 'Progress' (Ready). The main area has tabs for 'Script' and 'Help'. A message says 'Use partitions to divide a table into logical parts that can be processed independently.' Below that, 'Table:' is set to 'Tweets'. The 'Partitions' section contains a table with columns 'Partition Name', '# Rows', and 'Last Processed'. The table lists partitions from 2020 to 2021Q4111, followed by a '(DirectQuery) 2021Q41112-onward' entry. The 'Last Processed' column shows dates ranging from 11/10/2021 to 11/12/2021. At the bottom right are 'OK' and 'Cancel' buttons, and a magnifying glass icon.

Partition Name	# Rows	Last Processed
2020	0	11/10/2021 4:16:24 PM
2021Q1	0	11/10/2021 4:16:27 PM
2021Q2	0	11/10/2021 4:16:29 PM
2021Q3	0	11/10/2021 4:16:32 PM
2021Q410	774635	11/10/2021 4:16:54 PM
2021Q41101	211676	11/10/2021 4:16:55 PM
2021Q41102	171242	11/10/2021 4:16:54 PM
2021Q41103	259001	11/10/2021 4:16:55 PM
2021Q41104	208860	11/10/2021 4:16:55 PM
2021Q41105	239378	11/10/2021 4:16:56 PM
2021Q41106	213842	11/10/2021 4:17:03 PM
2021Q41107	182473	11/10/2021 4:17:02 PM
2021Q41108	219311	11/10/2021 4:17:04 PM
2021Q41109	233241	11/11/2021 2:02:25 AM
2021Q41110	178428	11/12/2021 2:02:30 AM
2021Q41111	0	11/12/2021 2:02:31 AM
(DirectQuery) 2021Q41112-onward	0	N/A

## Problem: Query performance is poor

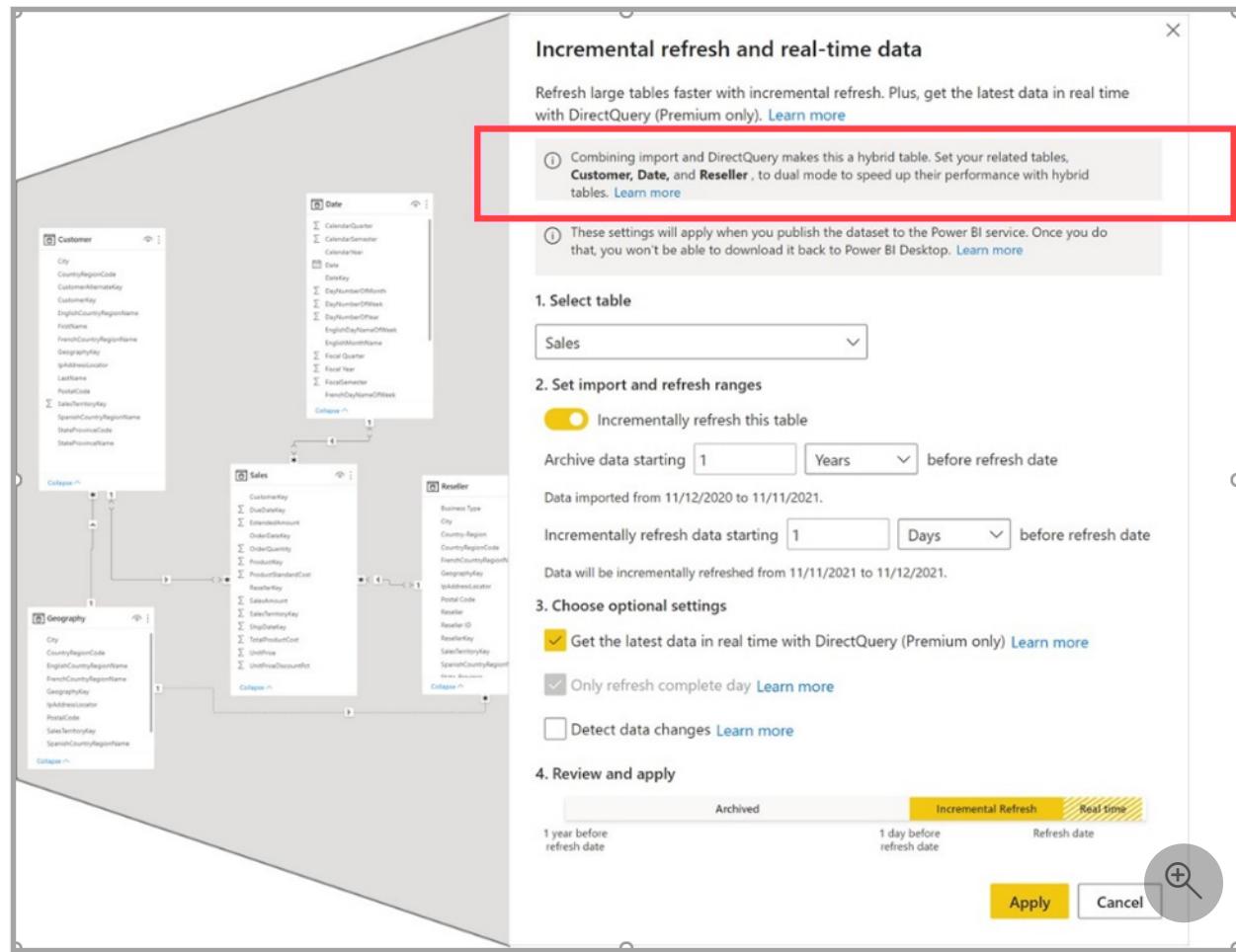
### Cause: Related tables aren't in Dual mode

Hybrid tables operating in both import and DirectQuery mode require any related tables to operate in Dual mode so that they can act as either cached or not cached, depending on the context of the query that's submitted to the Power BI dataset. Dual mode enables Power BI to reduce the number of limited relationships in the dataset and generate efficient data source queries to ensure good performance. Limited relationships can't be pushed to the data source requiring Power BI to retrieve more

data than necessary. Because Dual tables can act as either DirectQuery or Import tables, this situation is avoided.

## Solution: Convert related tables to Dual mode

When configuring an incremental refresh policy, Power BI Desktop reminds you to switch any related tables to Dual mode when you select **Get the latest data in real time with DirectQuery (Premium only)**. In addition, make sure you review all existing table relationships in Model View.



Tables currently operating in DirectQuery mode, are easily switched to Dual mode. In the table properties, under Advanced, select Dual from the Storage mode listbox. Tables currently operating in import mode, however, require manual work. Dual tables have the same functional constraints as DirectQuery tables. Power BI Desktop therefore can't convert import tables because they might rely on other functionality not available in Dual mode. You must manually recreate these tables in DirectQuery mode and then convert them to Dual mode. To learn more, see [Manage storage mode in Power BI Desktop](#).

## Problem: Report visuals don't show the latest data

## Cause: Power BI caches query results improve performance and reduce back-end load

By default, Power BI caches query results, so that queries of report visuals can be processed quickly even if they're based on DirectQuery. Avoiding unnecessary data source queries improves performance and reduces data source load, but it might also mean that the latest data changes at the source aren't included in the results.

## Solution: Configure automatic page refresh

To keep fetching the latest data changes from the source, configure automatic page refresh for your reports in the Power BI service. Automatic page refresh can be performed in fixed intervals, such as five seconds or ten minutes. When that specific interval is reached, all visuals in that page send an update query to the data source and update accordingly. Alternatively, you can refresh visuals on a page based on detecting changes in the data. This approach requires a change detection measure that Power BI then uses to poll the data source for changes. Change detection is only supported in workspaces that are part of a Premium capacity. To learn more, see [Automatic page refresh in Power BI](#).

## See also

- [Data refresh in Power BI](#)
- [Advanced incremental refresh with the XMLA endpoint](#)
- [Incremental refresh for dataflows](#)

# Refresh datasets created from local Power BI Desktop files

Article • 02/08/2023

Power BI supports **Refresh now** and **Schedule refresh** for datasets that are created from imported local Power BI Desktop files. Power BI supports refresh for any of the following data sources that you connect to or load with **Get data** and Power Query Editor.

## Power BI gateway (personal mode)

On-premises data gateway (personal mode) supports refresh for the following data sources:

- All online data sources that appear in Power BI Desktop **Get data** and Power Query Editor.
- All on-premises data sources that appear in Power BI Desktop **Get data** and Power Query Editor, except for Hadoop files (HDFS) and Microsoft Exchange.

## On-premises data gateway

On-premises data gateway supports refresh for the following data sources:

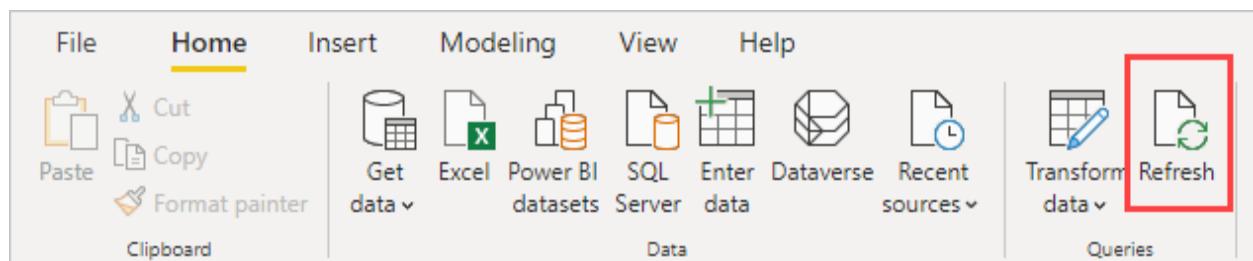
- Analysis Services Tabular
- Analysis Services Multidimensional
- SQL Server
- SAP HANA
- Oracle
- Teradata
- File
- Folder
- SharePoint list (on-premises)
- Web
- OData
- IBM DB2
- MySQL
- Sybase
- SAP BW
- IBM Informix Database
- ODBC

## ⓘ Note

A gateway must be installed and running for Power BI to connect to on-premises data sources and refresh the dataset.

# Refresh in Power BI Desktop vs. Power BI service

You can do a one-time, manual refresh in Power BI Desktop by selecting **Refresh** on the **Home** tab of the ribbon. When you select **Refresh**, the data in the file's model refreshes with updated data from the original data source.



This kind of refresh from within Power BI Desktop is different from manual or scheduled refresh in the Power BI service. It's important to understand the distinction.

When you import your Power BI Desktop file from a local drive, data and other information about the model is loaded into a dataset in the Power BI service. You base your reports in the Power BI service on the dataset. You refresh the data in the Power BI service, not in Power BI Desktop, because you based your reports on the dataset in the service. Because the data sources are external, you can manually refresh the dataset by using **Refresh now**, or you can set up a refresh schedule by using **Schedule refresh**.

When you refresh the dataset, Power BI doesn't connect to the file on the local drive to query for updated data. Power BI uses information in the dataset to connect directly to the data sources, query for updated data, and then load the updated data into the dataset.

## ⓘ Note

Refreshed data in the dataset doesn't synchronize back to the file on the local drive.

# Scheduled refresh

When you set up a refresh schedule, Power BI connects directly to the data sources by using the connection information and credentials in the dataset. Power BI queries for updated data, then loads the updated data into the dataset. Any visualizations in reports and dashboards that are based on that dataset also update.

For details on how to set up scheduled refresh, see [Configure scheduled refresh](#).

## Troubleshooting

When things go wrong, it's usually because Power BI can't sign into data sources. Make sure Power BI can sign into your data sources. If the dataset connects to an on-premises data source, the gateway might be offline. If the password you use to sign in to the data source changes, or Power BI gets signed out, try signing into your data sources again in [Data source credentials](#).

Be sure to set **Send refresh failure notifications to Dataset owner**, so you know right away if a scheduled refresh fails.

Sometimes refreshing data might not go as you expect. This issue often involves a gateway. For tools and known issues, see the following gateway troubleshooting articles:

- [Troubleshoot the on-premises data gateway](#)
- [Troubleshoot the Power BI Gateway - Personal](#)

More questions? [Try asking the Power BI Community](#) ↗

# Refresh a dataset stored on OneDrive or SharePoint Online

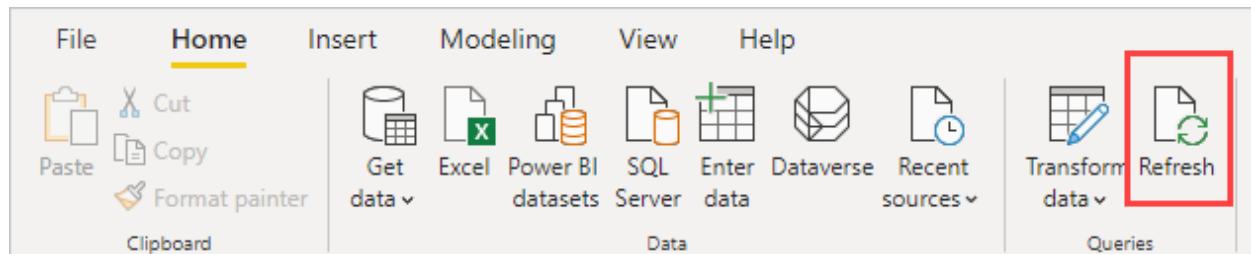
Article • 01/09/2023

Importing files from OneDrive or SharePoint Online into the Power BI service is a great way to make sure your work in Power BI Desktop stays in sync with the Power BI service.

## Advantages of storing a Power BI Desktop file on OneDrive or SharePoint Online

When you store a Power BI Desktop file on OneDrive or SharePoint Online, any data you've loaded into your file's model is imported into the dataset. Any reports you've created from the file are loaded into **Reports** in the Power BI service. Let's say you make changes to your file on OneDrive or SharePoint Online. These changes can include adding new measures, changing column names, or editing visualizations. Once you save the file, Power BI service syncs with those changes too, usually within about an hour.

You can do a one-time, manual refresh right in Power BI Desktop by selecting **Refresh** on the **Home** ribbon. When you select **Refresh**, you refresh the file's model with updated data from the original data source. This kind of refresh happens entirely from within the Power BI Desktop application itself. It's different from a manual or scheduled refresh in Power BI, and it's important to understand the distinction.



When you import your Power BI Desktop file from OneDrive or SharePoint Online, you load data and model information into a dataset in Power BI. You'll want to refresh the dataset in the Power BI service because that's what your reports are based on. Because the data sources are external, you can manually refresh the dataset by using **Refresh now** or you can set up a refresh schedule by using **Schedule refresh**.

The screenshot shows the Power BI desktop application. On the left, the navigation pane is visible with sections for Home, Favorites, Recent, Apps, Shared with me, Workspaces, and Sales and Marketing. Under Sales and Marketing, there are DASHBOARDS, REPORTS, WORKBOOKS, DATASETS, and DATAFLOWS. The DATASETS section contains 'Contoso Q2 Division Sales' and 'Customer Profitability Sample ...'. The DATAFLOWS section contains 'Sales'. At the bottom, there is a 'Get Data' button. In the center, a pie chart titled 'TotalSales by Division' is displayed, showing segments for TVs, Appliances, and Cameras. A context menu is open over the chart, listing options: RENAME, REMOVE, SCHEDULE REFRESH, REFRESH NOW, ANALYZE IN EXCEL, QUICK INSIGHTS, DOWNLOAD .PBIX, SECURITY, and MANAGE PERMISSIONS. The 'SCHEDULE REFRESH' option is highlighted with a red arrow pointing to it. The text 'Division Sales' is written next to the arrow. The top right of the screen shows the status bar with 'Explore', 'Refresh', and 'Pin Live' buttons.

When you refresh the dataset, Power BI doesn't connect to the file on OneDrive or SharePoint Online to query for updated data. It uses information in the dataset to connect directly to the data sources and query for updated data. Then, it loads that data into the dataset. This refreshed data in the dataset isn't synchronized back to the file on OneDrive or SharePoint Online.

## What's supported?

Power BI supports **Refresh** and **Schedule refresh** for datasets created from Power BI Desktop files imported from a local drive where you use **Get data** or **Power Query Editor** to connect to and load data from the following data sources.

### Note

Onedrive refresh for live connection datasets is supported. However, changing the live connection dataset, from one dataset to another in an already published report, is not supported in the OneDrive refresh scenario.

## Power BI Gateway - Personal

- All online data sources shown in Power BI Desktop's **Get data** and **Power Query Editor**.
- All on-premises data sources shown in Power BI Desktop's **Get data** and **Power Query Editor** except for Hadoop File (HDFS) and Microsoft Exchange.

## On-premises data gateway

On-premises data gateway supports refresh for the following data sources:

- Analysis Services Tabular
- Analysis Services Multidimensional
- SQL Server
- SAP HANA
- Oracle
- Teradata
- File
- Folder
- SharePoint list (on-premises)
- Web
- OData
- IBM DB2
- MySQL
- Sybase
- SAP BW
- IBM Informix Database
- ODBC

### ⓘ Note

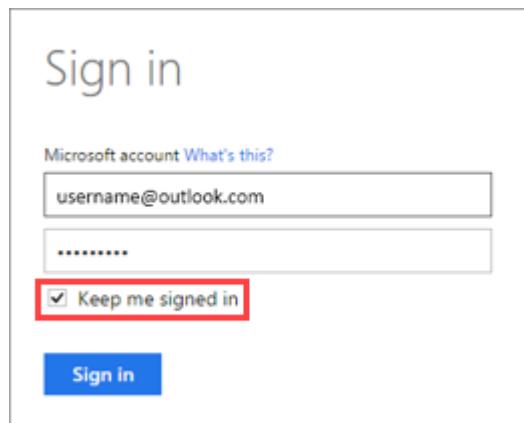
A gateway must be installed and running in order for Power BI to connect to on-premises data sources and refresh the dataset.

# OneDrive or OneDrive for work or school. What's the difference?

If you have both a personal OneDrive and OneDrive for work or school, you should keep any files you want to import into Power BI in OneDrive for work or school. Here's why: You likely use two different accounts to sign into them.

When you connect to OneDrive for work or school in Power BI, connection is easy because your Power BI account is often the same account as your OneDrive for work or school account. With personal OneDrive, you usually sign in with a different [Microsoft account](#).

When you sign in with your Microsoft account, be sure to select **Keep me signed in**. Power BI can then synchronize any updates you make in the file in Power BI Desktop with datasets in Power BI.



If you've changed your Microsoft credentials, you can't synchronize changes between your file on OneDrive and the dataset in Power BI. You need to connect to and import your file again from OneDrive.

## How do I schedule refresh?

When you set up a refresh schedule, Power BI connects directly to the data sources. Power BI uses connection information and credentials in the dataset to query for updated data. Then Power BI loads the updated data into the dataset. It then updates any report visualizations and dashboards based on that dataset in the Power BI service.

For details on how to set up schedule refresh, see [Configure scheduled refresh](#).

## When things go wrong

When things go wrong, it's usually because Power BI can't sign into data sources. Things may also go wrong if the dataset tries to connect to an on-premises data source but the gateway is offline. To avoid these issues, make sure Power BI can sign into data sources. Try signing into your data sources in **Data Source Credentials**. Sometimes the password you use to sign into a data source changes or Power BI gets signed out from a data source.

When you save your changes to the Power BI Desktop file on OneDrive and you don't see those changes in Power BI within an hour or so, it could be because Power BI can't connect to your OneDrive. Try connecting to the file on OneDrive again. If you're prompted to sign in, make sure you select **Keep me signed in**. Because Power BI wasn't able to connect to your OneDrive to synchronize with the file, you'll need to import your file again.

- Import of sensitivity-labeled *.pbix* files (both protected and unprotected) stored on OneDrive or SharePoint Online, as well as on-demand and automatic dataset refresh from such files, is supported, with the exception of the following scenarios:
  - Protected live-connected *.pbix* files and protected Azure Analysis Services *.pbix* files. Refresh will fail. Neither report content nor label will be updated.
  - Labeled unprotected Live Connect *.pbix* files: Report content will be updated but label will not be updated.
  - When the *.pbix* file has had a new sensitivity label applied that the dataset owner doesn't have usage rights to. In this case, refresh will fail. Neither report content nor label will be updated.
  - If the dataset owner's access token for OneDrive/SharePoint has expired. In this case, refresh will fail. Neither report content nor label will be updated.

## Troubleshooting

Sometimes refreshing data may not go as expected. You'll typically run into data refresh issues when you're connected with a gateway. Take a look at the gateway troubleshooting articles for tools and known issues.

[Troubleshooting the On-premises data gateway](#)

[Troubleshooting the Power BI Gateway - Personal](#)

More questions? Try asking the [Power BI Community](#).

# Refresh a dataset created from an Excel workbook on a local drive

Article • 08/28/2023

## What's supported?

### Important

The following capabilities are deprecated and will no longer be available starting September 29th, 2023:

- Upload of local workbooks to Power BI workspaces will no longer be allowed.
- Configuring scheduling of refresh and refresh now for Excel files that don't already have scheduled refresh configured will no longer be allowed.

The following capabilities are deprecated and will no longer be available starting October 31, 2023:

- Scheduled refresh and refresh now for existing Excel files that were previously configured for scheduled refresh will no longer be allowed.
- Local workbooks uploaded to Power BI workspaces will no longer open in Power BI.

After October 31, 2023:

- You can download existing local workbooks from your Power BI workspace.
- You can publish your Excel data model as a Power BI dataset and schedule refresh.
- You can import Excel workbooks from OneDrive and SharePoint Document libraries to view them in Power BI.

If your organization uses these capabilities, see more details in [Migrating your Excel workbooks](#).

In Power BI, Refresh Now and Schedule Refresh is supported for datasets created from Excel workbooks imported from a local drive where Power Query or Power Pivot is used to connect to any of the following data sources and load data into the Excel data model. Power Query is *Get & Transform data* in Excel 2016.

## Power BI Gateway - Personal

- All online data sources shown in Power Query.
- All on-premises data sources shown in Power Query except for Hadoop file (HDFS) and Microsoft Exchange.
- All online data sources shown in Power Pivot.
- All on-premises data sources shown in Power Pivot except for Hadoop file (HDFS) and Microsoft Exchange.

## On-premises data gateway

On-premises data gateway supports refresh for the following data sources:

- Analysis Services Tabular
- Analysis Services Multidimensional
- SQL Server
- SAP HANA
- Oracle
- Teradata
- File
- Folder
- SharePoint list (on-premises)
- Web
- OData
- IBM DB2
- MySQL
- Sybase
- SAP BW
- IBM Informix Database
- ODBC

Keep the following notes in mind:

- A gateway must be installed and running in order for the Power BI service to connect to on-premises data sources and refresh the dataset.
- When using Excel 2013, make sure you've updated Power Query to the latest version.
- Refresh isn't supported for Excel workbooks imported from a local drive where data exists only in worksheets or linked tables. Refresh is supported for worksheet

data if it's stored and imported from OneDrive. To learn more, see [Refresh a dataset created from an Excel workbook on OneDrive, or SharePoint Online](#).

- When you refresh a dataset created from an Excel workbook imported from a local drive, only the data queried from data sources is refreshed.

If you change the structure of the data model in Excel or Power Pivot, for example, create a new measure or change the name of a column, those changes aren't copied to the dataset. If you make such changes, reupload or republish the workbook.

If you expect to make regular changes to the structure of your workbook and you want those changes to be reflected in the dataset in the Power BI service without having to reupload, consider putting your workbook on OneDrive. The Power BI service automatically refreshes both the structure and worksheet data from workbooks stored and imported from OneDrive.

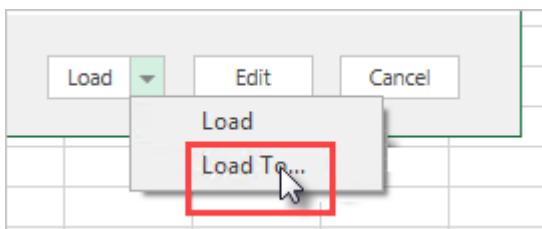
## How do I make sure data is loaded to the Excel data model?

When you use Power Query to connect to a data source, you have several options where to load the data. Power Query is *Get & Transform data* in Excel 2016. To make sure you load data into the data model, you must select the **Add this data to the Data Model** option in the **Load To** dialog.

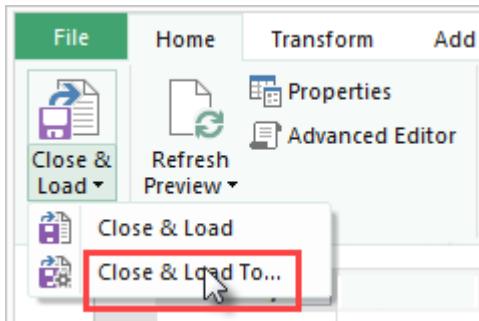
### ⓘ Note

The images here show Excel 2016.

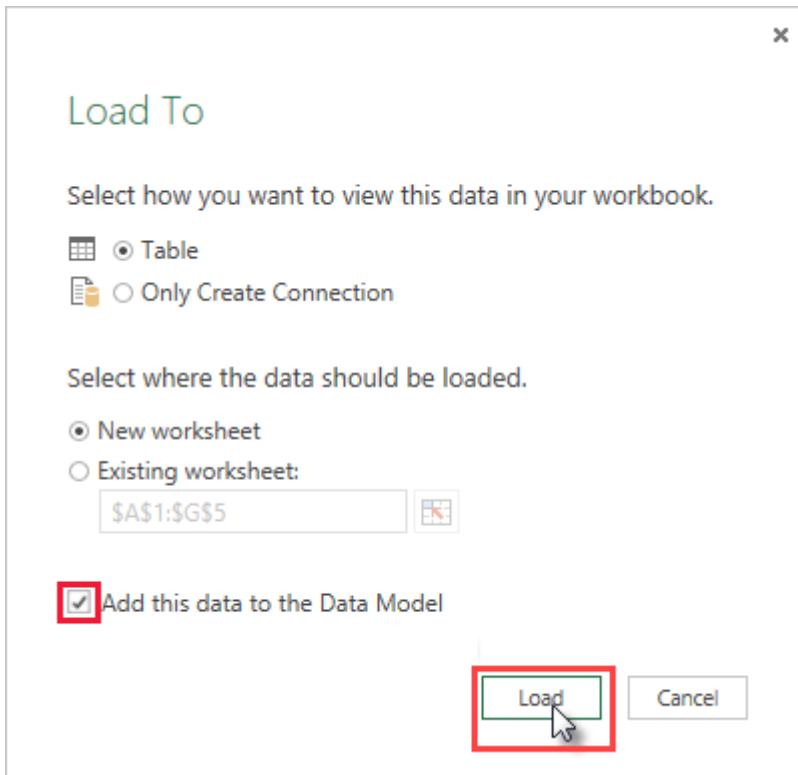
In Navigator, select **Load To...**



Or, if you select **Edit** in Navigator, you open the Query Editor. There you can select **Close & Load To....**



Then in Load To, make sure you select Add this data to the Data Model.



## What if I use Get External Data in Power Pivot?

No problem. Whenever you use Power Pivot to connect to and query data from an on-premises or online data source, the data is automatically loaded to the data model.

## How do I schedule refresh?

When you set up a refresh schedule, Power BI connects directly to the data sources using connection information and credentials in the dataset to query for updated data, then loads the updated data into the dataset. Any visualizations in reports and dashboards based on that dataset in the Power BI service are also updated.

For details on how to setup schedule refresh, see [Configure Schedule Refresh](#).

## When things go wrong

When things go wrong, it's usually because Power BI can't sign into data sources, or if the dataset connects to an on-premises data source, the gateway is offline. Make sure Power BI can sign into data sources. If a password you use to sign into a data source changes, or Power BI gets signed out from a data source, be sure to try signing into your data sources again in Data Source Credentials.

Be sure to leave the **Send refresh failure notification email to me** selected. You want to know right away if a scheduled refresh fails.

**ⓘ Important**

Refresh isn't supported for OData feeds connected to and queried from Power Pivot. When using an OData feed as a data source, use Power Query.

## Troubleshooting

Sometimes refreshing data might not go as expected. Typically problems are caused by an issue connected with a gateway. Take a look at the gateway troubleshooting articles for tools and known issues.

- [Troubleshooting the On-premises data gateway](#)
- [Troubleshooting the Power BI Gateway - Personal](#)

## Next steps

More questions? [Try the Power BI Community](#) ↗

# Refresh a dataset created from an Excel workbook on OneDrive or SharePoint Online

Article • 08/28/2023

## Important

The following capabilities are deprecated and will no longer be available starting September 29th, 2023:

- Upload of local workbooks to Power BI workspaces will no longer be allowed.
- Configuring scheduling of refresh and refresh now for Excel files that don't already have scheduled refresh configured will no longer be allowed.

The following capabilities are deprecated and will no longer be available starting October 31, 2023:

- Scheduled refresh and refresh now for existing Excel files that were previously configured for scheduled refresh will no longer be allowed.
- Local workbooks uploaded to Power BI workspaces will no longer open in Power BI.

After October 31, 2023:

- You can download existing local workbooks from your Power BI workspace.
- You can publish your Excel data model as a Power BI dataset and schedule refresh.
- You can import Excel workbooks from OneDrive and SharePoint Document libraries to view them in Power BI.

If your organization uses these capabilities, see more details in [Migrating your Excel workbooks](#).

You can import Excel workbooks from your local machine, or from cloud storage such as OneDrive for work or school or SharePoint Online. This article explores the advantages of using cloud storage for your Excel files. For more information about how to import Excel files into Power BI, see [Get data from Excel workbook files](#).

# What are the advantages?

When you import files from OneDrive, or SharePoint Online, it ensures the work you're doing in Excel stays in sync with the Power BI service. Any data that you've loaded into your file's model then updates in the dataset. Any reports you've created in the file load into Reports in Power BI. If you make and save changes to your file on OneDrive or SharePoint Online, Power BI shows the updates to those changes. For example, if you add new measures, change column names, or edit visualizations, Power BI reflects the changes. Your changes typically update within an hour after you've saved them.

When you import an Excel workbook from your personal OneDrive, any data in the workbook loads into a new dataset in Power BI. For example, tables in worksheets, data loaded into the Excel data model, and the structure of the data model goes into a new dataset. Power BI automatically connects to the workbook on OneDrive, or SharePoint Online, approximately every hour to check for updates. If the workbook changed, Power BI refreshes the dataset and reports in the Power BI service.

You can refresh the dataset in the Power BI service. When you manually refresh or schedule a refresh on the dataset, Power BI connects directly to the external data sources to query for any updated data. It then loads updated data into the dataset. Refreshing a dataset from within Power BI doesn't refresh the data in the workbook on OneDrive or SharePoint Online.

# What's supported?

Power BI supports the **Refresh Now** and **Schedule Refresh** options for datasets that meet the following conditions:

- The datasets are created from Power BI Desktop files that are imported from a local drive.
- Get data or Power Query Editor in Power BI is used to connect to and load the data.
- The data is from a source that's described in one of the following sections.

## Power BI gateway - personal

- All online data sources shown in Power BI Desktop's Get data and Power Query Editor.
- All on-premises data sources shown in Power BI Desktop's Get data and Power Query Editor except for Hadoop file (HDFS) and Microsoft Exchange.

# On-premises data gateway

On-premises data gateway supports refresh for the following data sources:

- Analysis Services Tabular
- Analysis Services Multidimensional
- SQL Server
- SAP HANA
- Oracle
- Teradata
- File
- Folder
- SharePoint list (on-premises)
- Web
- OData
- IBM DB2
- MySQL
- Sybase
- SAP BW
- IBM Informix Database
- ODBC

## ⓘ Note

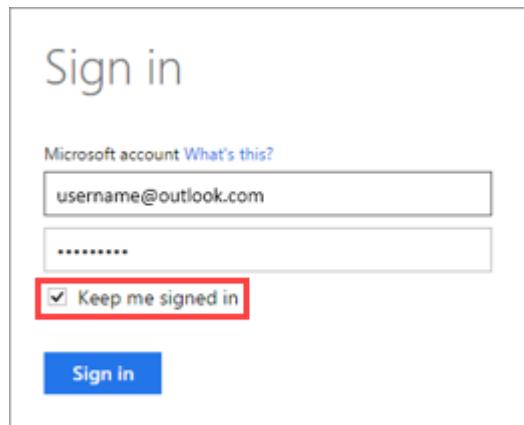
A gateway must be installed and running in order for Power BI to connect to on-premises data sources and refresh the dataset.

# OneDrive or OneDrive for work or school. What's the difference?

If you have both a personal OneDrive and OneDrive for work or school, it's recommended you keep files you want to import in OneDrive for work or school. Here's why: You likely use two different accounts to sign in and access your files.

In Power BI, connecting to OneDrive for work or school is typically seamless because you likely use the same account to sign in to Power BI as OneDrive for work or school. But with personal OneDrive, it's more common to sign in with a different [Microsoft account](#).

When you sign in to OneDrive for work or school with your Microsoft account, select **Keep me signed in**. Power BI can then synchronize any updates you make in the file in Power BI Desktop with datasets in Power BI.



If your Microsoft account credentials change, edits to your file on OneDrive can't synchronize with the dataset or reports in Power BI. You need to reconnect and import the file again from your personal OneDrive.

## Options for connecting to an Excel file

When you connect to an Excel workbook in OneDrive for work or school, or SharePoint Online, you have two options on how to get what's in your workbook into Power BI.

**Import Excel data into Power BI** – When you import an Excel workbook from your OneDrive for work or school or SharePoint Online, it works as described previously.

**Connect, manage, and view Excel in Power BI** – When using this option, you create a connection from Power BI right to your workbook on OneDrive for work or school or SharePoint Online.

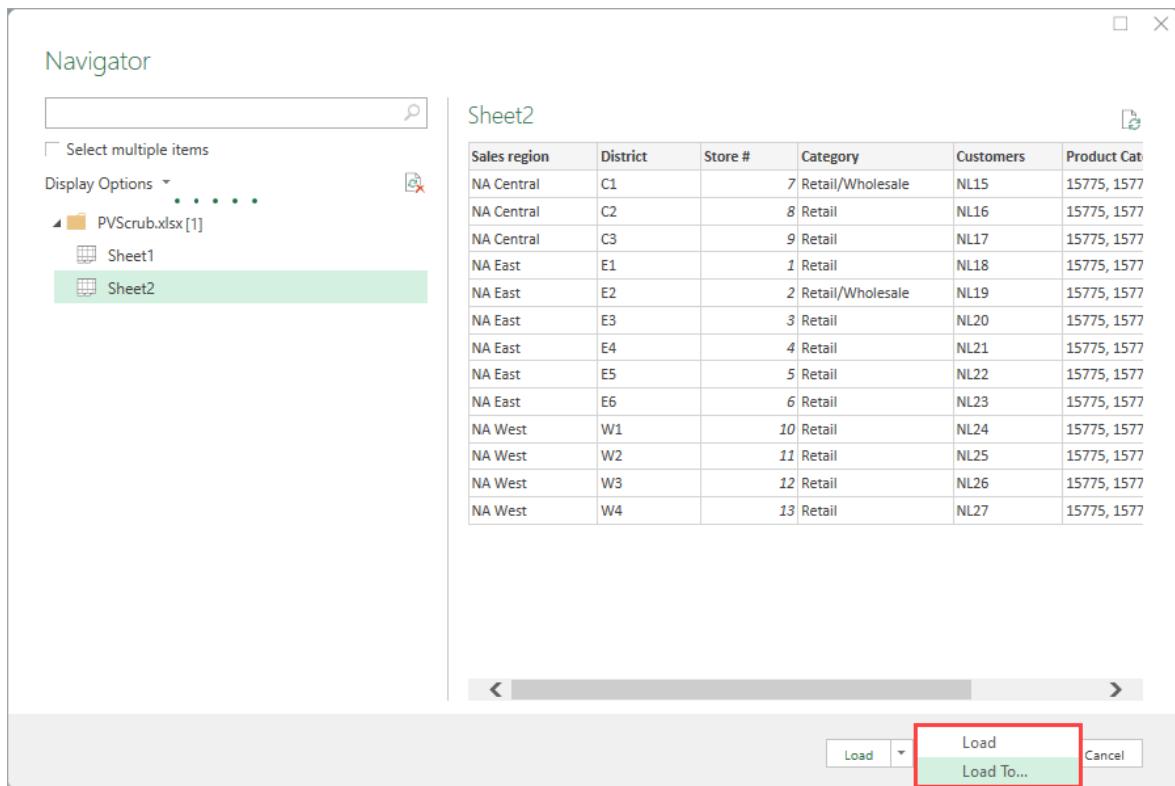
When you connect to an Excel workbook this way, a dataset isn't created in Power BI. But the workbook appears in the Power BI service under Reports with an Excel icon next to the name. Unlike with Excel Online, when you connect to your workbook from Power BI, if your workbook has connections to external data sources that load data into the Excel data model, you can set up a refresh schedule.

When you set up a refresh schedule this way, the only difference is refreshed data goes into the workbook's data model on OneDrive, or SharePoint Online, rather than a dataset in Power BI.

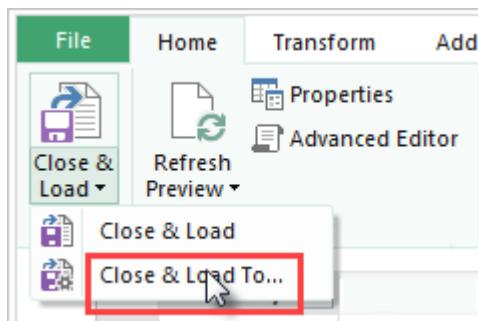
## How do I make sure data is loaded to the Excel data model?

When you use Power Query (Get & Transform Data in Excel 2016) to connect to a data source, you have several options of where to load the data. To ensure that you load data into the data model, you must select the **Add this data to the Data Model** option in the Import Data dialog box.

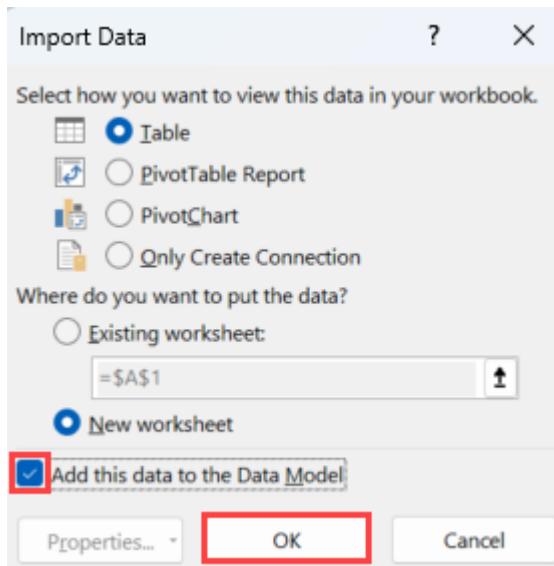
1. In Excel, select **Data > Get Data** and select where you want your data to come from. In this example, the data loads from an Excel workbook file.
2. In the file browser window, locate and select your data file and then select **Import**.
3. In **Navigator**, select your file and choose **Load To...**



Or, in Excel, select **Data > Get Data > Launch Power Query Editor** to open the Query Editor. There you can select **Close & Load To....**



4. Then, in **Import Data**, be sure to select **Add this data to the Data Model** and select **OK**.



## What if I use Get External Data in Power Pivot?

No problem. Whenever you use Power Pivot to connect to and query data from an on-premises or online data source, the data automatically loads to the data model.

## How do I schedule a refresh?

When you set up a refresh schedule, Power BI connects directly to the data sources by using connection information and credentials in the dataset to query for updated data. It then loads the updated data into the dataset. Any visualizations in reports and dashboards based on that dataset in the Power BI service also update.

For more information about how to set up a scheduled refresh, see [Configure scheduled refresh](#).

## When things go wrong

When things go wrong, it's usually because Power BI can't sign in to data sources. Or it's because the dataset connects to an on-premises data source and the gateway is offline. Be sure Power BI can sign in to data sources. If a password you use to sign in to a data source changes, or Power BI is signed out from a data source, be sure to sign in to your data sources again in Data Source Credentials.

Be sure to leave the **Send refresh failure notification email to me** setting selected. You want to know right away if a scheduled refresh fails.

## Important notes

Refresh isn't supported for OData feeds connected to and queried from Power Pivot.  
When using an OData feed as a data source, use Power Query.

## Troubleshooting

Sometimes refreshing data might not go as expected. Typically, problems with refreshing are an issue with the data gateway. For tools, tips, and known issues, see the following articles about troubleshooting the gateway.

- [Troubleshoot the on-premises data gateway](#)
- [Troubleshoot the Power BI gateway - personal](#)

More questions? [Try the Power BI Community](#).

# Refresh a dataset created from a .CSV file on OneDrive or SharePoint

Article • 06/06/2023

When you connect to a comma separated value (.csv) file on OneDrive or SharePoint, a dataset is created in Power BI. Data from the .csv file is imported into the dataset in Power BI. Power BI then automatically connects to the file and refreshes any changes with the dataset in Power BI. If you edit the .csv file in OneDrive, or SharePoint, after you save, those changes will appear in Power BI, usually within about an hour. Any visualizations in Power BI based on the dataset are automatically updated.

## ⓘ Note

By default, using the **Get Data** experience for specific file type connectors in Power BI Desktop uses a local reference to the file stored on OneDrive, which will not automatically update unless you have a gateway configured. To have your CSV automatically update without having to configure a gateway, rather than using the **Text/CSV** connector, use the **Web** connector and reference the online version of your CSV.

## Advantages

If your files are in a shared folder on OneDrive for work or school, or SharePoint, other users can work on the same file. After they save the file, changes are automatically updated in Power BI, usually within an hour.

Many organizations run processes that automatically query databases for data that's saved to a .csv file each day. If the file is stored on OneDrive, or SharePoint, and the same file is overwritten each day, as opposed to a new file with a different name being created each day, you can connect to that file in Power BI. Your dataset that connects to the file will be synchronized soon after the file on OneDrive, or SharePoint, is updated. Any visualizations based on the dataset are automatically updated.

## What's supported

Comma separated value files are simple text files, connections to external data sources and reports aren't supported. You can't schedule a refresh on a dataset created from a

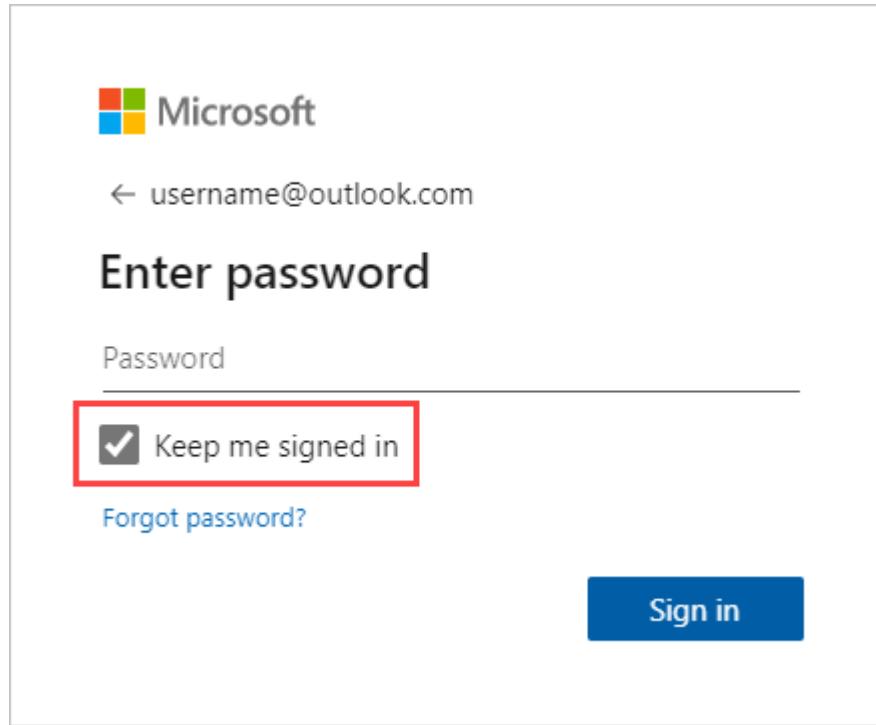
.csv file. However, when the file is on OneDrive, or SharePoint, Power BI will automatically synchronize any changes to the file with the dataset about every hour.

## What's the difference between personal OneDrive and OneDrive for work or school

If you have both a personal OneDrive and OneDrive for work or school, we recommended you keep any files you want to connect to Power BI on OneDrive for work or school. Why? Because you likely use two different accounts to sign into them.

Connecting to OneDrive for work or school and Power BI is typically seamless because the same account you use to sign into Power BI is often the same account used to sign into OneDrive for work or school. But, with personal OneDrive, you're likely to use a different [Microsoft account ↗](#).

When you sign into your Microsoft account, be sure to select **Keep me signed in**. Power BI can then synchronize any updates with datasets in Power BI.



When you make changes to your .csv file on OneDrive, and they don't synchronize with the dataset in Power BI. It might be because your Microsoft account credentials changed, you'll need to connect to the file and import it again from your personal OneDrive.

## When things go wrong

If data in the .csv file on OneDrive changes, and the changes aren't reflected in Power BI, it's most likely because Power BI can't connect to OneDrive. Try connecting to the file and importing it again. If you're prompted to sign in, make sure you select **Keep me signed in**.

## Next steps

- [Troubleshoot gateways - Power BI](#)
- [Troubleshoot refresh scenarios](#)

More questions? [Ask the Power BI Community](#) ↗

# Query caching in Power BI Premium or Power BI Embedded

Article • 01/09/2023

Organizations with Power BI Premium or Power BI Embedded can take advantage of *query caching* to speed up reports associated with a dataset. Query caching instructs the Power BI Premium or Power BI Embedded capacity to use its local caching service to maintain query results, avoiding having the underlying data source compute those results.

## Important

Query caching is only available on Power BI Premium or Power BI Embedded, for Import datasets. It is not applicable DirectQuery or LiveConnect datasets that use Azure Analysis Services or SQL Server Analysis Services.

The caching is performed the first time a user opens the report. At present, the service only does query caching for the initial page that they land on. In other words, queries aren't cached when you interact with the report. Cached query results are specific to user and dataset context and always respect security rules. The query cache respects [personal bookmarks](#) and [persistent filters](#), so queries generated by a personalized report are cached. [Dashboard tiles](#) that are powered by the same queries also benefit once the query is cached. Performance especially benefits when a dataset is accessed frequently and doesn't need to be refreshed often. Query caching can also reduce load on your capacity by reducing the overall number of queries.

You control query caching behavior on the [Settings](#) page for the dataset in the Power BI service. It has three possible settings:

- **Capacity default:** Query caching Off
- **Off:** Don't use query caching for this dataset.
- **On:** Use query caching for this dataset.

#### ▲ Query Caching

- Capacity default: query caching Off
- Off: Do not cache query results for this dataset
- On: Speed up reports by using previously saved query results. [Learn more](#)

Apply

Discard

## Considerations and limitations

- When you change caching settings from **On** to **Off**, all previously saved query results for the dataset are removed from the capacity cache. You can turn off caching either explicitly or by reverting to capacity default setting that an administrator has set to **Off**. Turning it off can introduce a small delay the next time any report runs queries against this dataset. The delay is caused by those report queries running on demand and not applying saved results. Also, the required dataset might need to be loaded into memory before it can service queries.
- The query cache is refreshed when Power BI performs a dataset refresh. When the query cache is refreshed, Power BI must run queries against the underlying data models to get the latest results. If a large number of datasets have query caching enabled and the Premium/Embedded capacity is under heavy load, some performance degradation might occur during cache refresh. Degradation results from the increased volume of queries being executed.

## Next steps

[What is Power BI embedded analytics?](#)

# What are Power BI template apps?

Article • 01/09/2023

The new Power BI *template apps* enable Power BI partners to build Power BI apps with little or no coding, and deploy them to any Power BI customer. This article is an overview of the Power BI template app program.

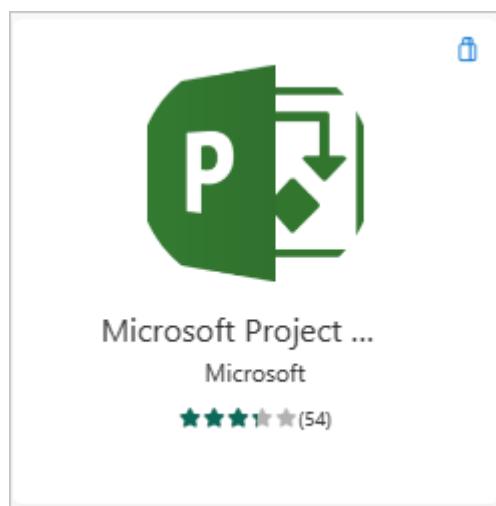
As a Power BI partner, you create a set of out-of-the-box content for your customers and publish it yourself.

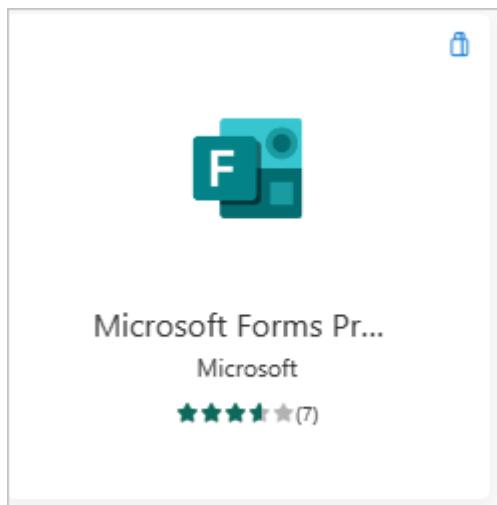
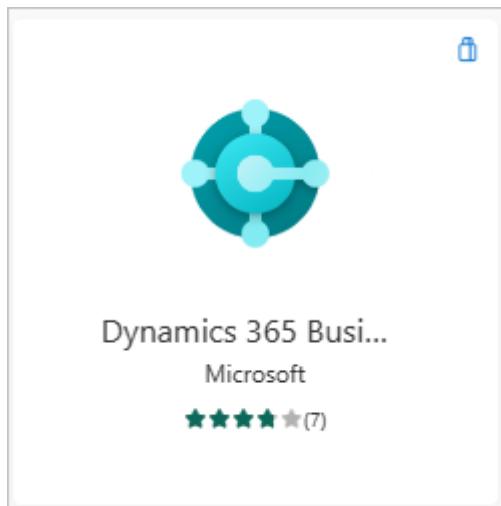
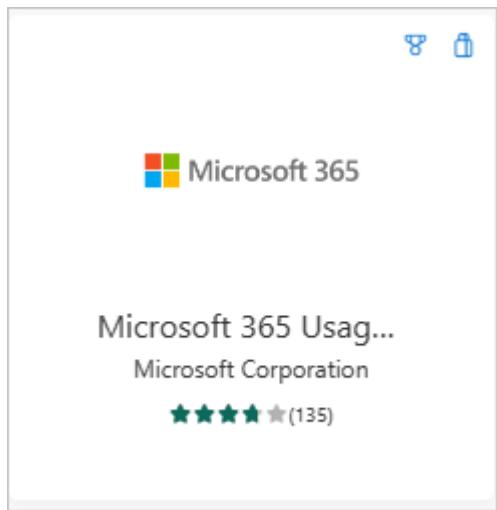
You build template apps that allow your customers to connect and instantiate within their own accounts. As domain experts, they can unlock the data in a way that's easy for their business users to consume.

You submit a template app to the Partner center. The apps then become publicly available in the [Power BI apps](#) marketplace and on [the Microsoft commercial marketplace](#). Here's a high-level look at the public template app creation experience.

## Power BI Apps marketplace

Power BI template apps allow Power BI Pro or Power BI Premium users to gain immediate insights through prepackaged dashboards and reports that can be connected to live data sources. Many Power BI Apps are already available in the [Power BI Apps](#) marketplace.





### ⓘ Note

Marketplace apps aren't available for US government cloud instances. For more information, see [Power BI for US government customers](#).

## Process

The general process to develop and submit a template app involves several stages. Some stages can include more than one activity at the same time.

Stage	Power BI Desktop	Power BI service	Partner Center
One	Build a data model and report in a .pbix file	Create a workspace. Import .pbix file. Create a complementary dashboard	Register as a partner
Two		Create a test package and run internal validation	
Three		Promote the test package to preproduction for validation outside your Power BI tenant, and submit it to AppSource	With your preproduction package, create a Power BI template app offer and start the validation process
Four		Promote the preproduction package to production	Go live

## Before you begin

To create the template app, you need permissions to create one. For more information, see [Template app tenant settings](#).

To publish a template app to the Power BI service and AppSource, you must meet the requirements for [becoming a Partner Center publisher](#).

## High-level steps

Here are the high-level steps.

1. [Review the requirements](#) to make sure you meet them.
2. Build a report in Power BI Desktop. Use parameters so you can save it as a file other people can use.
3. Create a workspace for your template app in your tenant on the Power BI service ([app.powerbi.com](http://app.powerbi.com)).
4. Import your .pbix file and add content such as a dashboard to your app.
5. Create a test package to test the template app yourself within your organization.

6. Promote the test app to pre-production to submit the app for validation in AppSource, and to test outside your own tenant.
7. Submit the content to [Partner center](#) for publishing.
8. Make your offer go *Live* in AppSource, and move your app to production in Power BI.
9. Now you can start developing the next version in the same workspace, in preproduction.

## Requirements

To create the template app, you need permissions to create one. For more information, see [Template app tenant settings](#).

To publish a template app to the Power BI service and AppSource, you must meet the requirements for [becoming a Partner Center publisher](#).

### Note

Template apps submissions are managed in [Partner Center](#). Use the same Microsoft Developer Center registration account to sign in. You should have only one Microsoft account for your AppSource offerings. Accounts shouldn't be specific to individual services or offers.

## Tips

- Make sure your app includes sample data to get everyone started in a click.
- Limit dataset size (rule of thumb: *.pbix* file < 10MBs). This typically means keeping the size of sample data as small as possible.
- Carefully examine your application by installing it in your tenant and in a secondary tenant. Make sure customers only see what you want them to see.
- Use AppSource as your online store to host your application. This way everyone using Power BI can find your app.
- Consider offering more than one template app for separate unique scenarios.
- Enable data customization. For example, support custom connection and parameters configuration by the installer.
- If you're an independent software vendor and are distributing your app through your web service, consider automating parameter configuration during installation to make things easier for your customers and to increase the likelihood of a

successful installation. For more information, see [Automated configuration of a template app installation](#).

See [Tips for authoring template apps in Power BI](#) for more suggestions.

## Known limitations

Feature	Known Limitation
Contents: Datasets	Exactly one dataset should be present. Only datasets built into Power BI Desktop (.pbix files) are allowed. Not supported: Datasets from other template apps, cross-workspace datasets, paginated reports (.rdl files), and Excel workbooks.
Contents: Reports	A single template app can't include more than 20 reports.
Contents: Dashboards	Real-time tiles aren't allowed. In other words, no support for push or streaming datasets.
Contents: Dataflows	Not supported: Dataflows.
Contents from files	Only .pbix files are allowed. Not supported: .rdl files (paginated reports) and Excel workbooks.
Data sources	Data sources supported for cloud Scheduled Data refresh are allowed. Not supported: Live connections, on-premises data sources (personal and enterprise gateways aren't supported), real time (no support for push dataset), and composite models.
Dataset: cross-workspace	No cross-workspace datasets are allowed.
Query parameters	Not supported: Parameters of type <i>Any</i> , <i>Date</i> , or <i>Binary</i> type block refresh operation for dataset.
Incremental refresh	Template apps don't support incremental refresh.
Power BI visuals	Only publicly available Power BI visuals are supported. <a href="#">Organizational Power BI visuals</a> aren't supported.
Sovereign clouds	Template apps aren't available in sovereign clouds.
Composite models	Composite models shouldn't be used in the app builder workspace. App installers can use composite models after installing the app.

Feature	Known Limitation
Large dataset storage format	Large dataset storage format isn't supported for template apps.

## Support

For support during development, use <https://powerbi.microsoft.com/support>. We actively monitor and manage this site. Customer incidents quickly find their way to the appropriate team.

## Next steps

- [Create a template app](#)

# Create a template app in Power BI

Article • 03/21/2023

This article contains step-by-step instructions for creating a Power BI *template app*.

Power BI template apps let Power BI partners build Power BI apps with little or no coding, and deploy them to any Power BI customer.

If you can create Power BI reports and dashboards, you can become a *template app builder* and build and package analytical content into an app. You can then deploy your app to other Power BI tenants through any available platform, such as AppSource or your own web service. If you're distributing your template app through your own web service, you can [automate part of the installation process](#) to make things easier for your customers.

Power BI admins govern and control who in their organization can create template apps, and who can install them. Authorized users can install your template app, modify it, and distribute it to the Power BI consumers in their organizations.

## Prerequisites

Here are the requirements for building a template app:

- A [Power BI pro license](#)
- [Power BI Desktop](#) (optional)
- Familiarity with [basic Power BI concepts](#)
- Permissions to share a template app publicly as shown in [Template app tenant settings](#)

## Create the template workspace

To create a template app you can distribute to other Power BI tenants, you need to create it in a workspace.

1. In the Power BI service, create a workspace as described in [Create a workspace in Power BI](#). In the **Advanced** section, select **Develop a template app**.

Develop a template app

Template apps are developed for sharing outside your organization.  
A template app workspace will be created for developing and releasing the app. [Learn more](#)

Security settings

Allow contributors to update the app for this workspace

**Save** **Cancel**

**ⓘ Important**

The capacity that the app builder workspace is assigned to does not determine the capacity assignment of workspaces where app installers install the app. This means that an app developed in a premium capacity workspace will not necessarily be installed on a premium capacity workspace. Therefore it is **not** recommended to use premium capacity for the builder workspace, as installer workspaces might not be premium capacity, and functionality that relies on premium capacity won't work unless the installer manually reassigns the installed workspace to premium capacity.

2. When you're done creating the workspace, select **Save**.

**ⓘ Note**

You need permissions from your Power BI admin to promote template apps.

## Add content to the template app workspace

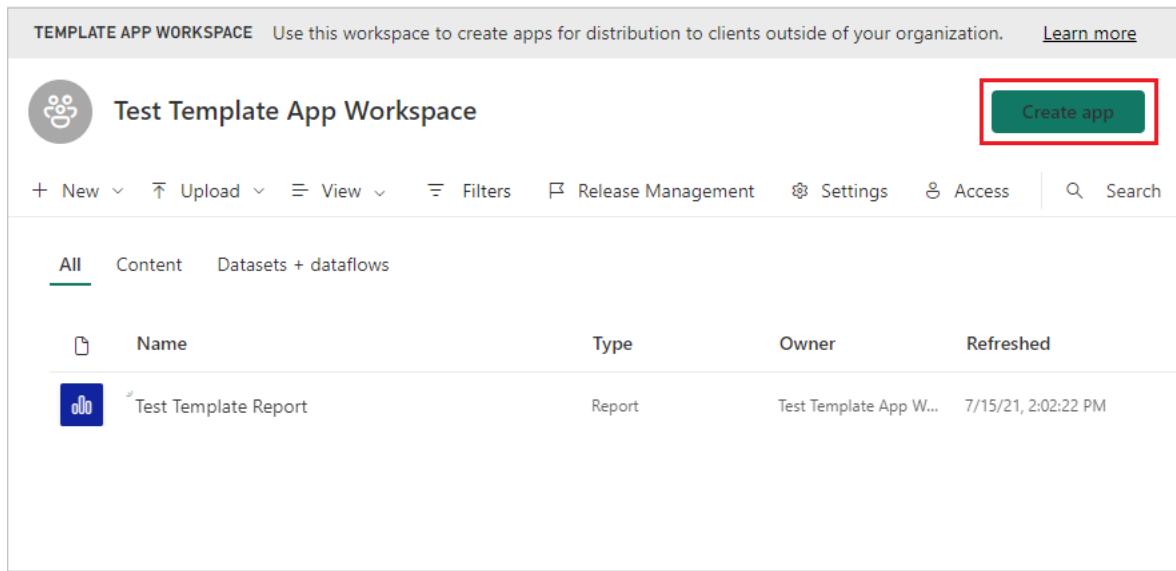
As with a regular Power BI workspace, your next step is to add content to the workspace. If you're using parameters in Power Query, make sure they have well-defined types, such as `Text`. The types `Any` and `Binary` aren't supported.

For suggestions to consider when creating reports and dashboards for your template app, see [Tips for authoring template apps in Power BI](#).

## Define the properties of the template app

Now that you have content in your workspace, you can package it in a template app. The first step is to create a test template app, accessible only from within your organization on your tenant.

1. In the template app workspace, select **Create app**.



The screenshot shows a Microsoft Power BI workspace titled "Test Template App Workspace". At the top, there's a banner that says "TEMPLATE APP WORKSPACE Use this workspace to create apps for distribution to clients outside of your organization." with a "Learn more" link. Below the banner, there's a navigation bar with icons for New, Upload, View, Filters, Release Management, Settings, Access, and Search. A green "Create app" button is located in the top right corner of the workspace area, which is highlighted with a red rectangular box. The main content area displays a table with three columns: "All", "Content", and "Datasets + dataflows". The "All" tab is selected. The table lists one item: "Test Template Report" (Type: Report, Owner: Test Template App W..., Refreshed: 7/15/21, 2:02:22 PM).

All	Content	Datasets + dataflows
 Test Template Report	Type: Report	Owner: Test Template App W... Refreshed: 7/15/21, 2:02:22 PM

Next, fill in more options for your template app in six tabs.

2. On the **Branding** tab, complete the following fields:

- **App name**
- **Description**
- **Support site**. The support link appears under app info after you redistribute the template app as an organizational app.
- **App logo**. The logo has a 45K file-size limit, must have a 1:1 aspect ratio, and must be in a *.png*, *.jpg*, or *.jpeg* file format.
- **App theme color**

## Test Template App Workspace

Branding Navigation Control Parameters Authentication Access

X

### Build your app

App name \*

Test Template App Workspace

Description \*

Enter a summary

Describe your app.

200 characters left

Support site

Share where your users can find help

App logo



Upload

Delete

App theme color



Create app

Save changes

Close

3. On the **Navigation** tab, you can turn on **New navigation builder** to define the navigation pane of the app.

## Test Template App Workspace My first app.

Branding Navigation Control Parameters Authentication Access

X

New navigation builder

On

Add reports and dashboards to this app. Then organize the custom navigation pane so it's easy for people to find what they're looking for.

Navigation \*

+ New

Test Template Report

Report details

Name \*

Test Template Report

Reset

Report link

<https://msit.powerbi.com/groups/872e9962-9b9f-48b7-a96e-e1e723e63590/reports/7bb0f2e6-39b6->

Open

Section

No section

Hide from navigation

> Advanced

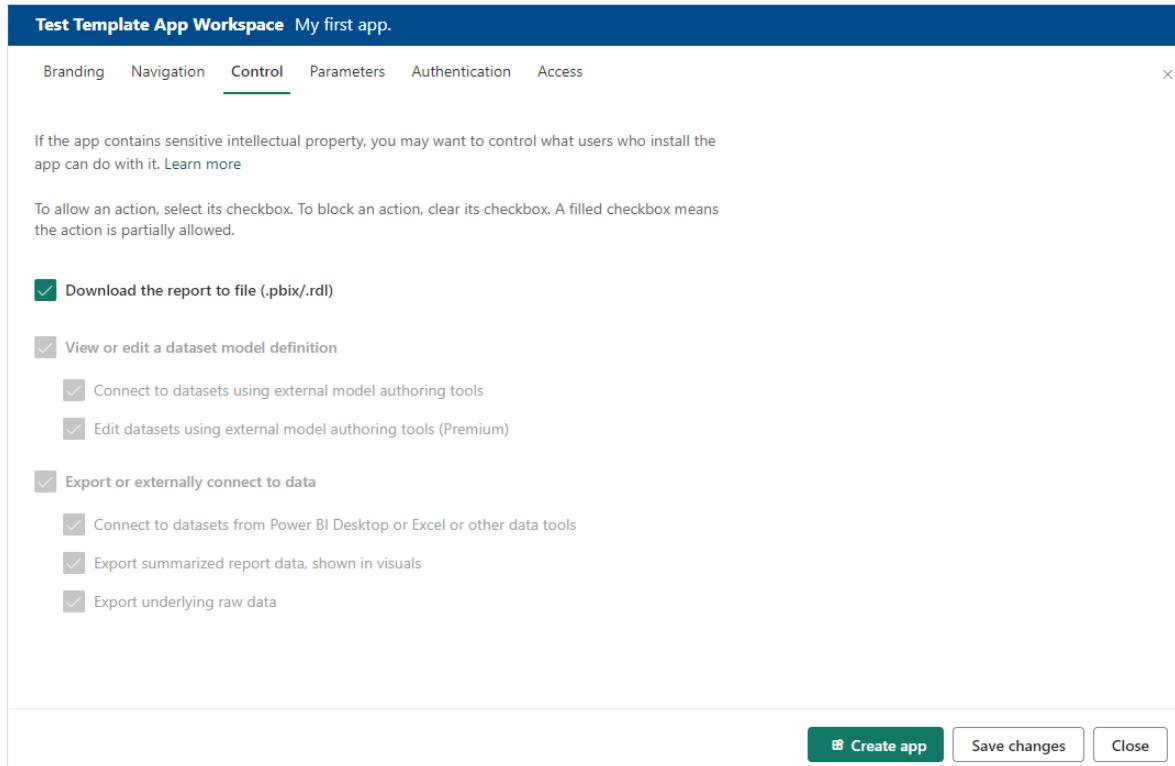
Create app

Save changes

Close

If you don't turn on **New navigation builder**, you have the option of selecting an app landing page. Define a report or dashboard to be the landing page of your app. Use a landing page that gives the impression you want.

4. On the **Control** tab, set your app users' limits and restrictions on your app's content. You can use this control to protect intellectual property in your app.



### ⚠ Note

If you want to protect your data, disable the **Download the report to file** option and then configure the other two options as desired.

Why:

The view, edit, and export controls on this tab apply only to the Power BI service. Once you download the *.pbix* file, it is no longer in the service. It puts a copy of your data, unprotected, in a location chosen by the user. You then no longer have any control over what the user can do with it.

If you want to limit access to your queries and measures while still allowing your users to add their own data sources, consider checking only the **Export or externally connect to data** options. This enables users to add their own data sources without being able to edit your dataset. For more information, see [Use composite models in Power BI Desktop](#).

5. Parameters are created in the original *.pbix* file (learn more about [creating query parameters ↗](#)). You use the capabilities on this tab to help the app installer configure the app after installation when they connect to their data.

**Test Template App Workspace** My first app.

Branding Navigation Control **Parameters** Authentication Access

Manage Parameters

View and modify dataset parameters for this template app. These settings determine the default parameters installed with the app and if the value is required when the app is installed.

**Product**

**Value \***  
Bike

Required

**Color**

**Value \***  
Red

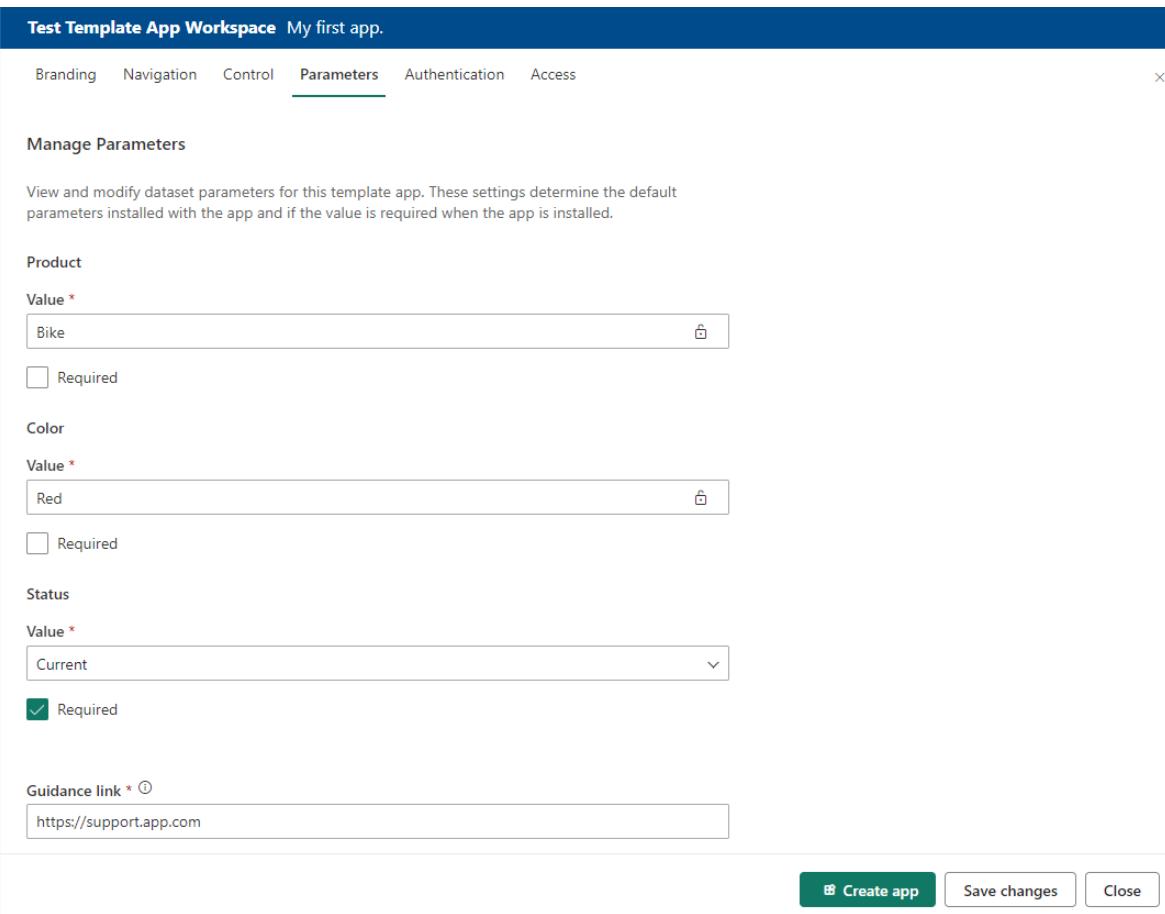
Required

**Status**

**Value \***  
Current

Required

**Guidance link \* ⓘ**  
<https://support.app.com>



Each parameter has a name, which comes from the query, and a **Value** field. There are three options for getting a value for the parameter during installation:

- You can require the user who installs the app to enter a value.

In this case, you provide an example that the user replaces. To configure a parameter in this way, select the **Required** checkbox, and then give an example in the textbox that shows the user what kind of value is expected, as shown in the following example.

**Color**

**Value \***  
Black

**Provide an example of user input \* ⓘ**  
Red

Required



- You can provide a pre-populated value that the user who installs the app can't change.

A parameter configured in this way is hidden from the user who installs the app. You should use this method only if you're sure that the pre-populated

value is valid for all users. If not, use the first method that requires user input.

To configure a parameter in this way, enter the value in the **Value** textbox, and then select the lock icon so the value can't be changed. The following example shows this option:

The screenshot shows a configuration interface with a title 'Product'. Below it is a section labeled 'Value \*' containing a text input field with the value 'Bike'. To the right of the input field is a small square icon containing a padlock symbol, indicating that the value is locked and cannot be changed. Below the input field is a checkbox labeled 'Required'.

- You can provide a default value that the user can change during installation.

To configure a parameter in this way, enter the desired default value in the **Value** textbox, and leave the lock icon unlocked, as in the following example:

The screenshot shows a configuration interface with a title 'Product'. Below it is a section labeled 'Value \*' containing a text input field with the value 'Bike'. To the right of the input field is an empty square icon, indicating that the value is not locked. Below the input field is a checkbox labeled 'Required'.

In this tab, you also provide a link to the app documentation.

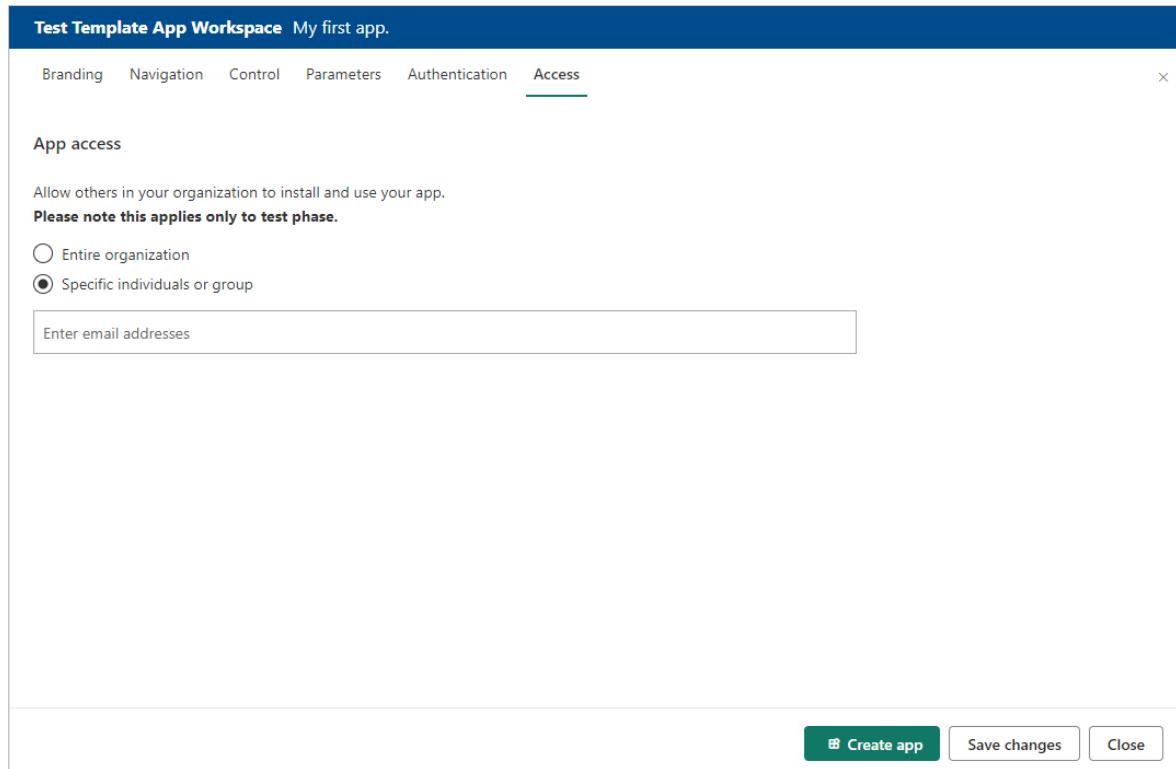
6. On the **Authentication** tab, select the authentication method to use. The available options depend on the data source types being used.

The screenshot shows the 'Test Template App Workspace' interface with the title 'My first app'. The top navigation bar includes tabs for 'Branding', 'Navigation', 'Control', 'Parameters', 'Authentication' (which is underlined, indicating it is the active tab), and 'Access'. Below the tabs, the 'App authentication' section is visible, which says: 'Choose the default authentication method for each type of data source. This is the method that will be required for connecting the app to data.' It lists a data source named 'SQL Server' (AdventureWorksDW2017-LAPTOP-RS) and provides a dropdown menu for selecting the authentication method. At the bottom of the screen are buttons for 'Create app', 'Save changes', and 'Close'.

Privacy level is configured automatically:

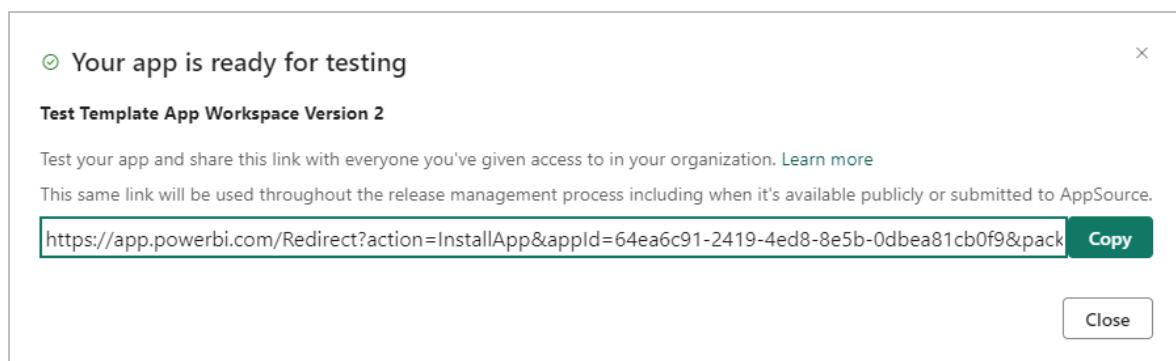
- A single datasource is automatically configured as private.
- A multi anonymous datasource is automatically configured as public.

7. In the test phase, on the **Access** tab decide who else in your organization can install and test your app. You will come back and change these settings later. The setting doesn't affect access of the distributed template app.



8. Select **Create app**.

You see a message that the test app is ready, with a link to copy and share with your app testers.



You've also done the first step of the following release management process.

## Manage the template app release

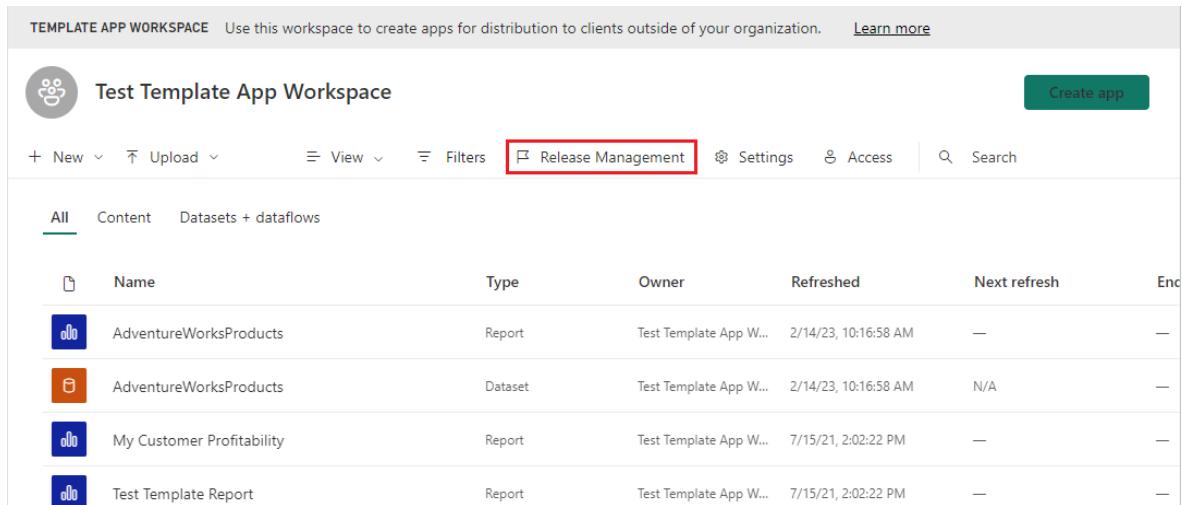
Before you release the template app publicly, you want to make sure it's ready. In the Power BI release management pane, you can follow and inspect the full app release path. You can also trigger the transition from stage to stage. The common stages are:

- Generate a test app for testing within your organization only.
- Promote the test package to pre-production stage and test outside of your organization.
- Promote the pre-production package to the production version in Production.
- Delete any package or start over from a previous stage.

The URL doesn't change as you move between release stages. Promotion doesn't affect the URL itself.

To go through the release stages:

1. In the template workspace, select **Release Management**.



The screenshot shows the 'Test Template App Workspace' interface. At the top, there is a header bar with the title 'TEST TEMPLATE APP WORKSPACE' and a sub-instruction 'Use this workspace to create apps for distribution to clients outside of your organization.' followed by a 'Learn more' link. Below the header is a navigation bar with several items: '+ New', 'Upload', 'View', 'Filters', 'Release Management' (which is highlighted with a red box), 'Settings', 'Access', and 'Search'. Underneath the navigation bar, there are three tabs: 'All' (which is underlined and highlighted with a green box), 'Content', and 'Datasets + dataflows'. The main area displays a table with four rows of data. Each row contains a small icon, the name of the item, its type, owner, refresh status, next refresh date, and end date. The data is as follows:

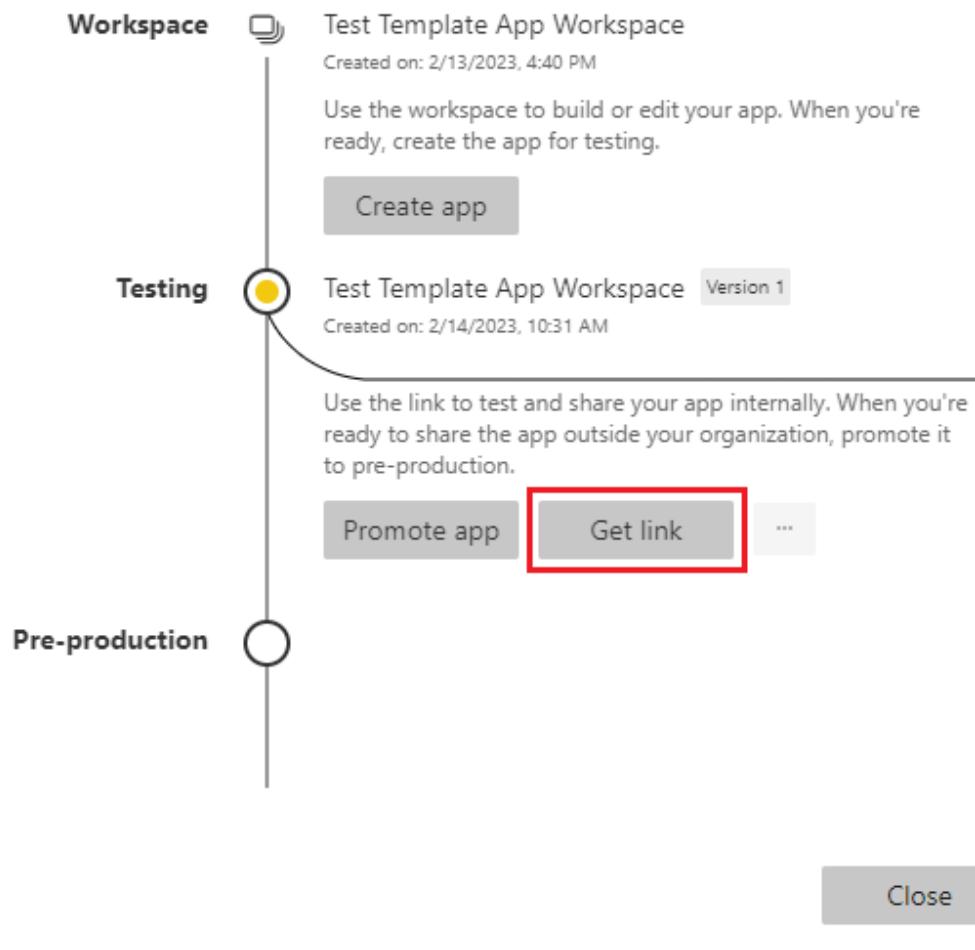
	Name	Type	Owner	Refreshed	Next refresh	End
	AdventureWorksProducts	Report	Test Template App W...	2/14/23, 10:16:58 AM	—	—
	AdventureWorksProducts	Dataset	Test Template App W...	2/14/23, 10:16:58 AM	N/A	—
	My Customer Profitability	Report	Test Template App W...	7/15/21, 2:02:22 PM	—	—
	Test Template Report	Report	Test Template App W...	7/15/21, 2:02:22 PM	—	—

2. If you followed the steps in this article to create the test app, the dot next to **Testing** will already be filled in. Select **Get link**.

If you haven't created the app yet, select **Create app** to start the template app creation process.

# Release Management

Track the progress of your template app development lifecycle, and prepare it for release outside your organization. [Learn more](#)



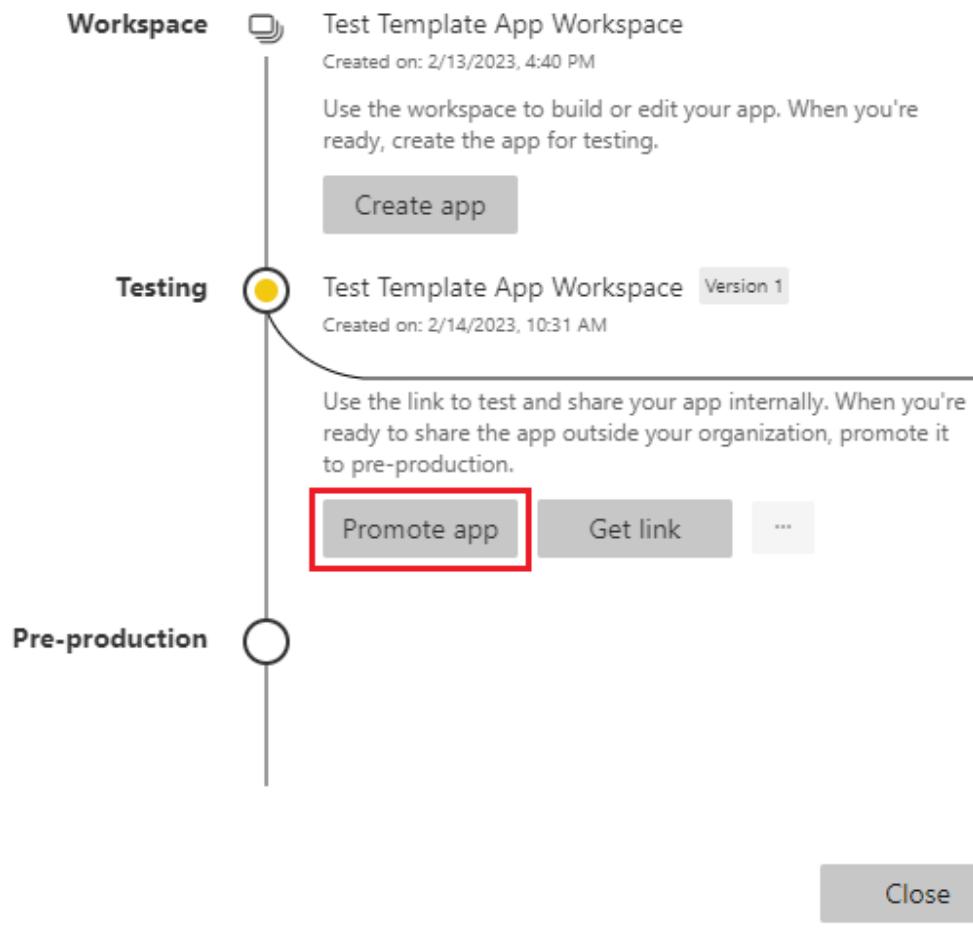
3. To test the app installation experience, copy the link in the window and paste it into a new browser window.

From here, you follow the same procedure your app installers will follow. For more information, see [Install and distribute template apps in your organization](#).

4. In the dialog box, select **Install**.
5. After installation succeeds, select the app in the **Apps** list to open it.
6. Verify that the test app has the sample data. To make any changes, go back to the app in the original workspace. Update the test app until you're satisfied.
7. When you're ready to promote your app to pre-production for testing outside your tenant, go back to the **Release Management** pane and select **Promote app**.

# Release Management

Track the progress of your template app development lifecycle, and prepare it for release outside your organization. [Learn more](#)



## ⓘ Note

When you promote the app, it becomes publicly available outside your organization.

If you don't see the **Promote app** option, contact your Power BI admin to grant you [permissions for template app development](#) in the admin portal.

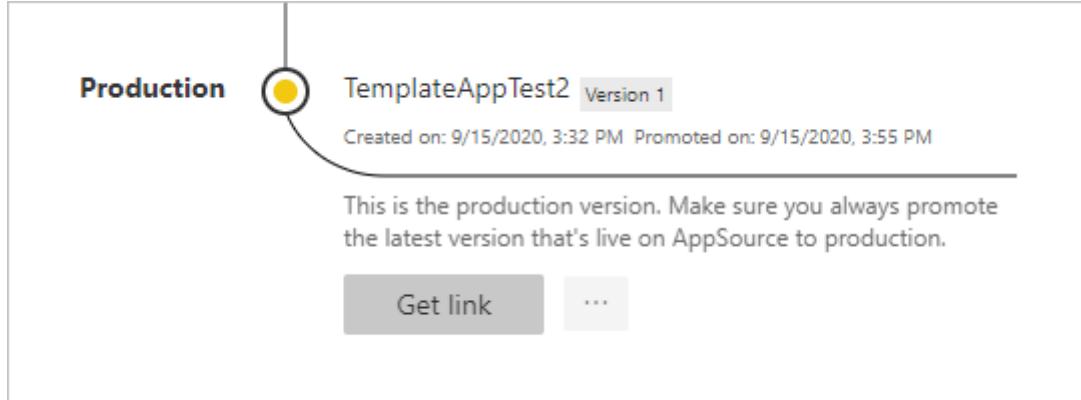
8. In the dialog box, select **Promote**.
9. Copy the new URL to share outside your tenant for testing. This link is also the one you submit to begin the process of distributing your app on AppSource by creating a [new Partner center offer](#).

Submit only pre-production links to the Partner center. After the app is approved and you get notification that it's published in AppSource, you can promote the package to production in Power BI.

10. When your app is ready for production or sharing via AppSource, go back to the **Release Management** pane and select **Promote app** next to **Pre-production**.

11. Select **Promote**.

Now your app is in production and ready for distribution.



To make your app widely available to Power BI users throughout the world, submit it to AppSource. For more information, see the [Create a Power BI app offer](#).

## Automate parameter configuration during installation

If you're an independent software vendor and distribute your template app via your web service, you can create automation that configures template app parameters automatically when your customers install the app in Power BI. Automatic configuration makes things easier for your customers and increases the likelihood of a successful installation, because customers don't have to supply details that they might not know. For more information, see [Automated configuration of a template app installation](#).

## Next steps

- To learn how your customers interact with your template app, see [Install, customize, and distribute template apps in your organization](#).
- For details on distributing your app, see the [Create a Power BI app offer](#).

# Tips for authoring template apps in Power BI

Article • 02/06/2023

When you're [authoring your template app](#) in Power BI, part of the process is the logistics of creating the workspace, testing it, and production. But the other important part is obviously authoring the report and the dashboard. You can break down the authoring process into several components. Working on these components helps you create the best possible template app:

- **Queries.** With queries, you [connect](#) and [transform](#) the data, and define [parameters ↗](#).
- **Data model.** In the data model, you create [relationships](#), [measures](#), and Q&A improvements.
- **Report pages.** [Report pages](#) include visuals and filters to provide insights into your data.
- **Dashboard and tiles.** [Dashboards](#) and [tiles](#) offer an overview of the insights included.
- **Sample data.** A sample makes your app discoverable immediately after installation.

You might be familiar with each piece as existing Power BI features. When you build a template app, there are other things to consider for each piece. For details, see the following sections.

## Queries

For template apps, queries developed in Power BI Desktop are used to connect to your data source and import data. These queries are required to return a consistent schema and are supported for Scheduled Data refresh.

## Connect to your API

To get started, you need to connect to your API from Power BI Desktop to start building your queries.

You can use the Data Connectors that are available in Power BI Desktop to connect to your API. You can use the Web Data Connector ([Get data > Web](#)) to connect to your Rest API or the OData connector ([Get data > OData feed](#)) to connect to your OData feed.

### Note

Currently, template apps do not support custom connectors. We recommend exploring using Odatafeed Auth 2.0 as a mitigation for some of the connection use-cases or to submit your connector for certification. For details on how to develop a connector and certify it, see [Data Connectors](#).

## Consider the source

The queries define the data that's included in the data model. Depending on the size of your system, these queries should also include filters to ensure your customers are dealing with a manageable size that fits your business scenario.

Power BI template apps can run multiple queries in parallel and for multiple users concurrently. Plan your throttling and concurrency strategy and ask us how to make your template app fault tolerant.

## Schema enforcement

Ensure your queries are resilient to changes in your system. Changes in schema on refresh can break the model. If the source could return null or a missing schema result for some queries, consider returning an empty table or a meaningful custom error message.

## Parameters

[Parameters](#) in Power BI Desktop allow your users to provide input values that customize the data retrieved by the user. Think of the parameters up front to avoid rework after investing time to build detailed queries or reports.

### Note

Template apps support all parameters except `Any` and `Binary`.

## Additional query tips

- Ensure that all columns are typed appropriately.
- Assign columns informative names. For more information, see [Q&A](#).
- For shared logic, consider using functions or queries.

- Privacy levels are currently not supported in the service. If you get a prompt about privacy levels, you might need to rewrite the query to use relative paths.

## Data models

A well-defined data model ensures that your customers can easily and intuitively interact with the template app. Create the data model in Power BI Desktop.

### Note

You should do much of the basic modeling, such as typing and column names, in the [queries](#).

## Q&A

The modeling also affects how well Q&A can provide results for your customers. Be sure to add synonyms to commonly used columns, and properly name your columns in the [queries](#).

## Additional data model tips

Make sure you've:

- Applied formatting to all value columns. Apply types in the query.
- Applied formatting to all measures.
- Set default summarization. In particular, set *Do Not Summarize* when applicable, for unique values, for example.
- Set a data category, when applicable.
- Set relationships, as necessary.

## Reports

The report pages offer extra insight into the data included in your template app. Use the pages of the reports to answer the key business questions your template app is trying to address. Create the report using Power BI Desktop.

## Additional report tips

- Use more than one visual per page for cross-filtering.

- Align the visuals carefully, with no overlapping.
- Ensure that the page is set to 4:3 or 16:9 mode for layout.
- Ensure that all of the aggregations presented make numeric sense, for instance, averages or unique values.
- Check that slicing produces rational results.
- Include your logo on at least the top report.
- Ensure that elements are in the client's color scheme to the extent possible.

① Note

A single template app cannot include more than twenty reports.

## Dashboards

The dashboard is the main point of interaction with your template app for your customers. It should include an overview of the content included, especially the important metrics for your business scenario.

To create a dashboard for your template app, just upload your PBIX through [Get data > Files](#) or publish directly from Power BI Desktop.

## Additional dashboard tips

- Maintain the same theme when pinning so that the tiles on your dashboard are consistent.
- Pin a logo to the theme so consumers know where the pack is from.
- Suggested layout to work with most screen resolutions is five to six small tiles wide.
- All dashboard tiles should have appropriate titles and subtitles.
- Consider groupings in the dashboard for different scenarios, either vertically or horizontally.

## Sample data

A template app, as part of the app creation stage, wraps the cache data in the workspace as part of the app:

- Allows the installer to understand the functionality and purpose of the app before connecting data.

- Creates an experience that drives the installer to further explore app capabilities, which leads to connecting the app dataset.

We recommend having quality sample data before creating the app to ensure that the app's report and dashboards are populated with data. Try to keep sample data size as small as possible.

## Publishing on AppSource

Template apps can be published on AppSource. Follow these guidelines before submitting your app to AppSource:

- Make sure that you create a template app with engaging sample data that can help the installer understand what the app can do. Empty reports and dashboards won't be approved.
- Template apps support sample data only apps. Make sure to [check the static app checkbox](#).
- Have instructions for the validation team to follow that include credentials and parameters they can use to connect to the data.
- Your application must include an [App icon](#) in Power BI and on your cloud partner portal (CPP) offer.
- Configure the [landing page](#).
- Make sure to follow the documentation about the [Power BI App offer](#).
- If a dashboard is part of your app, make sure it's not empty.
- Install the app using the app link before submitting it. Make sure that you can connect the dataset and that the app experience is as you planned.
- Before uploading a PBIX file into the template workspace, make sure to unload any unnecessary connections.
- Follow Power BI [Best design practices for reports and visuals](#) to achieve maximum impact on your users and getting approved for distribution.

## Create a download link for the app

After publishing the template app on AppSource, consider creating a download link from your website to either:

- The AppSource download page, which can be viewed by publicly. Get the link from your AppSource page.
- Power BI, which can be viewed by a Power BI user.

In order to redirect a user to the app's download link in Power BI, see the following code example: [GitHub repo ↗](#).



## Automate parameter configuration during installation

If you're an independent software vendor (ISV) and are distributing your template app via your web service, you can create automation that configures template app parameters automatically when your customers install the app in their Power BI account. This approach makes things easier for your customers. It also increases the likelihood of a successful installation because they don't have to supply details that they might not know. For more information, see [Automated configuration of a template app installation](#).

## Next steps

[What are Power BI template apps?](#)

# Manage your published template app

Article • 03/15/2023

If you have a Power BI template app in production, when you want to make changes to the app, you can start over in the test phase, without interfering with the app in production.

## Update your app

Go to the template app workspace. Then, if you made your changes in Power BI Desktop, start at Step 1. If you did not make any changes in Power BI Desktop, start at Step 2.

### 1. Upload your updated dataset and **make sure to overwrite the existing dataset**.

- If the *.pbix* file you're uploading has the same name as the dataset and report used in the app, uploading will overwrite the existing dataset.
- If you're changing the name of the dataset and report used in the app, and the *.pbix* file you want to upload has a different name than the dataset and report used in the app, do the following:
  - Rename the dataset and report used in the app so that their names exactly match the name of your updated *.pbix* file.
  - Upload your *.pbix* file and overwrite the existing dataset and report that you just renamed.

In either case, to upload a local *.pbix* file to the service, select **Upload > Browse**, navigate to the file, and select **Open**. A dialog will ask for your permission to overwrite the dataset the app uses. If you don't overwrite the existing dataset, customers won't be able to install your updated app.

#### A dataset with the same name exists

You already have a dataset named AdventureWorksProducts.pbix in Power BI. If you replace the existing dataset, all changes you've made to reports online will be lost. The dashboard will remain, though some visuals might be impacted. Do you want to replace the existing dataset with this one?

[Learn more](#)

ⓘ If the existing dataset or report has a sensitivity label, it may be overwritten by the label in the imported file.

**Replace it**

**Cancel**

 **Important**

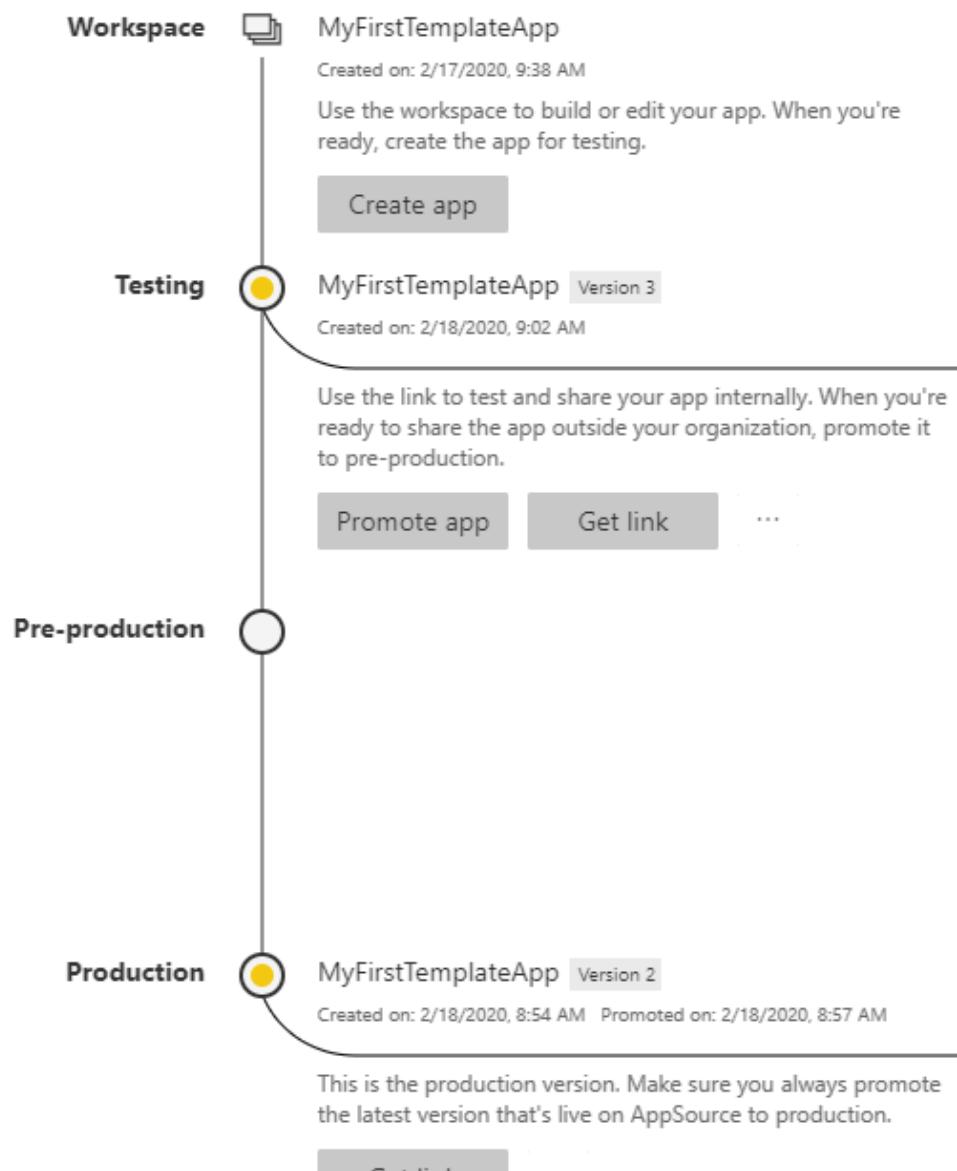
Never delete the dataset the app uses. Deleting the dataset makes it impossible for customers to update their copies of the app.

2. In the **Release management** pane for the app, select **Create app**.
3. Repeat the [app creation process](#). If you changed the name of the dataset and report used in the app, you might want to rename the app as well.
4. After you set **Branding**, **Navigation**, **Control**, **Parameters**, **Authentication**, and **Access**, select **Create app** again to save your changes, and then select **Close**.
5. Select **Release management** again.

In the **Release management** pane, you now see two versions of the app: The version in **Production**, plus a new version in **Testing**.

# Release Management

Track the progress of your template app development lifecycle, and prepare it for release outside your organization. [Learn more](#)

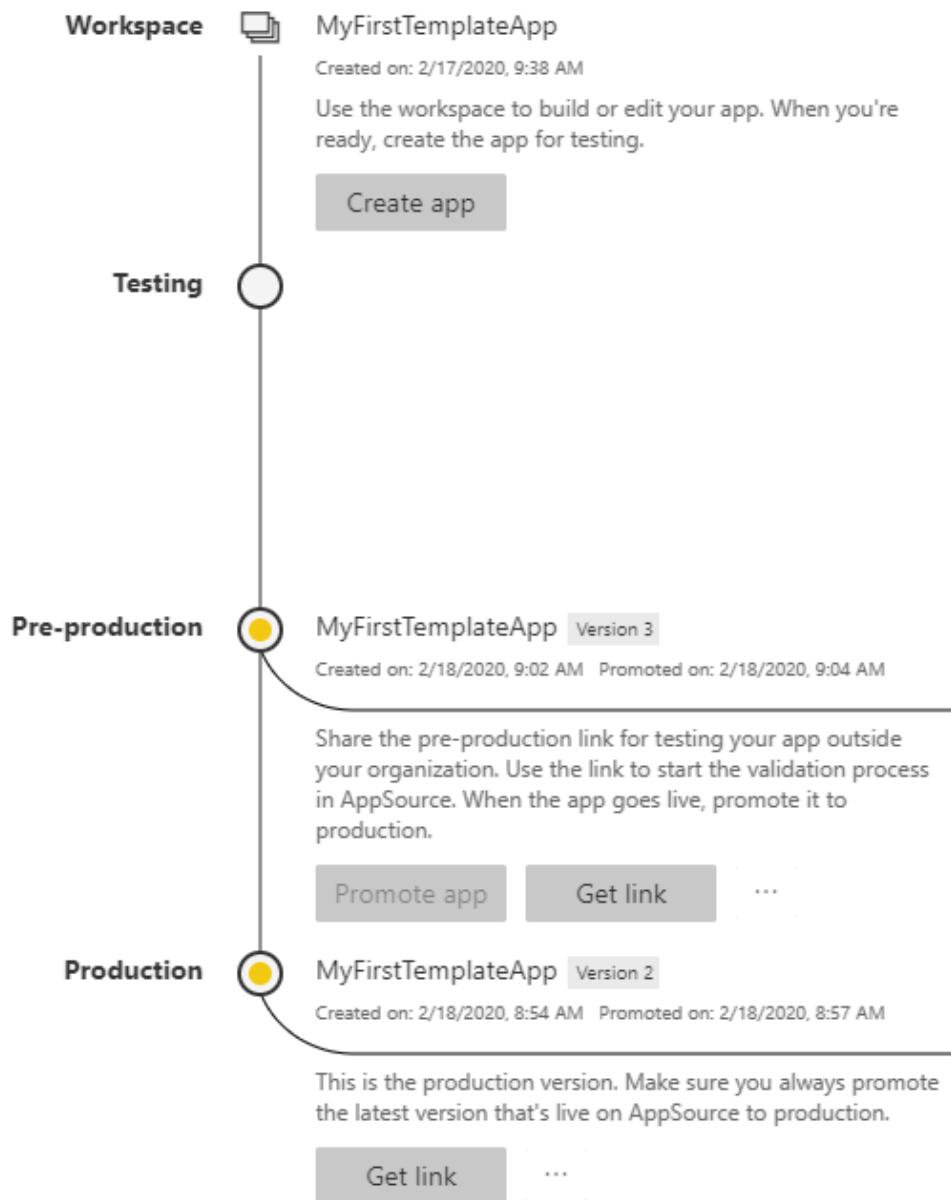


- When you're ready to promote your app to pre-production for further testing outside your tenant, go back to the **Release Management** pane and select **Promote app** next to **Testing**.

You now have a version in **Production** and a version in **Pre-production**.

# Release Management

Track the progress of your template app development lifecycle, and prepare it for release outside your organization. [Learn more](#)



Your link is now live.

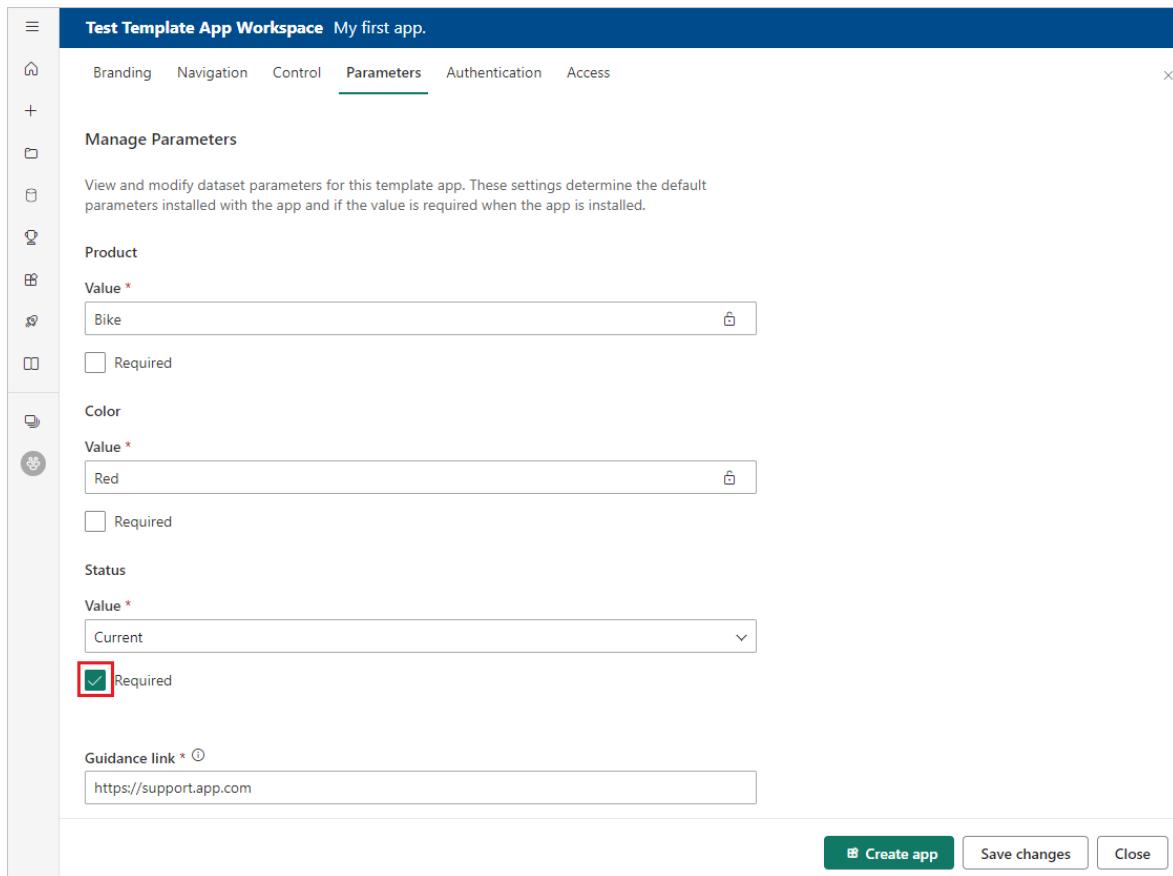
**Note**

The **Promote app** button at the pre-production stage is disabled. Disabling the button prevents accidentally overwriting the live production link with the current app version before the Cloud Partner Portal (CPP) validates and approves the new app version.

7. Submit your link again to the CPP by following the steps at [Power BI App offer update](#). In the CPP, you must publish your offer again and have it validated and approved. If you've changed the name of the app, be sure to also change the name in the CPP. When your offer is approved, the **Promote app** button becomes active again.
8. Promote your app to the **Production** stage.

## Update behavior

- Updating the app lets template app installers [update their template app](#) in the already installed workspaces without losing the connection configuration.
- To learn how changes in the dataset affect the installed template app, see [Overwrite behavior](#).
- When a template app is overwritten and updates, it first reverts back to sample data, and automatically reconnects using the installer's configuration parameters and authentication. Until refresh is complete, the reports, dashboards, and organizational app display the sample data banner.
- If you added a new query parameter to the updated dataset that requires user input, you must select the **Required** checkbox. This selection prompts the installer with the connection string after updating the app.



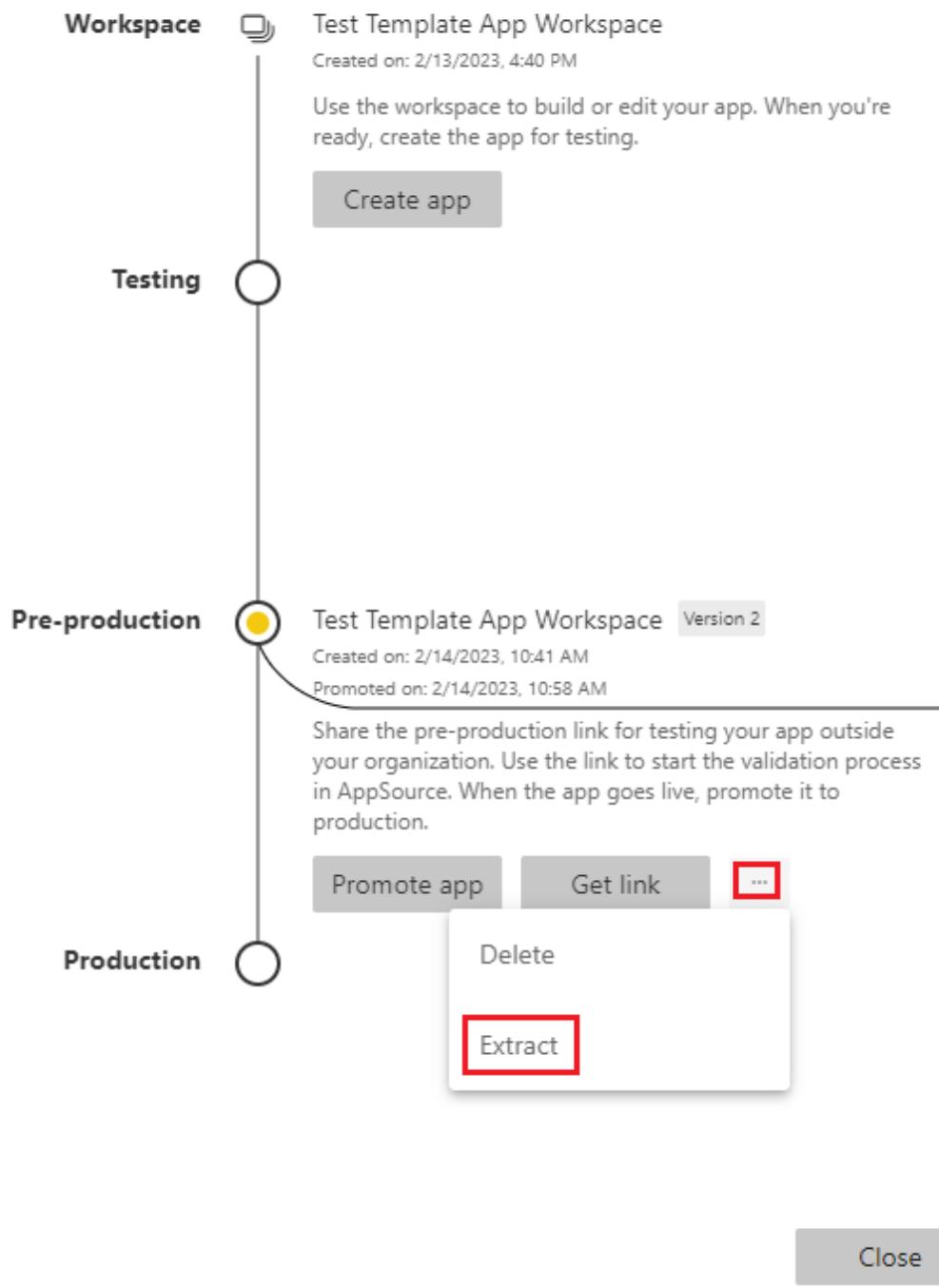
## Extract workspace

It's easy to roll back to the previous version of a template app with the extract capability. The following steps extract a specific app version from a release stage into a new workspace:

1. In the **Release Management** pane, next to an app version, select **More options (...)** and then select **Extract**.

# Release Management

Track the progress of your template app development lifecycle, and prepare it for release outside your organization. [Learn more](#)



2. In the confirmation dialog box, enter a name for the extracted workspace, and select **Extract**. Power BI adds a new workspace for the extracted app.

## Extract this app?

The contents of this app will be extracted and copied to a new template app workspace.

Enter a name for the new workspace

ExtractCancel

Your new workspace versioning resets, and you can continue to develop and distribute the template app from the newly extracted workspace.

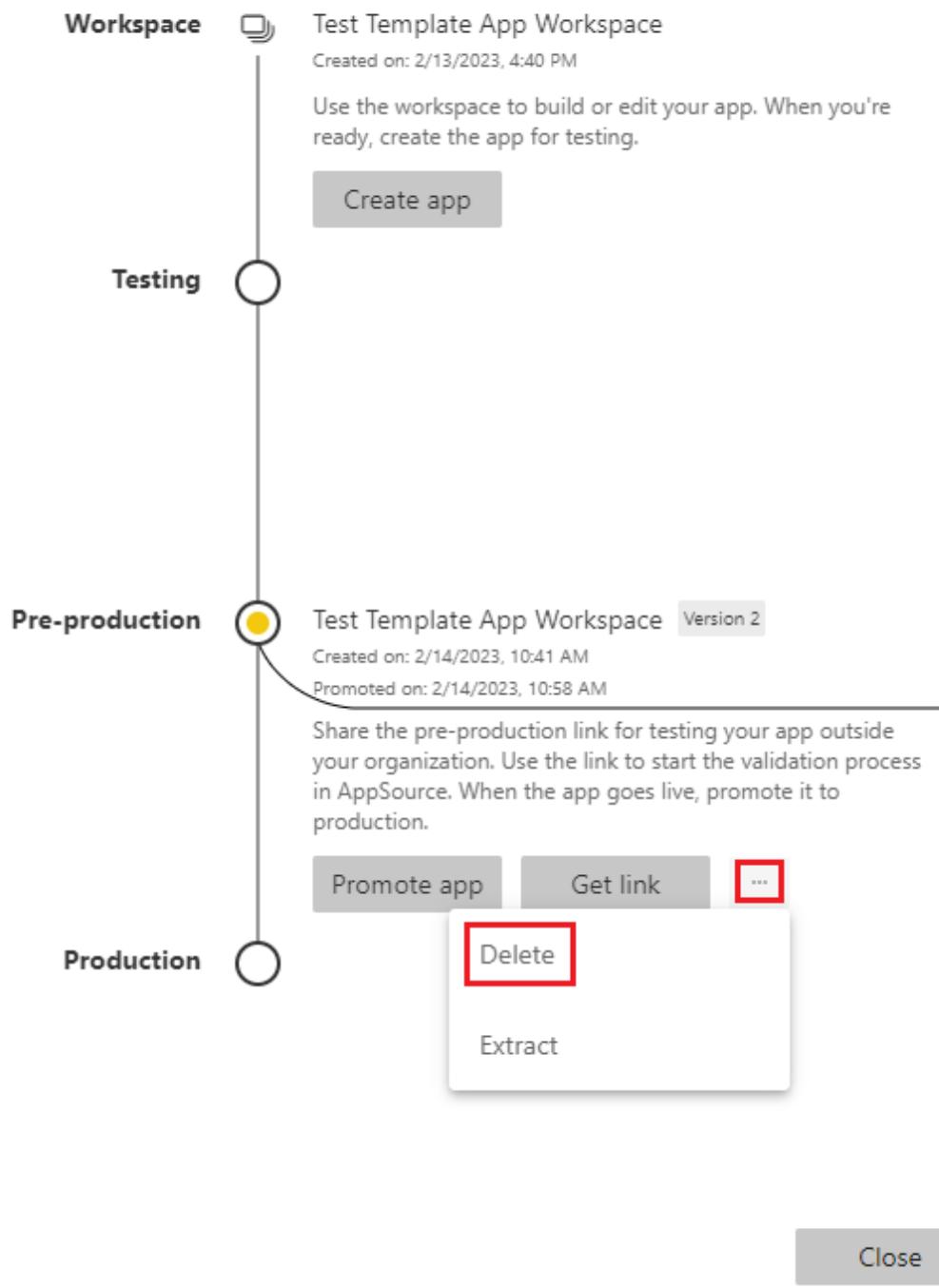
## Delete template app version

A template app workspace is the source of an active distributed template app. To protect the template app users, it's not possible to delete a template app workspace without first removing all the created app versions in the workspace. Deleting an app version also deletes the app URL, which no longer works.

1. In the **Release Management** pane, next to the app version you want to delete, select **More options (...)** and then select **Delete**.

# Release Management

Track the progress of your template app development lifecycle, and prepare it for release outside your organization. [Learn more](#)



2. In the confirmation dialog box, select **Delete**.

## Delete this app?

×

If you delete this app, "Test Template App Workspace" v2, it will be permanently deleted and the link used for installing it will stop working.

If you submitted this link to AppSource, it will no longer work.

**Delete**

**Cancel**

### ① Note

Make sure not to delete app versions that customers or AppSource are using, or they will no longer work.

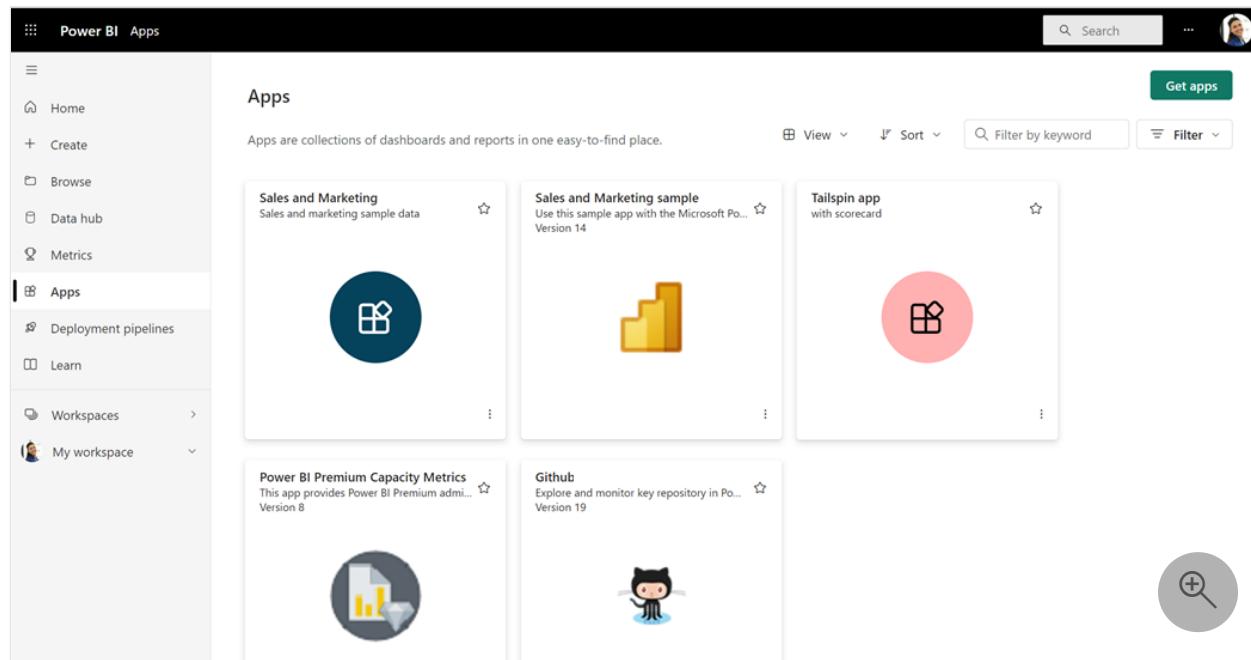
## Next steps

- See how your customers interact with your template app in [Install, customize, and distribute template apps in your organization](#).
- See the [Power BI Application offer](#) for details on distributing your app.

# Install, share, and update template apps in your organization

Article • 02/08/2023

Are you a Power BI analyst? Here you can learn more about [template apps](#) and how to connect to many of the services that you use to run your business, such as Salesforce, Microsoft Dynamics, and Google Analytics. You can then modify the template app's pre-built dashboard and reports to suit the needs of your organization, and distribute them to your colleagues as [apps](#).



If you're interested in creating template apps yourself for distribution outside your organization, see [Create a template app in Power BI](#). With little or no coding, Power BI partners can build Power BI apps and make them available to Power BI customers.

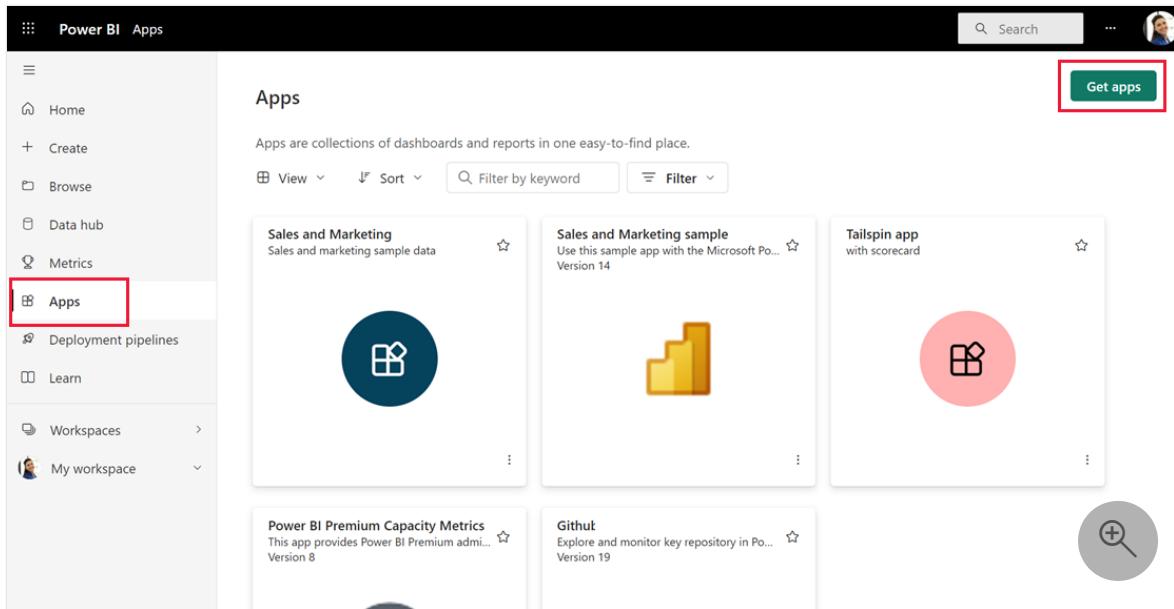
## Prerequisites

To install, customize, and distribute a template app, you need:

- A [Power BI pro license](#).
- Permissions to install template apps on your tenant.
- A valid installation link for the app, which you get either from AppSource or from the app creator.
- A good familiarity with the [basic concepts of Power BI](#).

## Install a template app

1. In the nav pane in the Power BI service, select **Apps > Get apps**.



The screenshot shows the Power BI Apps interface. On the left, there's a navigation pane with various options like Home, Create, Browse, Data hub, Metrics, Apps (which is highlighted with a red box), Deployment pipelines, Learn, Workspaces, and My workspace. On the right, the main area is titled 'Apps' with a sub-instruction: 'Apps are collections of dashboards and reports in one easy-to-find place.' It features a search bar, filter buttons for View, Sort, and Filter, and a 'Get apps' button at the top right. Below these are five app cards: 'Sales and Marketing' (blue icon), 'Sales and Marketing sample' (yellow icon), 'Tailspin app' (pink icon), 'Power BI Premium Capacity Metrics' (grey icon), and 'Github' (grey icon). A magnifying glass icon is also present.

2. In the Power BI apps marketplace that appears, select **Template apps**. All the template apps available in AppSource are shown. Browse to find the template app you're looking for, or get a filtered selection by using the search box. Type part of the name of the template app, or select a category such as finance, analytics, or marketing to find the item you're looking for.



The screenshot shows the Power BI apps marketplace. At the top, it says 'Power BI apps' and 'Install apps that provide actionable insights and drive business results'. Below that are tabs for 'All apps', 'Organizational apps', and 'Template apps' (which is highlighted with a red box). There's a search bar, a 'Filter by' dropdown set to 'All', and a 'Sort by: Popularity' button. The interface is clean and modern, designed for easy navigation through the app catalog.

3. When you find the template app you're looking for, select it. The template app offer appears. Select **Get It Now**.

Screenshot of the Microsoft AppSource page for the GitHub Power BI app.

The page shows the GitHub logo and title "Github". It has a rating of 3.4 (20) stars. The "Reviews" tab is selected. A large blue button labeled "GET IT NOW" is highlighted with a red box.

On the left, there's a sidebar with links to Pricing (Free), Products (Power BI apps), Publisher (Microsoft), Acquire Using (Work or school account), Categories (Analytics), Support, Legal, License Agreement, and Privacy Policy.

The main content area describes the app: "Visualize different types of repository activity such as how many commits, active users and more." It also provides a detailed description of the app's features, mentioning Power BI dashboards and reports.

To the right, a screenshot of the Power BI dashboard titled "Power BI" is shown, displaying various metrics and charts related to GitHub repository usage.

4. In the dialog box that appears, select **Install**.

Screenshot of the "Install this Power BI app?" dialog box.

The dialog shows the GitHub logo and title "Github Repository Usage" by Microsoft. There is a link to "View in AppSource".

A note states: "Apps may contain security or privacy risks. Only install apps from trusted authors and sources." with a "Learn more" link.

By installing this app I agree to the publisher's [privacy policy](#) and [terms of service](#).

At the bottom, there are two buttons: a green "Install" button with a red border and a "Cancel" button.

The app is installed, along with a workspace of the same name that has all the artifacts needed for further [customization](#).

### ⓘ Note

If you use an installation link for an app that isn't listed on AppSource, a validation dialog box will ask you to confirm your choice.

To be able to install a template app that isn't listed on AppSource, you can request the relevant permissions from your admin. See the [template app settings](#) in Power BI admin portal for details.

When the installation finishes successfully, a notification tells you that your new app is ready.

✓ Your new app is ready!  
App Github  
successfully installed.

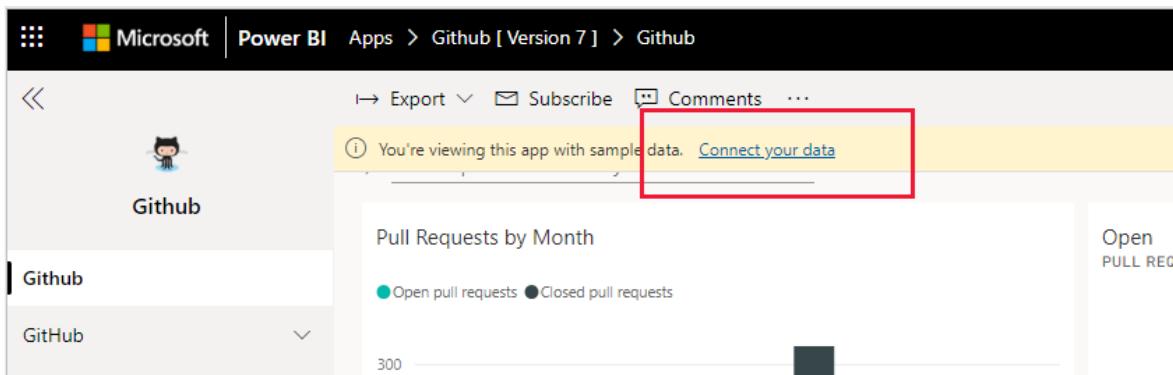
[Go to app](#)

## Connect to data

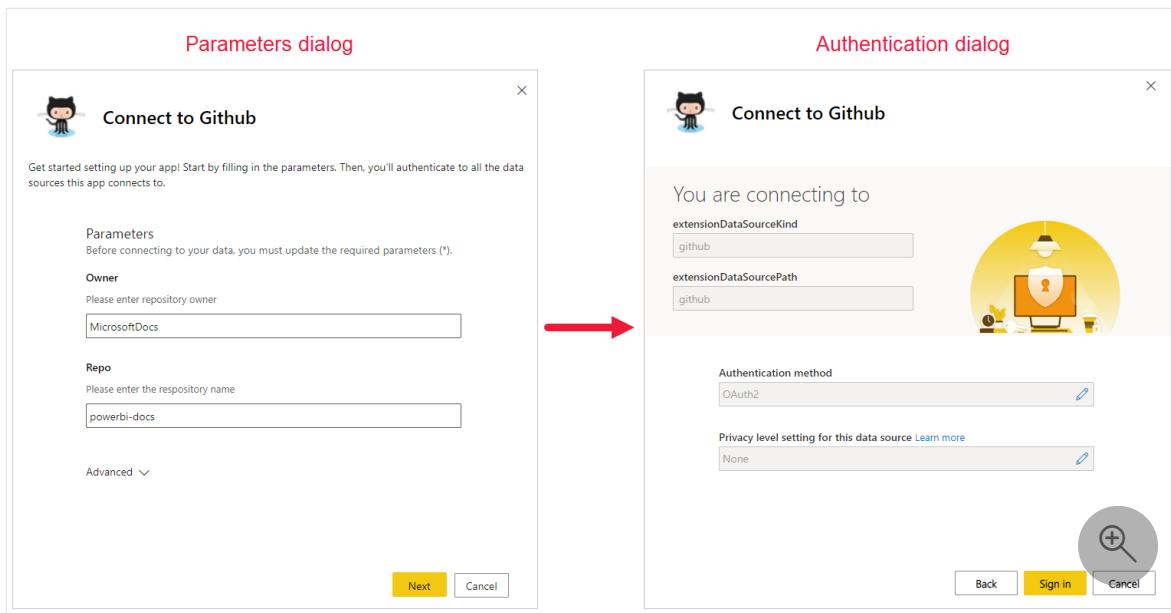
1. Select **Go to app**.

The app opens, showing sample data.

2. Select the **Connect your data** link on the banner at the top of the page.



This link opens the parameters dialog, where you change the data source from the sample data to your own data source (see [known limitations](#)), followed by the authentication method dialog. You might have to redefine the values in these dialogs. See the documentation of the specific template app you're installing for details.



Once you've finished filling out the connection dialogs, the connection process starts. A banner informs you that the data is being refreshed, and that in the meantime you're viewing sample data.

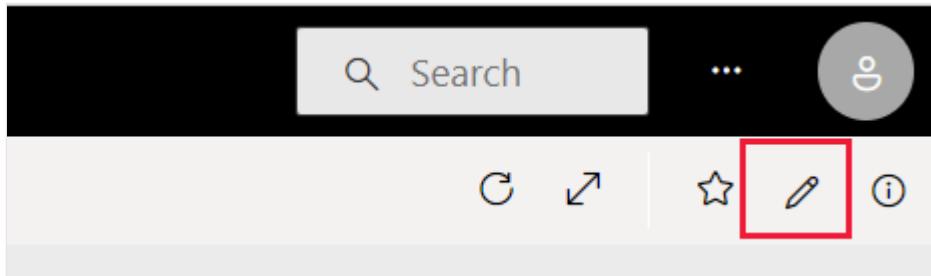
The screenshot shows the Power BI app interface for a GitHub report. The top navigation bar includes 'Microsoft', 'Power BI', 'Apps > Github [Version 7] > Github'. Below the navigation is a toolbar with 'Export', 'Subscribe', 'Comments', and more. A yellow banner at the top states 'You're viewing this app with sample data. Refresh is in progress.' A chart titled 'Pull Requests by Month' is displayed, showing a single dark bar reaching the 300 mark on the y-axis. The left sidebar lists 'Github' and 'GitHub' under the main 'Github' category.

Your report data will automatically refresh once a day, unless you disabled this setting during the sign-in process. You can also [set up your own refresh schedule](#) to keep the report data up to date if you so desire.

## Customize and share the app

After you've connected to your data and data refresh is complete, you can customize any of the reports and dashboards the app includes, as well as share the app with your colleagues. Remember, however that any changes you make will be overwritten when you update the app with a new version, unless you save the items you changed under different names. [See details about overwriting](#).

To customize and share your app, select the pencil icon at the top right corner of the page.



For information about editing artifacts in the workspace, see

- [Tour the report editor in Power BI](#)
- [Basic concepts for designers in the Power BI service](#)

When you're done making changes to the artifacts in the workspace, you're ready to publish and share the app. See [Publish your app](#) to learn how.

## Update a template app

From time to time, template app creators release new improved versions of their template apps, via AppSource, a direct link, or both.

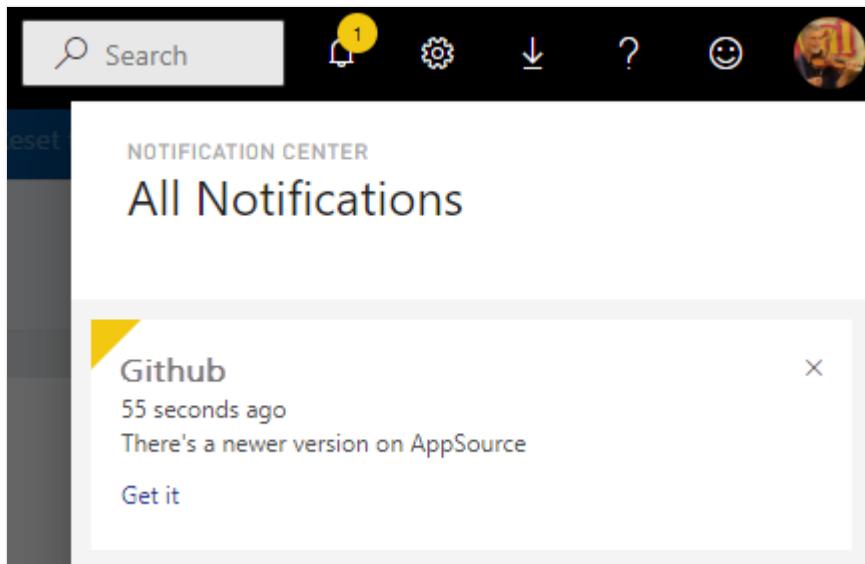
If you originally downloaded the app from AppSource, when a new version of the template app becomes available, you get notified in two ways:

- An update banner appears in the Power BI service informing you that a new app version is available.

A screenshot of the Power BI service interface. At the top, there's a navigation bar with a GitHub logo, the text "Github Power BI private repo", and a yellow "Update app" button. Below the navigation bar, a banner says "A newer version of this app is available on AppSource" with a "Get it now" button. The main area shows a table of apps with columns: Name, Type, Owner, Refreshed, Sensitivity (preview), Endorsement, and Include in app. Two apps are listed: "Github" (Dashboard, Owner: GitHub Power BI priva..., Refreshed: —, Sensitivity: —, Endorsement: —, Include in app: Yes) and "GitHub" (Report, Owner: GitHub Power BI priva..., Refreshed: 1/17/20, 1:31:05 AM, Sensitivity: —, Endorsement: —, Include in app: Yes).

Name	Type	Owner	Refreshed	Sensitivity (preview)	Endorsement	Include in app
Github	Dashboard	Github Power BI priva...	—	—	—	
GitHub	Report	Github Power BI priva...	1/17/20, 1:31:05 AM	—	—	

- You receive a notification in Power BI's notification pane.



### ① Note

If you originally got the app via a direct link rather than through AppSource, the only way to know when a new version is available is to contact the template app creator.

To install the update, either select **Get it** on the notification banner or in the notification center, or find the app again in AppSource and choose **Get it now**. If you got a direct link for the update from the Template app creator, select the link.

You're asked how you want the update to affect your currently installed app.

### Update app



**GitHub Repository Usage**  
by Microsoft  
[View in AppSource](#)

Select how to update this app:

Update the workspace and the app

⚠ Any changes you've made to the workspace content or app settings may be overwritten. [Learn more](#)

Update only workspace content without updating the app

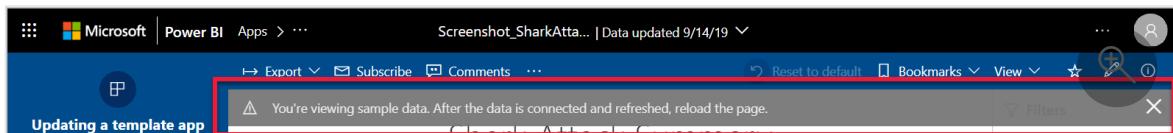
Install another copy of the app into a new workspace

Install Cancel

- **Update the workspace and the app:** Updates both the workspace and the app, and republishes the app to your organization. Choose this option if you didn't make any changes to the app or its content and want to overwrite the old app. Your connections will be re-established, and the new version of the app will include any updated app branding, such as app name, logo, and navigation, as well as the latest publisher improvements to content.
- **Update only workspace content without updating the app:** Updates the reports, dashboards, and dataset in the workspace. After updating the workspace, you can choose what you want to include in the app, and then you need to update the app to republish it to your organization with the changes.
- **Install another copy of the app into a new workspace:** Installs a fresh version of the workspace and app. Choose this option if you don't want to change your current app.

## Overwrite behavior

- Overwriting updates the reports, dashboards, and dataset in the workspace, not the app. Overwriting doesn't change app navigation, setup, and permissions.
- If you chose the second option, after you've updated the workspace **you need to update the app to apply changes from the workspace to the app.**
- Overwriting keeps configured parameters and authentication. After the update, an automatic dataset refresh starts. **During this refresh, the app, reports, and dashboards present sample data.**



- Overwriting always presents sample data until the refresh is complete. If the template app author made changes to the dataset or parameters, users of the workspace and app won't see the new data until the refresh is complete. Instead, they'll still see sample data during this time.
- Overwriting never deletes new reports or dashboards you've added to the workspace. It only overwrites the original reports and dashboards with changes from the original author.

**ⓘ Important**

Remember to **update the app** after overwriting to apply changes to the reports and dashboard for your organizational app users.

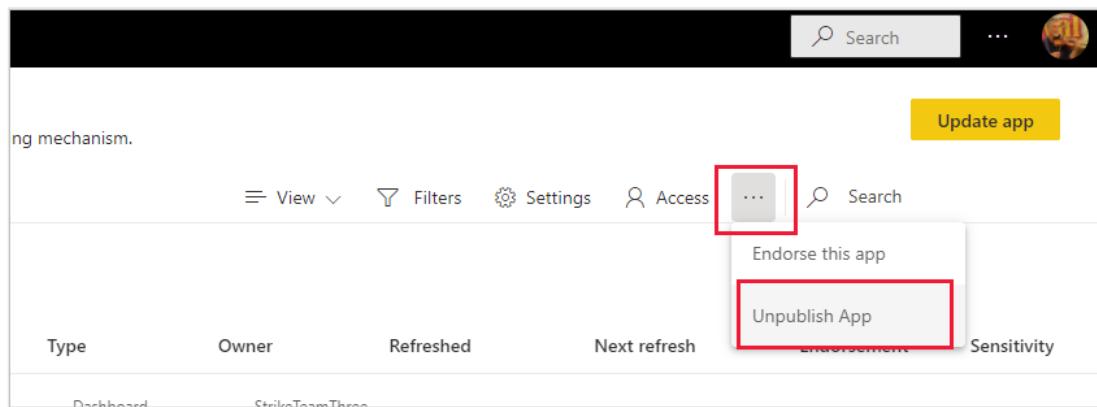
## Delete a template app

An installed template app consists of the app and its associated workspace. If you want to remove the template app, you have two options:

- **Completely remove the app and its associated workspace:** To completely remove a template app and its associated workspace, go to the app tile on the Apps page, select the trash icon, and then choose **Delete** in the dialog that appears.
- **Unpublish the app:** This option removes the app but keeps its associated workspace. This option is useful if there are customizations that you made and want to keep.

To unpublish the app:

1. Open the app.
2. Select the edit app pencil icon to open the template app's workspace.
3. In the template app workspace, select **More options (...)**, and then choose **Unpublish App**.



## Next steps

[Create a workspace in Power BI](#)

# Automated configuration of a template app installation

Article • 11/29/2021

Template apps are a great way for customers to start getting insights from their data. Template apps get them up and running quickly by connecting them to their data. The template apps provide them with prebuilt reports that they can customize if they so desire.

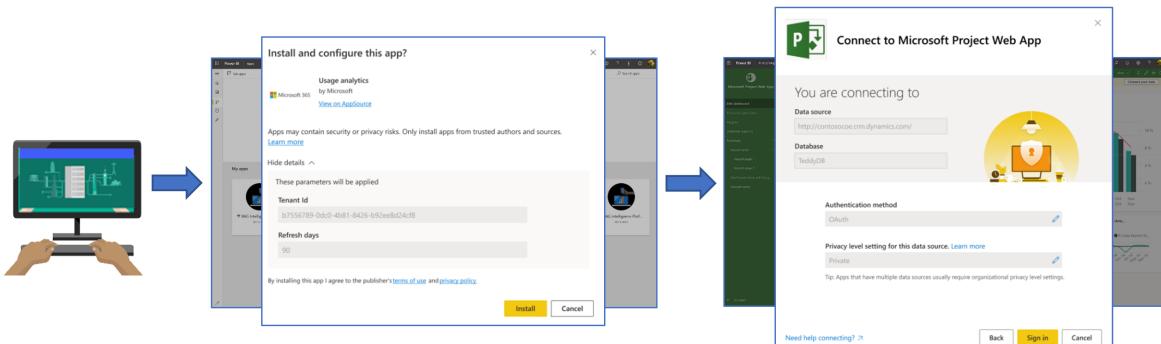
Customers aren't always familiar with the details of how to connect to their data. Having to provide these details when they install a template app can be a pain point for them.

If you are a data services provider and have created a template app to help your customers get started with their data on your service, you can make it easier for them to install your template app. You can automate the configuration of your template app's parameters. When the customer signs in to your portal, they select a special link you've prepared. This link:

- Launches the automation, which gathers the information it needs.
- Preconfigures the template app parameters.
- Redirects the customer to their Power BI account where they can install the app.

All they have to do is select **Install** and authenticate against their data source, and they're good to go!

The customer experience is illustrated here.



Customer logs into your portal and clicks the link to install your preconfigured template app.

This launches your automation which then gathers the info it needs and preconfigures the template app parameters.

The automation redirects the customer to their Power BI account where they can see the preconfigured parameters and install the template app.

Customer authenticates on the data source and starts getting insights from their own data in the prebuilt reports your template app offers

This article describes the basic flow, the prerequisites, and the main steps and APIs you need to automate the configuration of a template app installation. If you want to dive in and get started, you can skip to the [tutorial](#) where you automate the configuration of the template app installation by using a simple sample application we've prepared that uses an Azure function.

## Basic flow

The basic flow of automating the configuration of a template app installation is as follows:

1. The user signs in to the ISV's portal and selects the supplied link. This action initiates the automated flow. The ISV's portal prepares the user-specific configuration at this stage.
2. The ISV acquires an *app-only* token based on a [service principal \(app-only token\)](#) that's registered in the ISV's tenant.
3. Using [Power BI REST APIs](#), the ISV creates an *install ticket*, which contains the user-specific parameter configuration as prepared by the ISV.
4. The ISV redirects the user to Power BI by using a `POST` redirection method that contains the install ticket.
5. The user is redirected to their Power BI account with the install ticket and is prompted to install the template app. When the user selects **Install**, the template app is installed for them.

### Note

While parameter values are configured by the ISV in the process of creating the install ticket, data source-related credentials are only supplied by the user in the final stages of the installation. This arrangement prevents them from being exposed to a third party and ensures a secure connection between the user and the template app data sources.

## Prerequisites

To provide a preconfigured installation experience for your template app, the following prerequisites are required:

- A Power BI Pro license. If you're not signed up for Power BI Pro, [sign up for a free trial](#) before you begin.
- Your own Azure Active Directory (Azure AD) tenant set up. For instructions on how to set one up, see [Create an Azure AD tenant](#).
- A **service principal (app-only token)** registered in the preceding tenant. For more information, see [Embed Power BI content with service principal and an application secret](#). Make sure to register the application as a **server-side web application** app. You register a server-side web application to create an application secret. From this process, you need to save the *application ID* (ClientID) and *application secret* (ClientSecret) for later steps.
- A **parameterized template app** that's ready for installation. The template app must be created in the same tenant in which you register your application in Azure AD. For more information, see [Template app tips](#) or [Create a template app in Power BI](#). From the template app, you need to note the following information for the next steps:
  - *App ID*, *Package Key*, and *Owner ID* as they appear in the installation URL at the end of the process of [defining the properties of the template app](#) when the app was created. You can also get the same link by selecting **Get link** in the template app's [Release Management](#) pane.
  - *Parameter names* as they're defined in the template app's dataset. Parameter names are case-sensitive strings and can also be retrieved from the **Parameter Settings** tab when you [define the properties of the template app](#) or from the dataset settings in Power BI.
- To be able to test your automation work flow, add the service principal to the template app workspace as an Admin.

 **Note**

You can test your preconfigured installation application on your template app if the template app is ready for installation, even if it isn't publicly available on AppSource yet. For users outside your tenant to be able to use the automated installation application to install your template app, the template app must be publicly available in the [Power BI apps marketplace](#). Before you distribute your template app by using the automated installation application you're creating, be sure to publish it to [Partner Center](#).

## Main steps and APIs

The main steps for automating the configuration of a template app installation, and the APIs you'll need, are described in the following sections. While most of the steps are done with [Power BI REST APIs](#), the code examples described here are made with the .NET SDK.

## Step 1: Create a Power BI client object

Using Power BI REST APIs requires you to get an *access token* for your [service principal](#) from Azure AD. You're required to get an [Azure AD access token](#) for your Power BI application before you make calls to the [Power BI REST APIs](#). To create the Power BI client with your access token, you need to create your Power BI client object, which allows you to interact with the [Power BI REST APIs](#). You create the Power BI client object by wrapping the **AccessToken** with a **Microsoft.Rest.TokenCredentials** object.

C#

```
using Microsoft.IdentityModel.Clients.ActiveDirectory;
using Microsoft.Rest;
using Microsoft.PowerBI.Api.V2;

var tokenCredentials = new
TokenCredentials(authenticationResult.AccessToken, "Bearer");

// Create a Power BI client object. It's used to call Power BI APIs.
using (var client = new PowerBIClient(new Uri(ApiUrl), tokenCredentials))
{
 // Your code goes here.
}
```

## Step 2: Create an install ticket

Create an install ticket, which is used when you redirect your users to Power BI. The API used for this operation is the [CreateInstallTicket](#) API.

- [Template Apps CreateInstallTicket](#)

A sample of how to create an install ticket for template app installation and configuration is available from the [InstallTemplateApp/InstallAppFunction.cs](#) file in the [sample application](#).

The following code example shows how to use the template app [CreateInstallTicket](#) REST API.

C#

```

using Microsoft.PowerBI.Api.V2;
using Microsoft.PowerBI.Api.V2.Models;

// Create Install Ticket Request.
InstallTicket ticketResponse = null;
var request = new CreateInstallTicketRequest()
{
 InstallDetails = new List<TemplateAppInstallDetails>()
 {
 new TemplateAppInstallDetails()
 {
 AppId = Guid.Parse(AppId),
 PackageKey = PackageKey,
 OwnerTenantId = Guid.Parse(OwnerId),
 Config = new TemplateAppConfigurationRequest()
 {
 Configuration = Parameters
 .GroupBy(p => p.Name)
 .ToDictionary(k => k.Key, k =>
k.Select(p => p.Value).Single())
 }
 }
 }
};

// Issue the request to the REST API using .NET SDK.
InstallTicket ticketResponse = await
client.TemplateApps.CreateInstallTicketAsync(request);

```

## Step 3: Redirect users to Power BI with the ticket

After you've created an install ticket, you use it to redirect your users to Power BI to continue with the template app installation and configuration. You use a `POST` method redirection to the template app's installation URL, with the install ticket in its request body.

There are various documented methods of how to issue a redirection by using `POST` requests. Choosing one or another depends on the scenario and how your users interact with your portal or service.

A simple example, mostly used for testing purposes, uses a form with a hidden field, which automatically submits itself upon loading.

JavaScript

```

<html>
 <body onload='document.forms["form"].submit()'>
```

```

<!-- form method is POST and action is the app install URL -->
<form name='form' action='https://app.powerbi.com/....'
method='post' enctype='application/json'>
 <!-- value should be the new install ticket -->
 <input type='hidden' name='ticket' value='H4sI....AAA='>
</form>
</body>
</html>

```

The following example of the [sample application](#)'s response holds the install ticket and automatically redirects users to Power BI. The response for this Azure function is the same automatically self-submitting form that we see in the preceding HTML example.

C#

```

...
 return new ContentResult() { Content = RedirectWithData(redirectUrl,
ticket.Ticket), ContentType = "text/html" };
}

...

public static string RedirectWithData(string url, string ticket)
{
 StringBuilder s = new StringBuilder();
 s.Append("<html>");
 s.AppendFormat("<body onload='document.forms[\"form\"].submit()'>");
 s.AppendFormat("<form name='form' action='{0}' method='post'
enctype='application/json'>", url);
 s.AppendFormat("<input type='hidden' name='ticket' value='{0}' />",
ticket);
 s.Append("</form></body></html>");
 return s.ToString();
}

```

### Note

There are various methods of using `POST` browser redirects. You should always use the most secure method, which depends on your service needs and restrictions. Remember that some forms of insecure redirection can result in exposing your users or service to security issues.

## Step 4: Move your automation to production

When the automation you've designed is ready, be sure to move it to production.

## Next steps

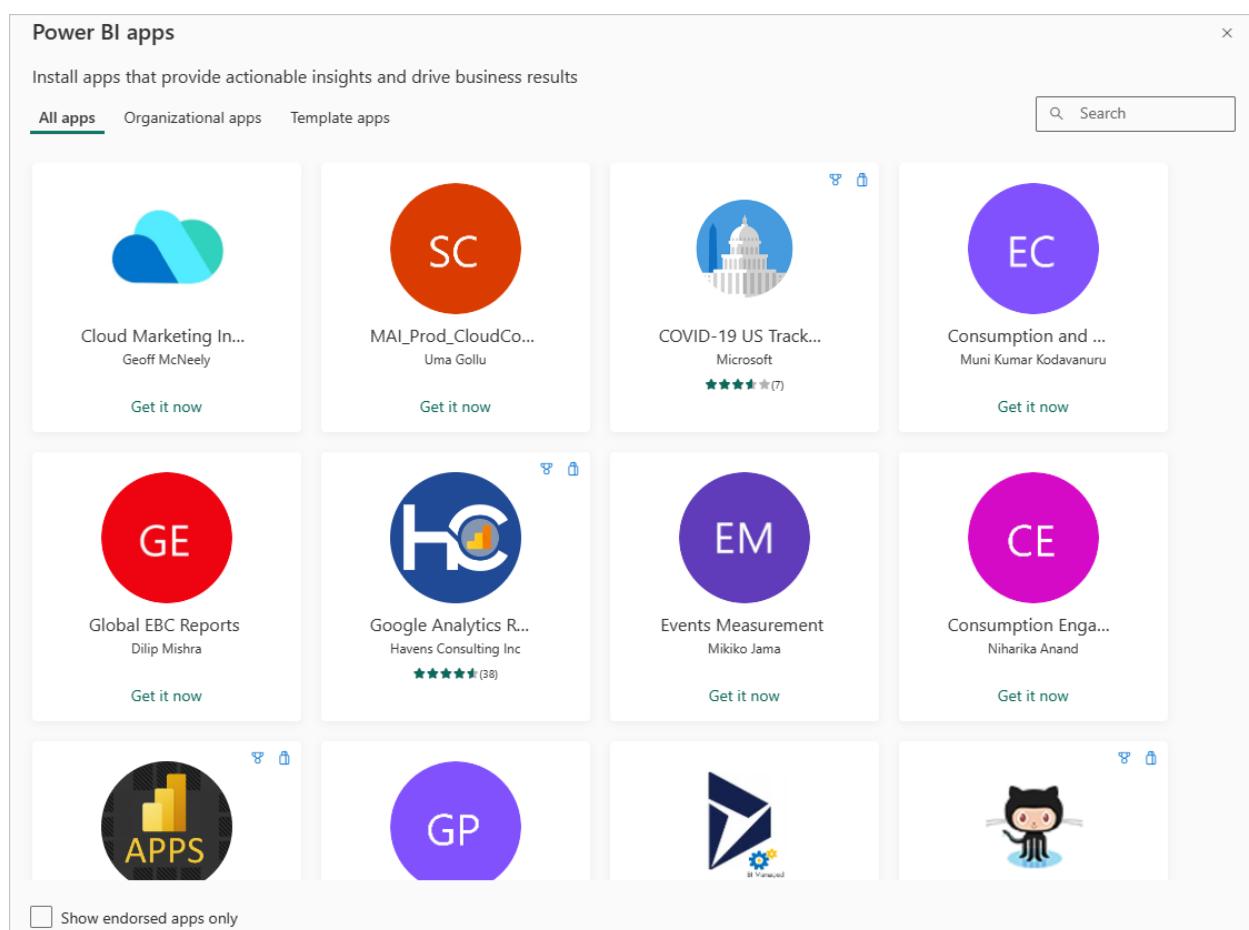
- Try our [tutorial](#), which uses a simple Azure function to automate the configuration of a template app installation.
- More questions? [Try asking the Power BI Community](#).

# Connect to the services you use with Power BI

Article • 01/09/2023

With Power BI, you can connect to many of the services you use to run your business, such as Salesforce, Microsoft Dynamics, and Google Analytics. Power BI starts by using your credentials to connect to the service. It creates a Power BI *workspace* with a dashboard and a set of Power BI reports that automatically show your data and provide visual insights about your business.

Sign in to Power BI to view all of the services you can connect to. Select **Apps > Get apps**.

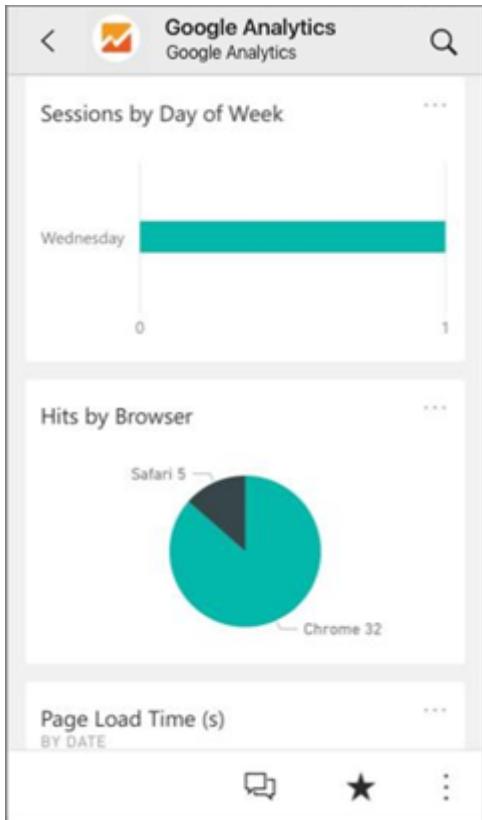


The screenshot shows the 'Power BI apps' page with a search bar at the top right. Below the search bar are three filter buttons: 'All apps' (which is selected), 'Organizational apps', and 'Template apps'. A 'Search' input field is also present. The main area displays a 4x3 grid of app cards. Each card includes the app icon, name, author, rating, and a 'Get it now' button. The apps listed are:

- Cloud Marketing In... (Geoff McNeely) - Get it now
- MAI\_Prod\_CloudCo... (Uma Gollu) - Get it now
- COVID-19 US Track... (Microsoft) - ★★★★☆(7)
- Consumption and ... (Muni Kumar Kodavanuru) - Get it now
- Global EBC Reports (Dilip Mishra) - Get it now
- Google Analytics R... (Havens Consulting Inc) - ★★★★☆(38)
- Events Measurement (Mikiko Jama) - Get it now
- Consumption Enga... (Niharika Anand) - Get it now
- APPS
- GP
- Advanced
- Github

At the bottom left, there is a checkbox labeled 'Show endorsed apps only'.

After you install the app, you can view the dashboard and reports in the app and the workspace in the Power BI service (<https://powerbi.com>). You can also view them in the Power BI mobile apps. In the workspace, you can modify the dashboard and reports to meet the needs of your organization, and then distribute them to your colleagues as an *app*.



## Get started

1. Select Apps in the navigation pane, then choose Get apps in the upper-right corner.

The screenshot shows the Power BI service interface. On the left, there is a navigation pane with the following items:

- Home
- Create
- Browse
- Data hub
- Metrics
- Apps** (highlighted with a red box)
- Deployment pipelines
- Learn

The main area is titled "Apps" and contains the following information:

- A description: "Apps are collections of dashboards and reports in one easy-to-find place."
- Filtering options: "View" and "Sort" dropdowns, a "Filter by keyword" search bar, and a "Filter" button.
- Two app cards:
  - "New Opportunity Analysis opportunity" (with a star icon)
  - "Covid-19 US Tracking" (Sample app that tracks Covid-19 US stats ... with a star icon)
- A large blue circular button with a white icon in the bottom right corner.

2. In Power BI apps, select the Apps tab, and search for the service you want.

## Power BI apps

Install apps that provide actionable insights and drive business results

All apps

Organizational apps

Template apps

Search

# Edit the dashboard and reports

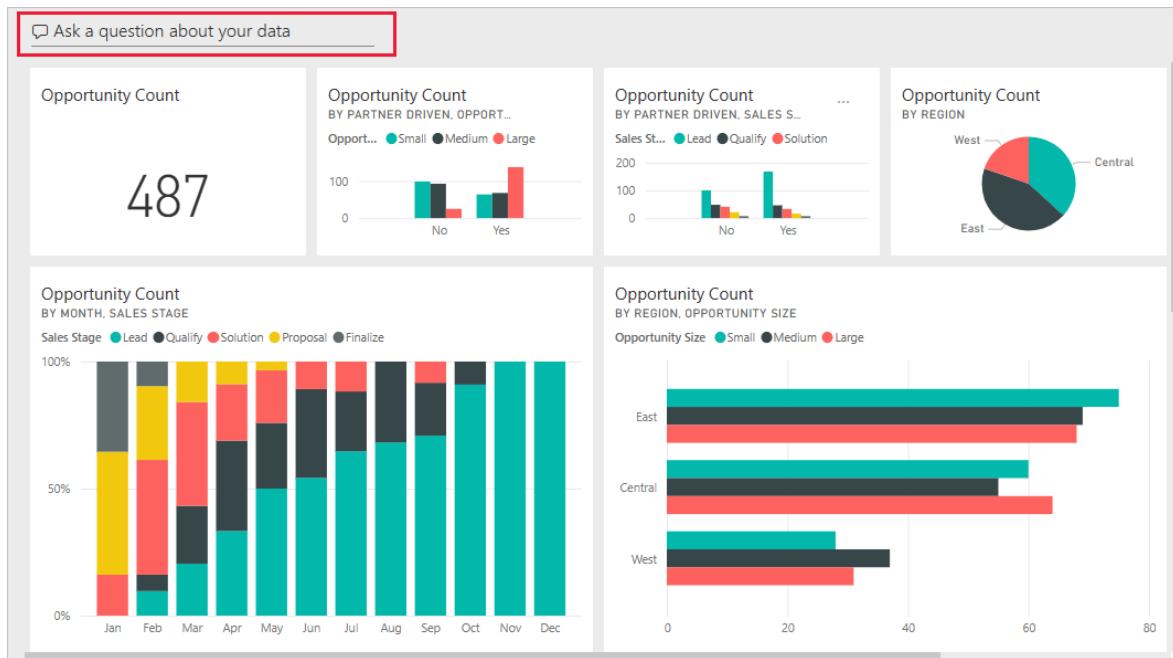
When the import is complete, the new app appears on the Apps page.

1. Select Apps in the navigation pane, then choose the app.

The screenshot shows the Power BI Apps interface. On the left, a navigation pane includes options like Home, Create, Browse, Data hub, Metrics, Apps (which is highlighted with a red box), Deployment pipelines, Learn, Workspaces, and My workspace. The main area is titled 'Apps' and contains a brief description: 'Apps are collections of dashboards and reports in one easy-to-find place.' It features a search bar ('Filter by keyword') and a filter dropdown ('Filter'). Below this, four app tiles are displayed:

- New Opportunity Analysis** (highlighted with a red box): opportunity. This tile shows a thumbnail of a dashboard with a yellow flower, the creation date (6/3/21, 2:51:09 PM), and a three-dot menu icon.
- Covid-19 US Tracking**: Sample app that tracks Covid-19 US stats ... (with a star icon). This tile shows a blue circular icon with a white chart and the creation date (6/25/20, 12:58:47 PM).
- Salesforce Analytics - Sales Manag...**: Analyze your Salesforce marketing perfor... Version 11. This tile shows the Salesforce logo and the creation date (12/29/22, 12:53:17 PM).
- Google Analytics template app 1/6...**: Evaluate the performance of your website ... Version 7. This tile shows a computer monitor icon with a chart and the creation date (1/6/23, 10:25:11 AM).

2. You can ask a question by typing in the Q&A box, or select a tile to open the underlying report.



Change the dashboard and report to fit the needs of your organization. Then [distribute your app to your colleagues](#)

## What's included

After connecting to a service, you see a newly created app and workspace with a dashboard, reports, and dataset. The data from the service is focused on a specific scenario and might not include all the information from the service. The data is scheduled to refresh automatically once per day. You can control the schedule by selecting the dataset.

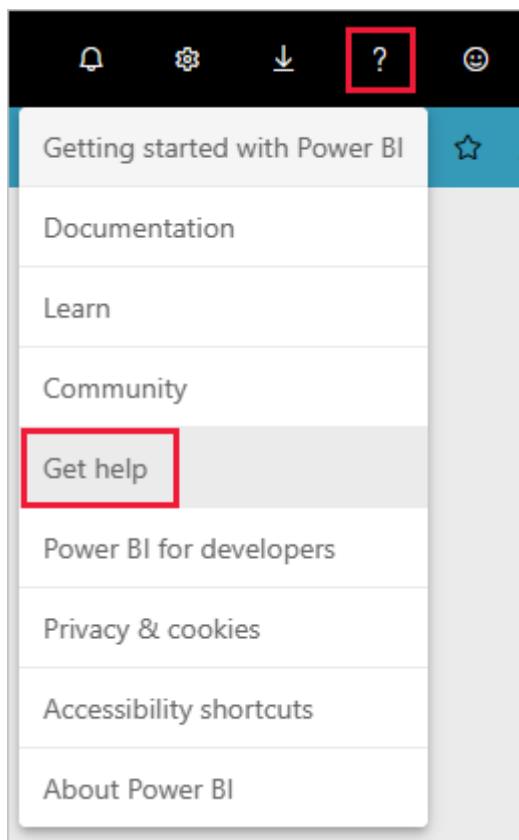
You can also [connect to many services in Power BI Desktop](#), such as Google Analytics, and create your own customized dashboards and reports.

For more information on connecting to specific services, see the individual help pages.

## Troubleshooting

**Empty tiles:** While Power BI is first connecting to the service, you might see an empty set of tiles on your dashboard. If you still see an empty dashboard after two hours, it's likely the connection failed. If you didn't see an error message with information on correcting the issue, file a support ticket.

- Select the question mark icon (?) in the upper-right corner > **Get help**.



**Missing information:** The dashboard and reports include content from the service focused on a specific scenario. If you're looking for a specific metric in the app and don't see it, add an idea on the [Power BI Support](#) page.

## Suggesting services

Do you use a service you'd like to suggest for a Power BI app? Go to the [Power BI Support](#) page and let us know.

If you're interested in creating template apps to distribute yourself, see [Create a template app in Power BI](#). Power BI partners can build Power BI apps with little or no coding, and deploy them to Power BI customers.

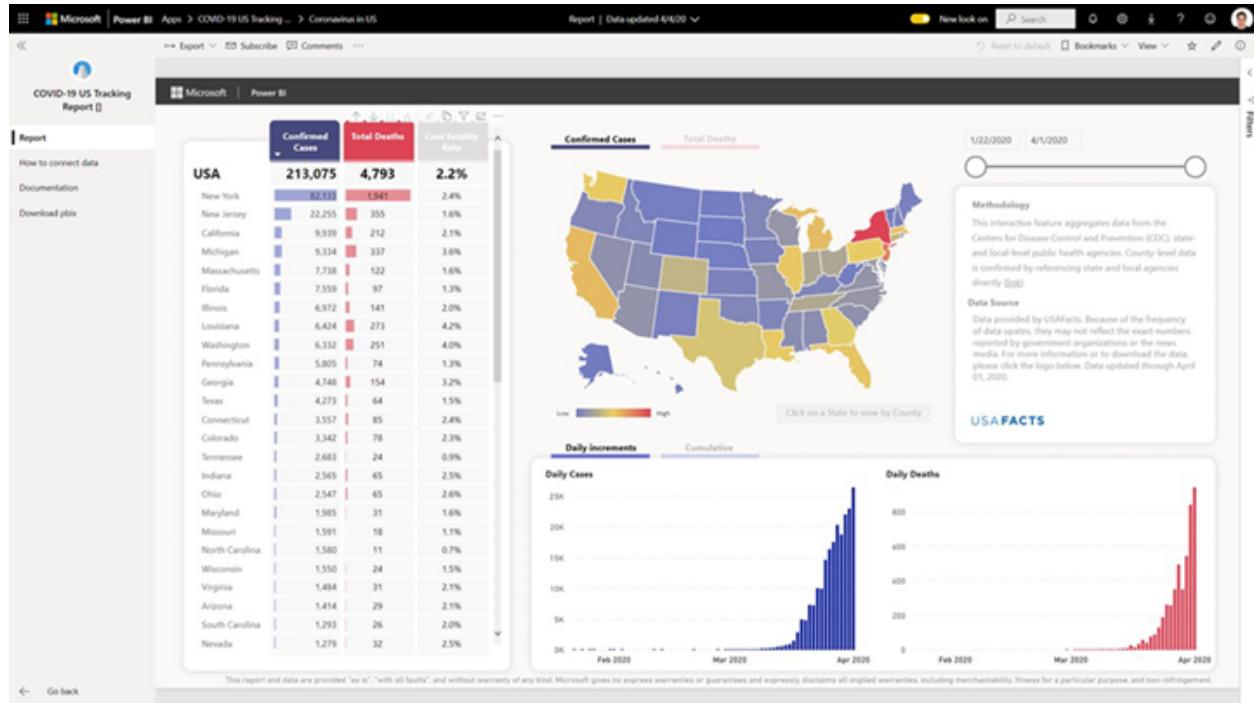
## Next steps

- [Distribute apps to your colleagues](#)
- [Create workspaces in Power BI](#)
- Questions? [Try asking the Power BI Community](#)

# Connect to the COVID-19 US tracking report

Article • 12/09/2020

This article tells you how to install the template app for the COVID-19 tracking report, and how to connect to the data sources.

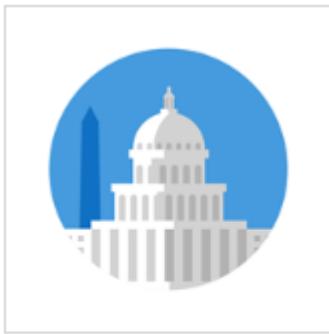


For detailed information about the report itself, including disclaimers and information about the data, see [COVID-19 tracking sample for US state and local governments](#).

After you've installed the template app and connected to the data sources, you can customize the report as per your needs. You can then distribute it as an app to colleagues in your organization.

## Install the app

1. Click the following link to get to the app: [COVID-19 US Tracking Report template app](#).
2. Once you're on the App's AppSource page, click [GET IT NOW](#).



GET IT NOW



3. When prompted , click **Install**. Once the app has installed, you will see it on your Apps page.

The screenshot shows the Microsoft Power BI Apps interface. The left sidebar includes options like Home, Favorites, Recent, Apps (which is selected), Shared with me, Deployment pipelines, Workspaces, and My workspace. The main area displays three apps: 'COVID-19 US Tracking R...' (version 23, April 5, 2020), 'Github' (version 7, March 30, 2020), and 'PowerApps Reports and ...' (version 1, March 26, 2020). A search bar at the top is labeled 'Search content...'. The overall theme is dark grey with white text.

## Connect to data sources

1. Click the icon on your Apps page to open the app. The app opens, showing sample data.
2. Select the **Connect your data** link on the banner at the top of the page.

COVID-19 US Tracking Report 11/28/2020, 3:59:16

Report

How to connect data

Documentation

Confirmed Cases

USA 213,075

Total D 4,7

3. The parameters dialog will appear. There are no required parameters. Click **Next**.

Connect to COVID-19 US Tracking Report

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

Parameters

Before connecting to your data, you must update the required parameters (\*).

There are no required parameters in this app

Advanced ▾

Next Cancel

4. The authentication method dialog will appear. Recommended values are prepopulated. Don't change these unless you have specific knowledge of different values.

Click **Next**.



## Connect to COVID-19 US Tracking Report

X

You are connecting to

url

<https://usafactsstatic.blob.core.windows.net/publ>



Authentication method

Anonymous



Privacy level setting for this data source [Learn more](#)

Public

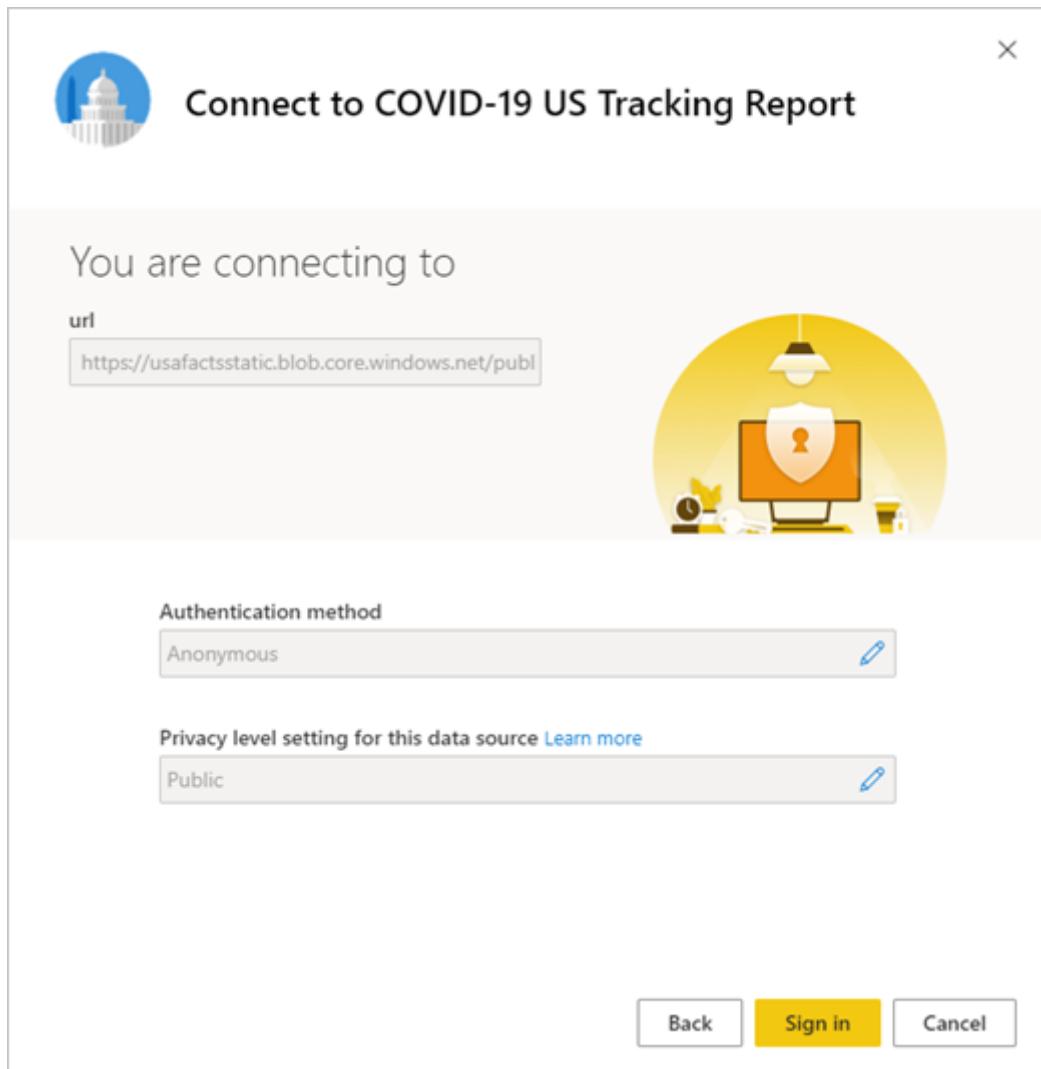


Back

Next

Cancel

5. Click Sign in.



The report will connect to the data sources and be populated with up-to-date data. During this time you will see sample data and that refresh is in progress.

## Schedule report refresh

When the data refresh has completed, you will be in the workspace associated with the app. [Set up a refresh schedule](#) to keep the report data up to date.

# Customize and share

See [Customize and share the app](#) for details. Be sure to review the [report disclaimers](#) before publishing or distributing the app.

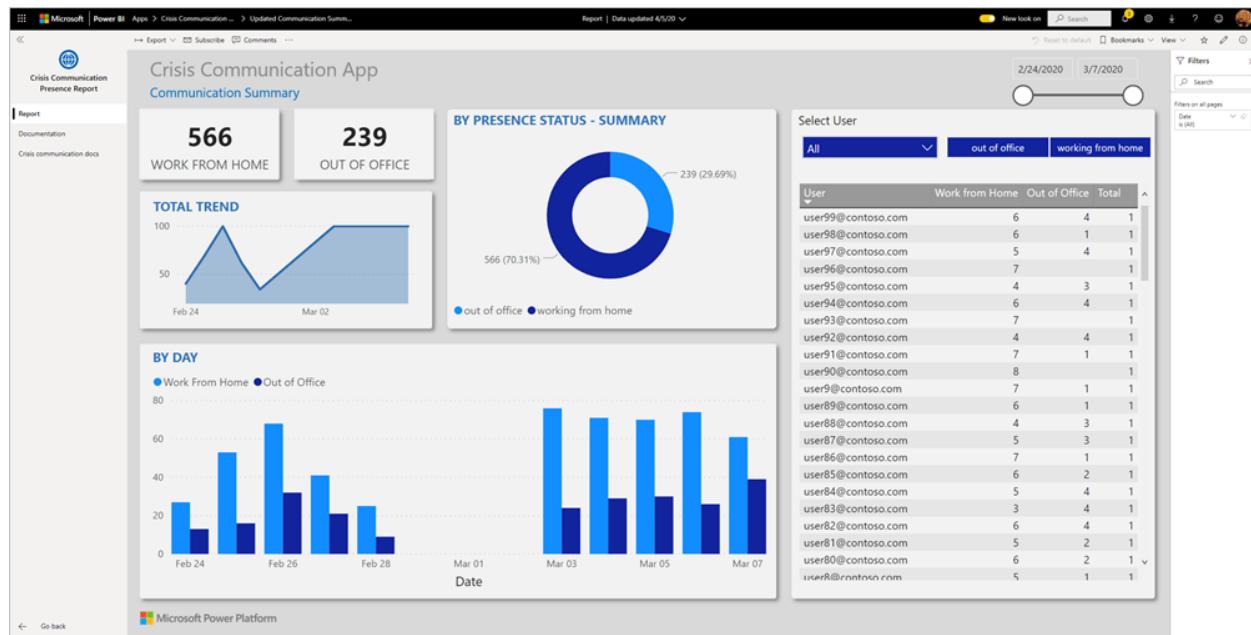
## Next steps

- [COVID-19 tracking sample for US state and local governments](#)
- Questions? [Try asking the Power BI Community](#)↗
- [What are Power BI template apps?](#)
- [Install and distribute template apps in your organization](#)

# Connect to the Crisis Communication Presence Report

Article • 12/09/2020

This Power BI app is the report/dashboard artifact in the Microsoft Power Platform solution for Crisis Communication. It tracks worker location for Crisis Communication app users. The solution combines capabilities of Power Apps, Power Automate, Teams, SharePoint and Power BI. It can be used on the web, mobile or in Teams.



The dashboard shows emergency managers aggregate data across their health system to help them to make timely, correct decisions.

This article tells you how to install the app and how to connect to the data sources. For more information about the Crisis Communication app, see [Set up and learn about the Crisis Communication sample template in Power Apps](#)

After you've installed the template app and connected to the data sources, you can customize the report as per your needs. You can then distribute it as an app to colleagues in your organization.

## Prerequisites

Before installing this template app, you must first install and set up the [Crisis Communication sample](#). Installing this solution creates the datasource references necessary to populate the app with data.

When installing the Crisis Communication sample, take note of the SharePoint list folder path of "CI\_Employee Status" and list ID.

## Install the app

1. Click the following link to get to the app: [Crisis Communication Presence Report template app ↗](#)
2. On the AppSource page for the app, select **GET IT NOW ↗**.

The screenshot shows the Microsoft AppSource interface. In the top navigation bar, 'AppSource' is highlighted. Below it, the page title is 'Crisis Communication Presence Report' by Microsoft. A large blue button labeled 'GET IT NOW' is prominently displayed. To its left is a circular icon with a globe. On the right side, there's a preview image titled 'Crisis Communication solution' showing a play button over a purple background. The main content area includes sections for 'Overview' and 'Reviews', and a detailed description of the app's features and capabilities.

3. Read the information in **One more thing**, and select **Continue**.

This screenshot shows a consent dialog box. It features a 'X' icon in the top right corner. The heading 'One more thing ...' is centered above the app logo and name. The app logo is a globe icon, and the name is 'Crisis Communication Presence Report' by Microsoft. Below the logo, a paragraph of text discusses data sharing and privacy policies. At the bottom, it shows the user is signed in as 'Megan Bowen (MeganB@M365x634194.OnMicrosoft.com)' and contains a large blue 'Continue' button.

4. Select **Install**.

Install this Power BI app?

X



Crisis Communication Presence Report

by Microsoft

[View in AppSource](#)

Apps may contain security or privacy risks. Only install apps from trusted authors and sources.

[Learn more](#)

By installing this app I agree to the publisher's [privacy policy](#) and [terms of service](#)

[Install](#)

[Cancel](#)

Once the app has installed, you see it on your Apps page.

The screenshot shows the Microsoft Power BI Apps interface. On the left, there is a navigation sidebar with options like Home, Favorites, Recent, Apps (which is currently selected), Shared with me, Deployment pipelines, Workspaces, and My workspace. The main area displays a grid of app cards. One card for 'COVID-19 US Tracking R...' is partially visible on the left. In the center, the 'Crisis Communication Pr...' app card is highlighted with a red border. To its right, another card for 'Hospital Emergency Res...' is visible. At the bottom of the screen, there are three colored circular progress bars (yellow, teal, and green).

## Connect to data sources

1. Select the icon on your Apps page to open the app.

The app opens, showing sample data.

2. Select the **Connect your data** link on the banner at the top of the page.

The screenshot shows the 'Crisis Communication Presence Report' app page. At the top, there is a banner with the text 'You're viewing this app with sample data' and a blue link labeled 'Connect your data'. Below the banner, the title 'Crisis Communication App' and subtitle 'Communication Summary' are displayed. On the left, there is a sidebar with 'Report' and 'Documentation' options. In the center, there are two large numerical values: '566' and '239'. The overall layout is clean and modern, designed for easy data visualization.

3. In the dialog box:

- a. In the SharePoint\_Folder field, enter your "CI\_Employee Status" SharePoint list path.
- b. In the List\_ID field, enter your list ID that you got from list settings. When done, click **Next**.



## Connect to Crisis Communication Presence Report

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

**Parameters**  
Before connecting to your data, you must update the required parameters (\*).

**SharePoint\_Folder \***  
Please enter the solution installation sharpoint folder.

**List\_ID \***  
Please enter the list ID as described at solution setup documentation:  
<https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/sample-crisis-communication-app#monitor-office-absences-with-power-bi>

**Next** **Cancel**

4. In the next dialog that appears, set the authentication method to **OAuth2**. You don't have to do anything to the privacy level setting.

Select **Sign in**.



## Connect to Crisis Communication Presence Report

X

sharePointSiteUrl

Authentication method

Privacy level setting for this data source

5. At the Microsoft sign-in screen, sign in to Power BI.



Microsoft

Pick an account

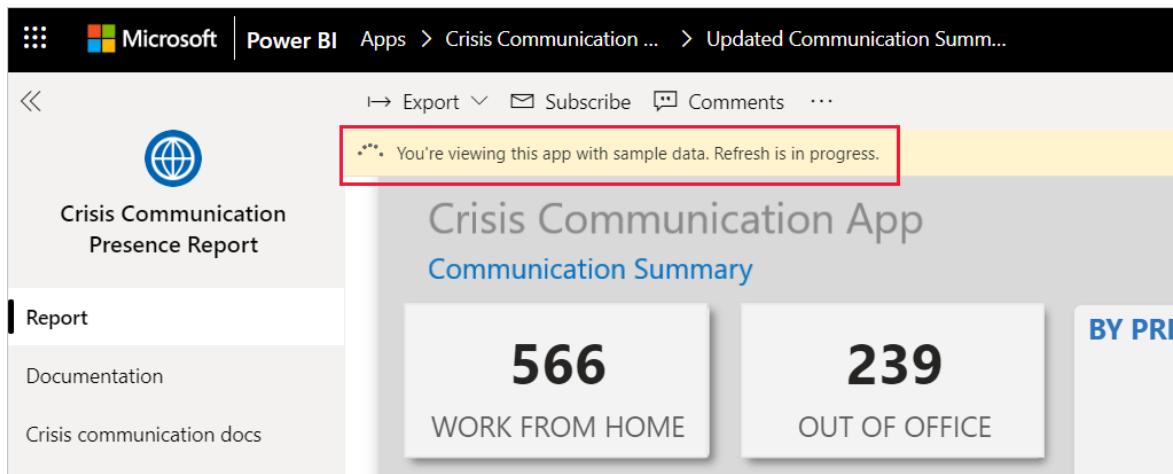


Megan Bowen  
meganb@contoso.com  
Connected to Windows



Use another account

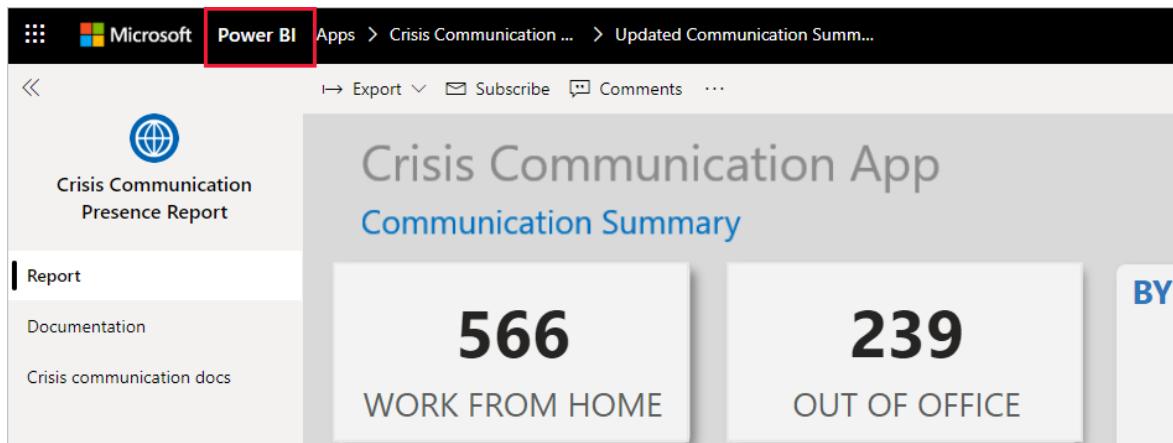
After you've signed in, the report connects to the data sources and is populated with up-to-date data. During this time, the activity monitor turns.



## Schedule report refresh

When the data refresh has completed, [set up a refresh schedule](#) to keep the report data up to date.

1. In the top header bar, select **Power BI**.



2. In the left navigation pane, look for the Hospital Emergency Response Decision Support Dashboard workspace under **Workspaces**, and follow the instruction described in the [Configure scheduled refresh](#) article.

## Customize and share

See [Customize and share the app](#) for details. Be sure to review the [report disclaimers](#) before publishing or distributing the app.

## Next steps

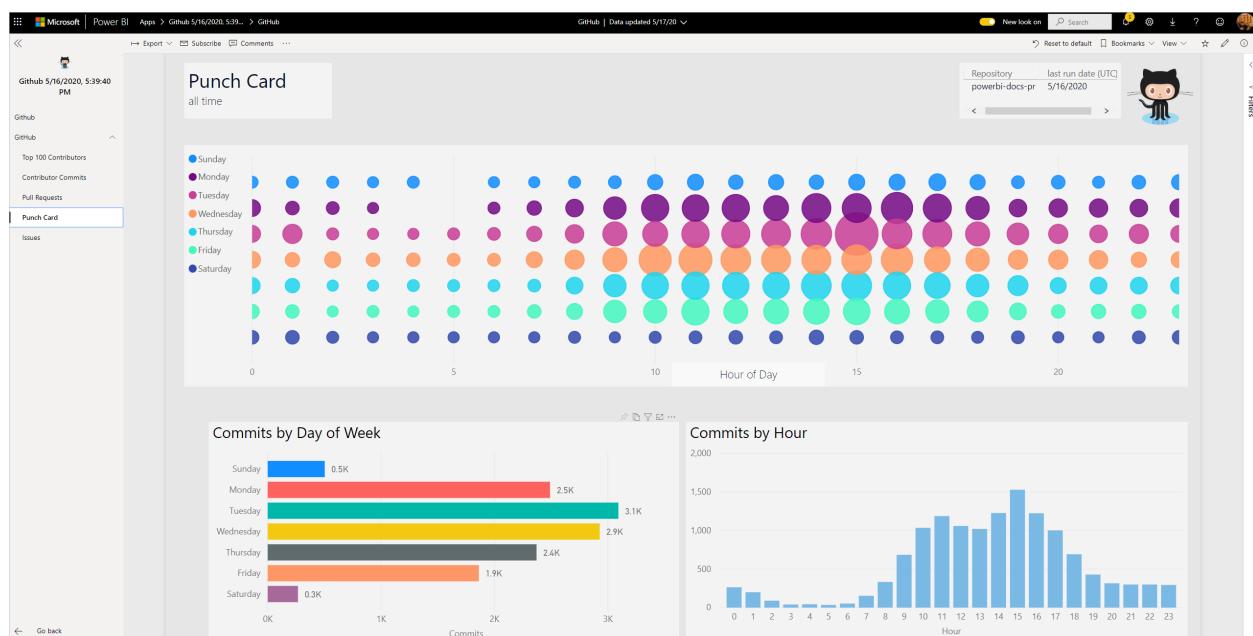
- [Set up and learn about the Crisis Communication sample template in Power Apps](#)
- Questions? [Try asking the Power BI Community](#) ↗

- What are Power BI template apps?
- Install and distribute template apps in your organization

# Connect to GitHub with Power BI

Article • 09/19/2022

This article walks you through pulling your data from your GitHub account with a Power BI template app. The template app generates a workspace with a dashboard, a set of reports, and a dataset to allow you to explore your GitHub data. The GitHub app for Power BI shows you insights into your GitHub repository, also known as repo, with data around contributions, issues, pull requests, and active users.



After you've installed the template app, you can change the dashboard and report. Then you can distribute it as an app to colleagues in your organization.

Connect to the [GitHub template app](#) or read more about the [GitHub integration](#) with Power BI.

You can also try the [GitHub tutorial](#). It installs real GitHub data about the public repo for the Power BI documentation.

## Note

This template app requires the GitHub account to have access to the repo. More details on requirements below.

This template app does not support GitHub Enterprise.

## Install the app

1. Click the following link to get to the app: [GitHub template app](#)

2. On the AppSource page for the app, select **GET IT NOW**.

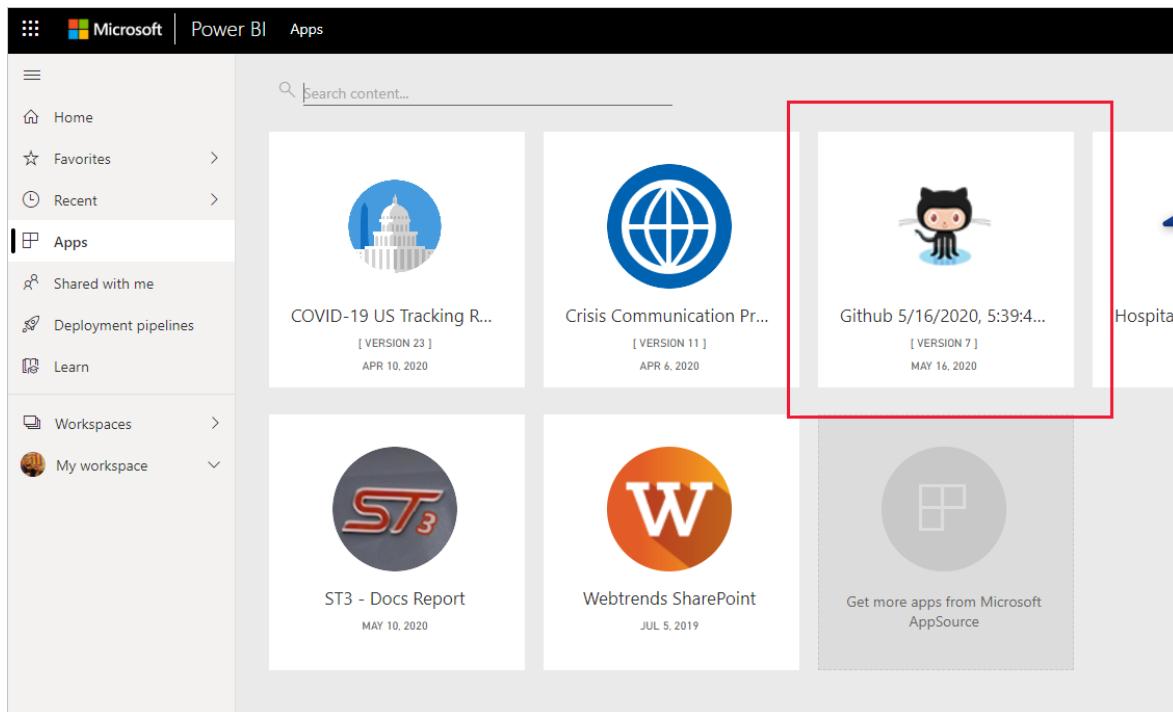
The screenshot shows the Microsoft AppSource interface. In the top navigation bar, 'Microsoft' is on the left, followed by 'AppSource Apps' and a search bar containing 'Search Microsoft AppSource'. To the right are 'More', a heart icon, a smiley face icon, and a user profile icon. Below the navigation, the path 'Apps > GitHub' is shown. The main content area features a large image of the GitHub logo (a black cat) and the title 'Github'. Below the title are the Microsoft logo, a rating of 3.4 (20 reviews), and tabs for 'Overview' and 'Reviews'. A prominent blue button labeled 'GET IT NOW' is highlighted with a red border. To the right of the button is a detailed description of the app's functionality, mentioning repository activity visualization. Further down, there is a screenshot of the Power BI dashboard generated by the app, displaying various metrics like 'Pull Requests by Month' and 'Contributors with open issues'. On the left side of the main content, there is a sidebar with links for 'Pricing Free', 'Products Power BI apps', 'Publisher Microsoft', 'Acquire Using Work or school account', 'Categories Analytics', 'Support Support Help', and 'Legal License Agreement Privacy Policy'.



3. Select **Install**.

The dialog box has a white background. At the top, it asks 'Install this Power BI app?'. Below this is a small GitHub logo. The app title is 'Github Repository Usage' by Microsoft, with a 'View in AppSource' link underneath. A note at the bottom says 'Apps may contain security or privacy risks. Only install apps from trusted authors and sources.' with a 'Learn more' link. Below that, a statement says 'By installing this app I agree to the publisher's [privacy policy](#) and [terms of service](#)'. At the bottom right are two buttons: a yellow 'Install' button and a white 'Cancel' button.

Once the app has installed, you see it on your Apps page.

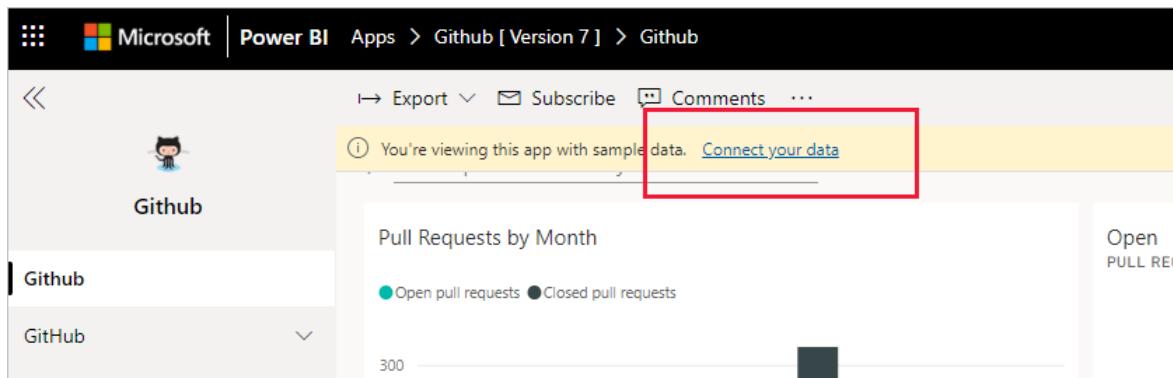


## Connect to data sources

1. Select the icon on your Apps page to open the app.

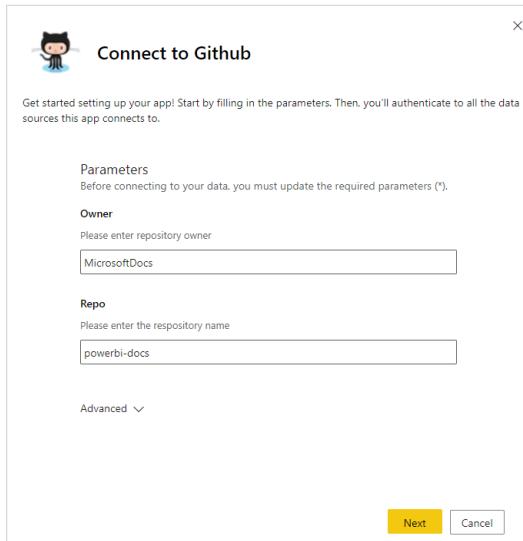
The app opens, showing sample data.

2. Select the **Connect your data** link on the banner at the top of the page.

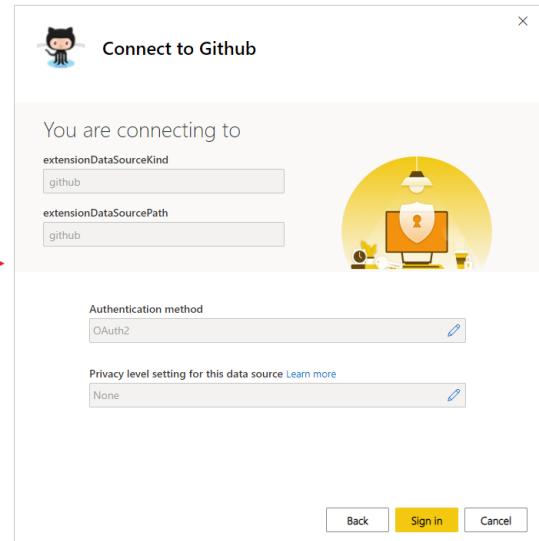


3. This opens the parameters dialog, where you change the data source from the sample data to your own data source (see [known limitations](#)), followed by the authentication method dialog. You may have to redefine the values in these dialogs.

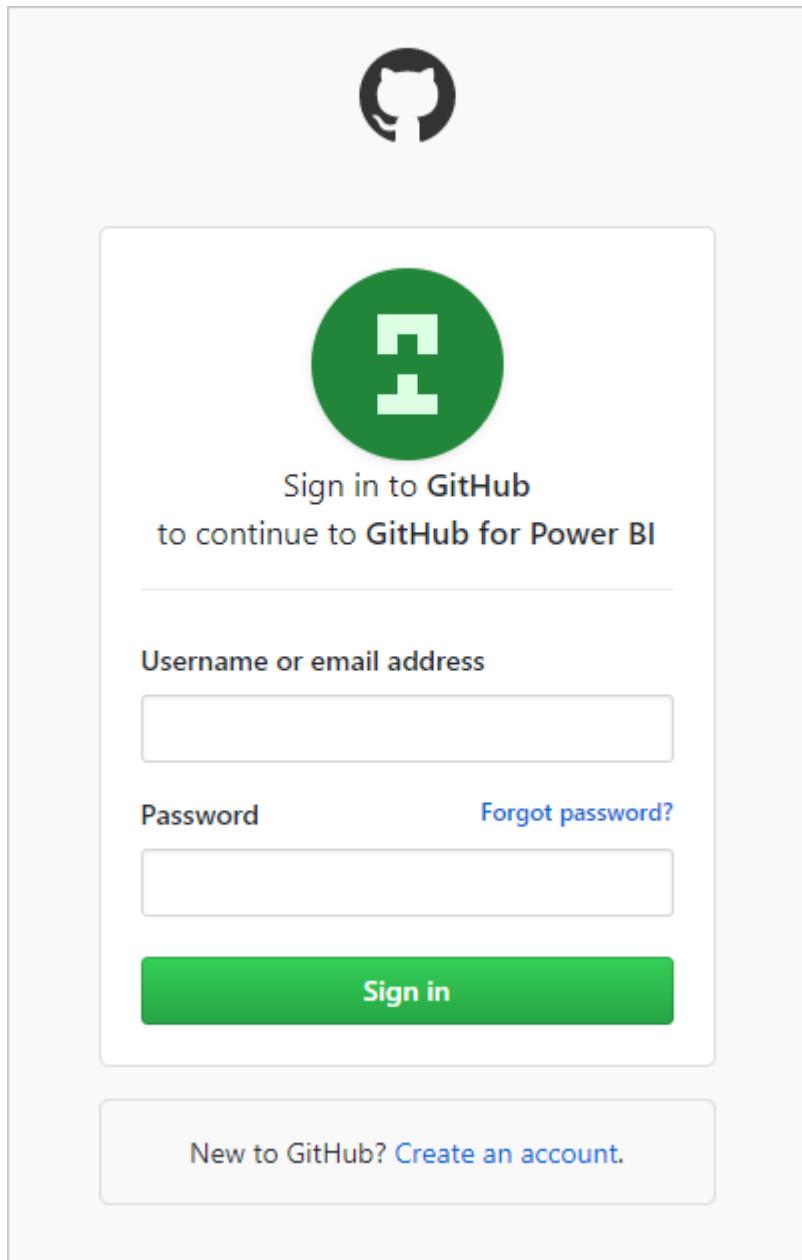
Parameters dialog



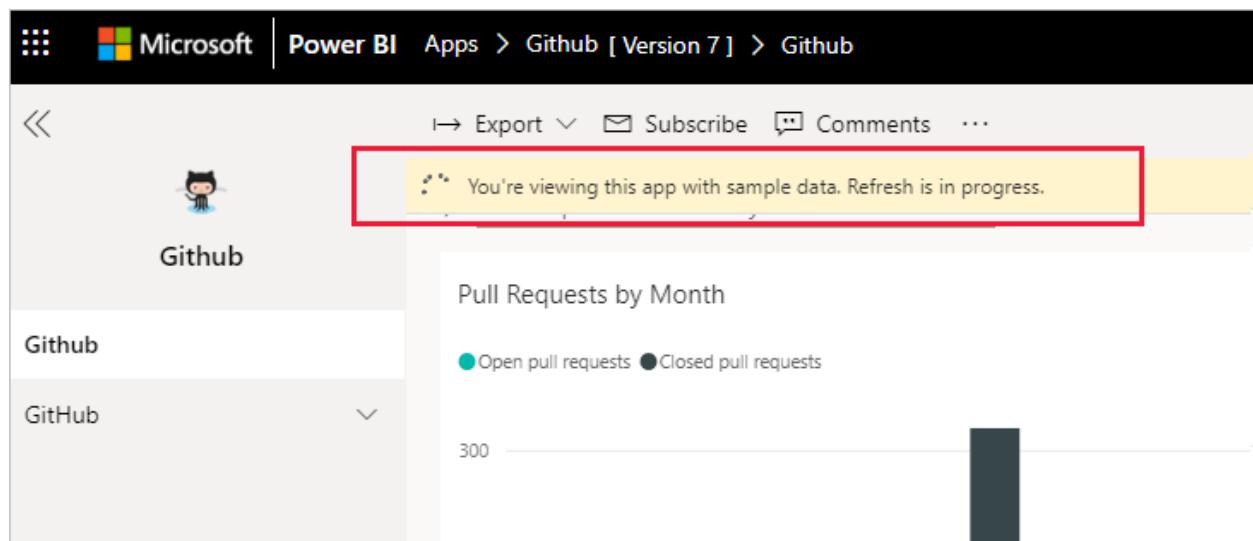
Authentication dialog



4. Enter your GitHub credentials and follow the GitHub authentication process (this step might be skipped if you're already signed in with your browser).



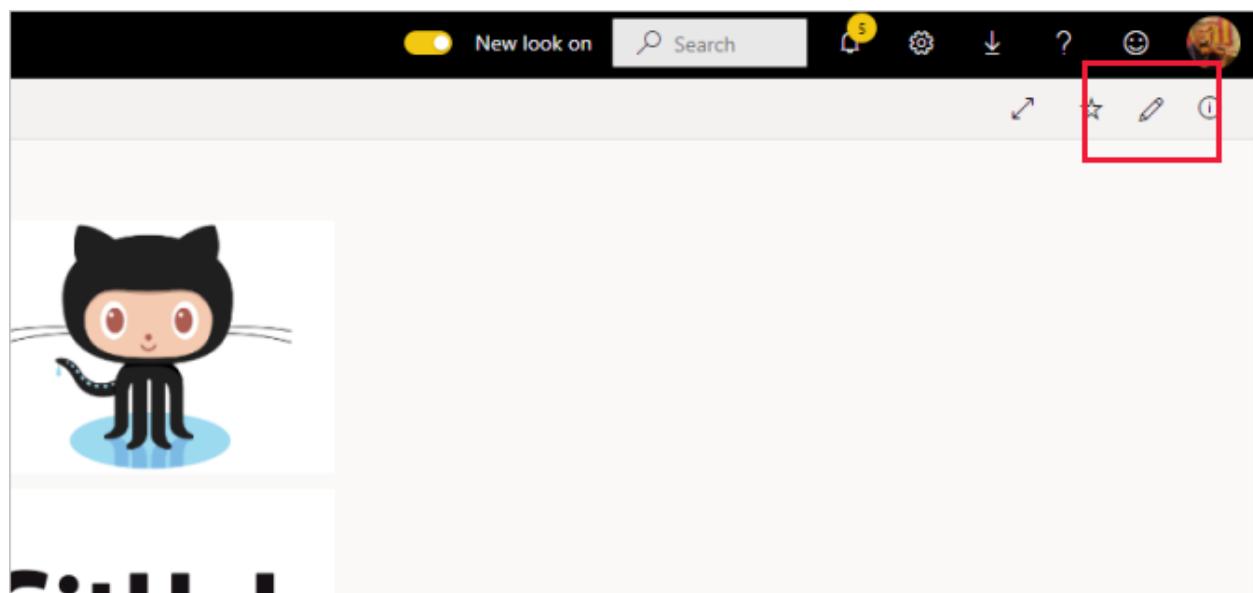
Once you've finished filling out the connection dialogs and signed in to GitHub, the connection process starts. A banner informs you that the data is being refreshed, and that in the meantime you are viewing sample data.



Your report data will automatically refresh once a day, unless you disabled this during the sign-in process. You can also [set up your own refresh schedule](#) to keep the report data up to date if you so desire.

## Customize and share

To customize and share your app, select the pencil icon at the top right corner of the page.



For information about editing items in the workspace, see

- [Tour the report editor in Power BI](#)
- [Basic concepts for designers in the Power BI service](#)

Once you are done making any changes you wish to the items in the workspace, you are ready to publish and share the app. See [Create and publish your app](#) to learn how to do this.

## What's included in the app

The following data is available from GitHub in Power BI:

Table name	Description
Contributions	The contributions table gives the total additions, deletions, and commits authored by the contributor aggregated per week. The top 100 contributors are included.
Issues	List all issues for the selected repo and it contains calculations like total and average time to close an issue, Total open issues, Total closed issues. This table will be empty when there are no issues in the repo.
Pull requests	This table contains all the Pull Requests for the repo and who pulled the request. It also contains calculations around how many open, closed, and total pull requests, how long it took to pull the requests and how long the average pull request took. This table will be empty when there are no pull requests in the repo.
Users	This table provides a list of GitHub users or contributors who have made contributions, filed issues, or solved Pull requests for the repo selected.
Milestones	It has all the Milestones for the selected repo.
DateTable	This table contains dates from today and for years in the past that allow you to analyze your GitHub data by date.
ContributionPunchCard	This table can be used as a contribution punch card for the selected repo. It shows commits by day of week and hour of day. This table is not connected to other tables in the model.
RepoDetails	This table provides details for the repo selected.

## System requirements

- The GitHub account that has access to the repo.
- Permission granted to the Power BI for GitHub app during first login. See details below on revoking access.
- Sufficient API calls available to pull and refresh the data.

## ⓘ Note

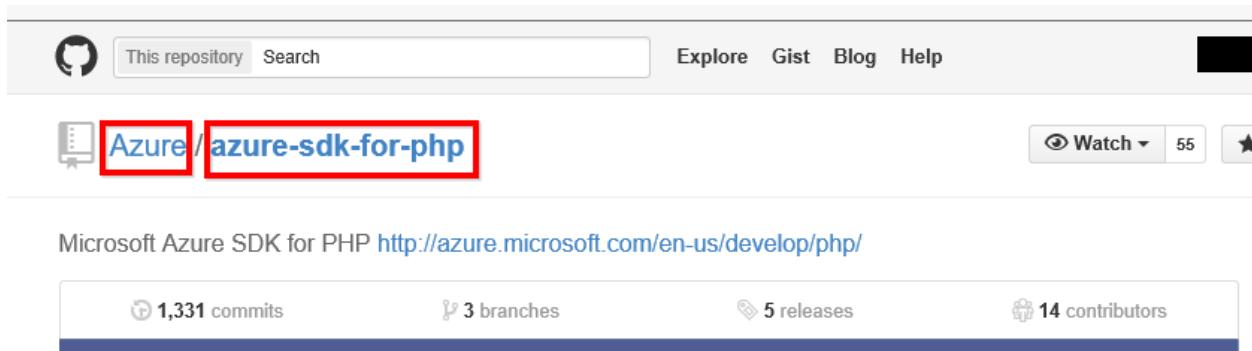
This template app does not support GitHub Enterprise.

## De-authorize Power BI

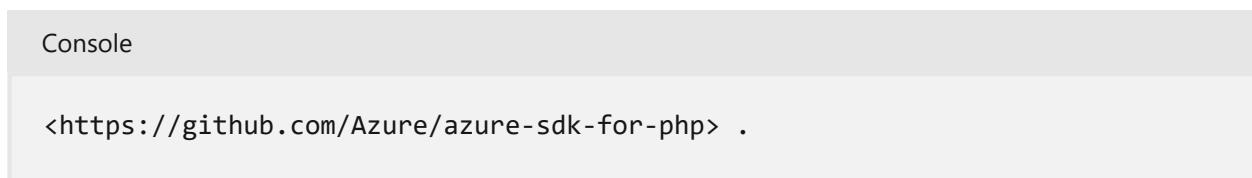
To de-authorize Power BI from being connected to your GitHub repo, you can Revoke access in GitHub. See this [GitHub help](#) topic for details.

## Finding parameters

You can determine the owner and repository by looking at the repository in GitHub itself:



The first part "Azure" is the owner and the second part "azure-sdk-for-php" is the repository itself. You see these same two items in the URL of the repository:



## Troubleshooting

If necessary, you can verify your GitHub credentials.

1. In another browser window, go to the GitHub web site and sign in to GitHub. You can see you're logged in, in the upper-right corner of the GitHub site.
2. In GitHub, navigate to the URL of the repo you plan to access in Power BI. For example: <https://github.com/dotnet/corefx>.
3. Back in Power BI, try connecting to GitHub. In the Configure GitHub dialog box, use the names of the repo and repo owner for that same repo.

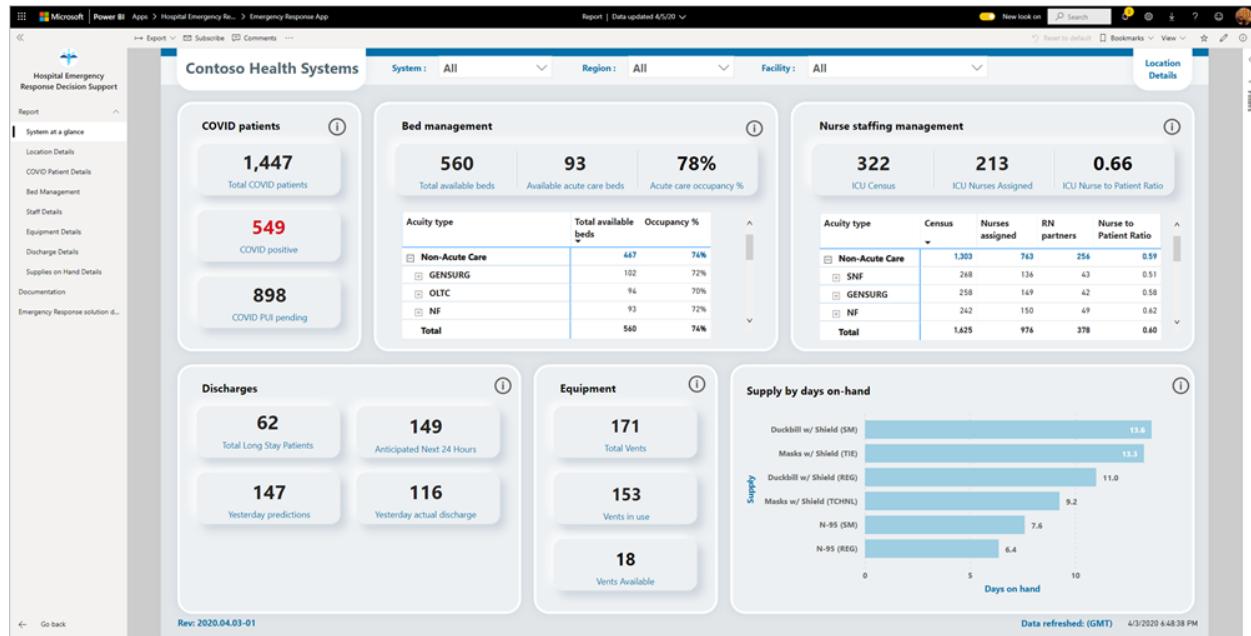
# Next steps

- Tutorial: [Connect to a GitHub repo with Power BI](#)
- Create workspaces in Power BI
- Install and use apps in Power BI
- [Connect to Power BI apps for external services](#)
- Questions? [Try asking the Power BI Community](#) ↗

# Connect to the Hospital Emergency Response Decision Support Dashboard

Article • 11/30/2020

The Hospital Emergency Response Decision Support Dashboard template app is the reporting component of the [Microsoft Power Platform solution for healthcare emergency response](#). The dashboard shows emergency managers aggregate data across their health system to help them to make timely, correct decisions.



This article tells you how to install the app and how to connect to the data sources. To learn how to use the report that you will see with this app, see the [Hospital Emergency Response Decision Support Dashboard documentation](#).

After you've installed the template app and connected to the data sources, you can customize the report as per your needs. You can then distribute it as an app to colleagues in your organization.

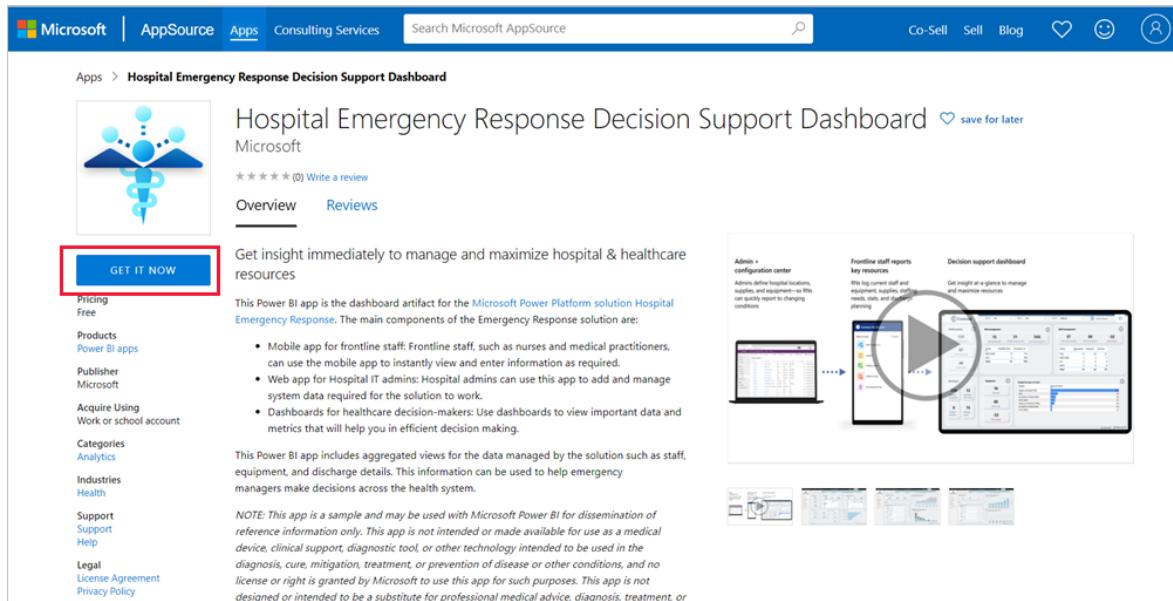
## Prerequisites

Before installing this template app, you must first install and set up the [Hospital Emergency Response Power Platform solution](#). Installing this solution creates the datasource references necessary to populate the app with data.

When installing Hospital Emergency Response Power Platform solution, take note of the [URL of your Common Data Service environment instance](#). You will need it to connect the template app to the data.

# Install the app

1. Click the following link to get to the app: [Hospital Emergency Response Decision Support Dashboard template app ↗](#)
2. On the AppSource page for the app, select [GET IT NOW ↗](#).



Hospital Emergency Response Decision Support Dashboard Microsoft

Overview    Reviews

Get insight immediately to manage and maximize hospital & healthcare resources

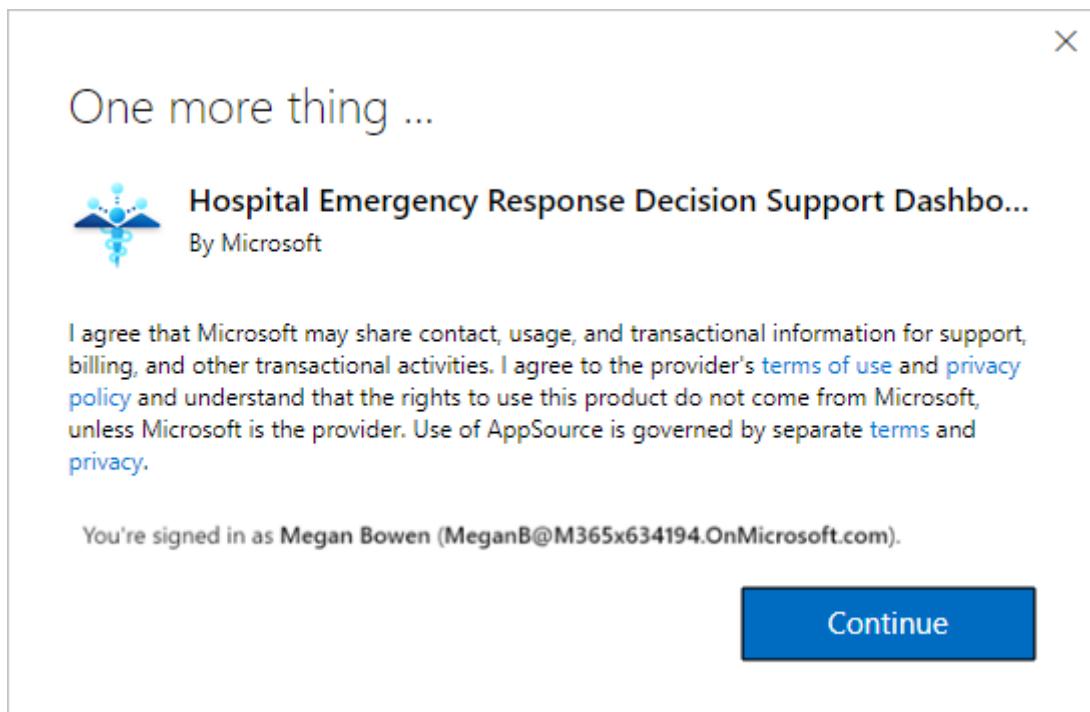
This Power BI app is the dashboard artifact for the [Microsoft Power Platform solution Hospital Emergency Response](#). The main components of the Emergency Response solution are:

- Mobile app for frontline staff: Frontline staff, such as nurses and medical practitioners, can use the mobile app to instantly view and enter information as required.
- Web app for Hospital IT admins: Hospital admins can use this app to add and manage system data required for the solution to work.
- Dashboards for healthcare decision-makers: Use dashboards to view important data and metrics that will help you in efficient decision making.

This Power BI app includes aggregated views for the data managed by the solution such as staff, equipment, and discharge details. This information can be used to help emergency managers make decisions across the health system.

NOTE: This app is a sample and may be used with Microsoft Power BI for dissemination of reference information only. This app is not intended or made available for use as a medical device, clinical support, diagnostic tool, or other technology intended to be used in the diagnosis, cure, mitigation, treatment, or prevention of disease or other conditions, and no license or right is granted by Microsoft to use this app for such purposes. This app is not designed or intended to be a substitute for professional medical advice, diagnosis, treatment, or

3. Read the information in **One more thing**, and select **Continue**.



One more thing ...

 Hospital Emergency Response Decision Support Dashbo...  
By Microsoft

I agree that Microsoft may share contact, usage, and transactional information for support, billing, and other transactional activities. I agree to the provider's [terms of use](#) and [privacy policy](#) and understand that the rights to use this product do not come from Microsoft, unless Microsoft is the provider. Use of AppSource is governed by separate [terms](#) and [privacy](#).

You're signed in as Megan Bowen (MeganB@M365x634194.OnMicrosoft.com).

Continue

4. Select **Install**.

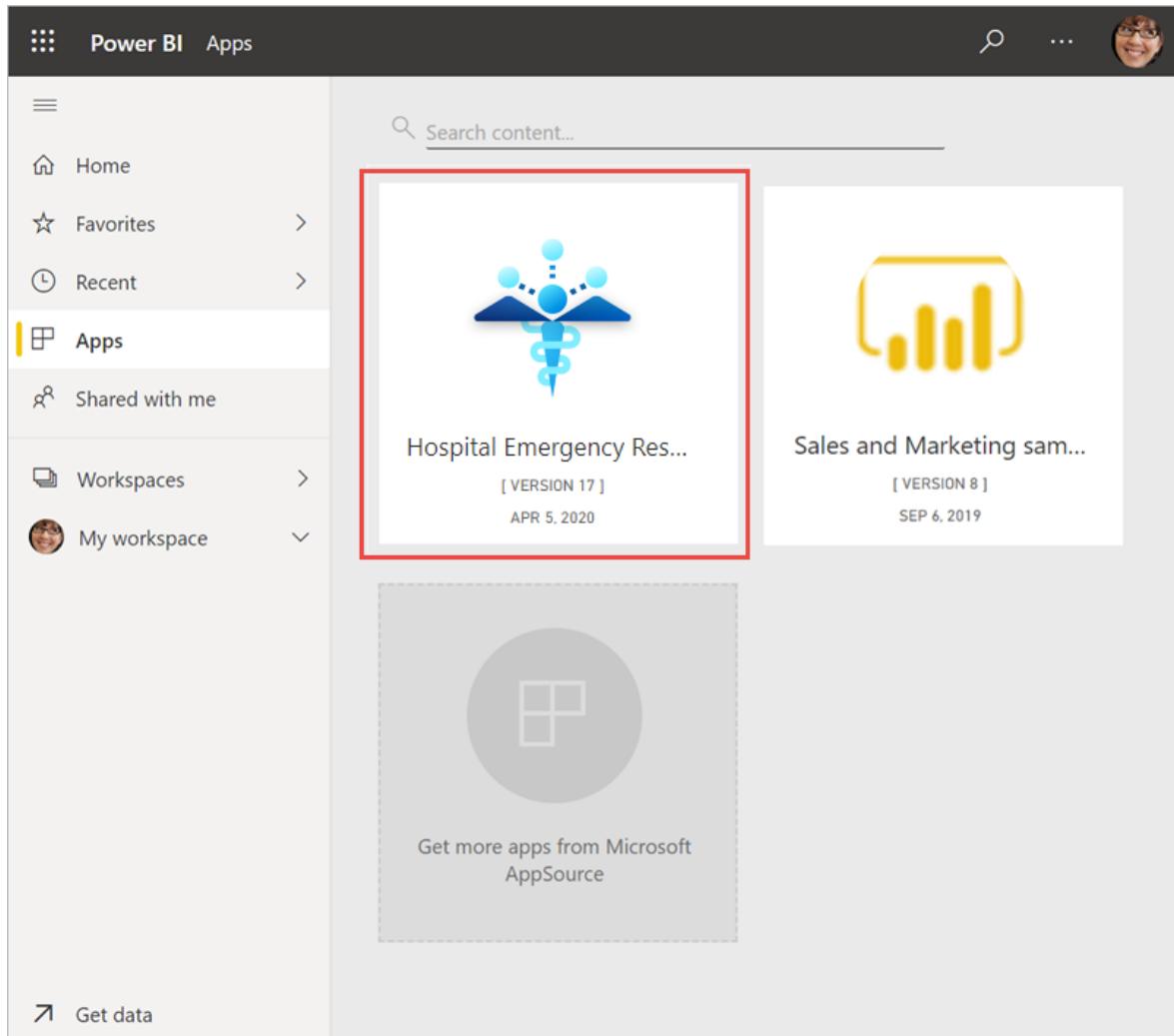
## Install this Power BI app?

Apps may contain security or privacy risks. Only install apps from trusted authors and sources.  
[Learn more](#)

**Install**

**Cancel**

Once the app has installed, you see it on your Apps page.



## Connect to data sources

1. Select the icon on your Apps page to open the app.
2. On the splash screen, select **Explore**.

## Get started with your new app

Explore your app with sample data, go to the workspace to customize as needed and share with your organization, or connect your data to get up and running.

**Connect your data**

Connect to a data source to view your new app with your own data.



**Explore with sample data**

Open your new app to start exploring with sample data.



**Customize and share**

Your app comes with a workspace, so you can customize and share it, just like an app you built yourself.



**Connect**

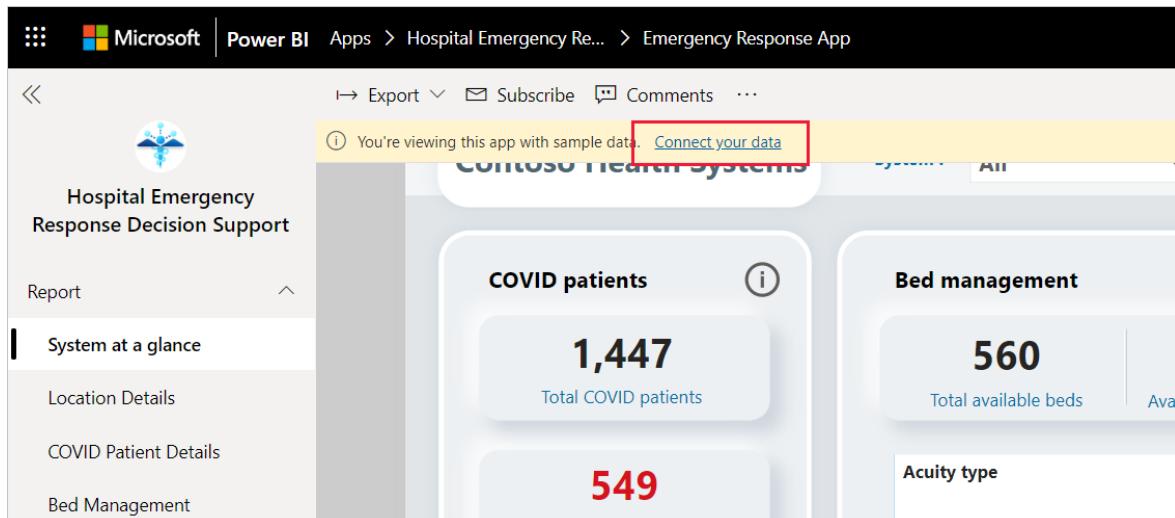
**Explore app**

**Edit workspace**

[Don't show this again](#)

The app opens, showing sample data.

3. Select the **Connect your data** link on the banner at the top of the page.



4. In the dialog box:

- In the organization name field, enter the name of your organization, for example, "Contoso Health Systems". This field is optional. This name appears in the upper-left side of the dashboard.
- In the CDS\_base\_solution field, Type the [URL of your Common Data Service environment instance](#). For example: [https://\[myenv\].crm.dynamics.com](https://[myenv].crm.dynamics.com). When done, click **Next**.



## Connect to Hospital Emergency Response Decision Support Dashboard

X

Before connecting to your data, you must update the required parameters (\*).

### Organization\_name

This is the Organization name that will be populated on the Upper Left of each report page - you can optionally edit the report and replace the text field with an image (logo) file of the organization/hospital.

Enter organization name

### CDS\_base\_solution\_URL

Please enter your organization's Emergency Response App URL (CDS Database) : Example:  
<https://COVID.crm.dynamics.com>

https://installation\_path.crm.dynamics.com

\*

Next

Cancel

5. In the next dialog that appears, set the authentication method to **OAuth2**. You don't have to do anything to the privacy level setting.

Select Sign in.



## Connect to Hospital Emergency Response Decision Support Dashboard

X

url

<https://covid.crm.dynamics.com/api/data/v9.1>

Authentication method

OAuth2

Privacy level setting for this data source

None

Sign in

Cancel

6. At the Microsoft sign-in screen, sign in to Power BI.



## Pick an account



Megan Bowen  
meganb@contoso.com  
Connected to Windows



Use another account

After you've signed in, the report connects to the data sources and is populated with up-to-date data. During this time, the activity monitor turns.

The screenshot shows the Microsoft Power BI interface. At the top, it says "Microsoft Power BI Apps > Hospital Emergency Re... > Emergency Response App". Below the header, there's a navigation bar with icons for back, forward, export, subscribe, comments, and more. A message box states "You're viewing this app with sample data. Refresh is in progress." A red box highlights this message. On the left, a sidebar titled "Report" lists "System at a glance", "Location Details", and "COVID Patient Details". The main area displays a dashboard for "Contoso Health Systems" with a card for "COVID patients" showing "1,447 Total COVID patients". Another card for "Bed manag" is partially visible.

## Schedule report refresh

When the data refresh has completed, [set up a refresh schedule](#) to keep the report data up to date.

1. In the top header bar, select Power BI.



2. In the left navigation pane, look for the Hospital Emergency Response Decision Support Dashboard workspace under **Workspaces**, and follow the instructions described in the [Configure scheduled refresh](#) article.

## Customize and share

See [Customize and share the app](#) for details. Be sure to review the [report disclaimers](#) before publishing or distributing the app.

## Next steps

- [Understanding the Hospital Emergency Response report](#)
- [Set up and learn about the Crisis Communication sample template in Power Apps](#)
- Questions? [Try asking the Power BI Community](#) ↗
- [What are Power BI template apps?](#)
- [Install and distribute template apps in your organization](#)

# Connect to the Emissions Impact Dashboard for Azure

Article • 04/08/2023

Calculate your cloud-based carbon emissions today with the Emissions Impact Dashboard for Azure.

Accurate carbon accounting requires good information from partners, vendors, and suppliers. The Emissions Impact Dashboard for Azure gives you transparency on the carbon emissions generated by your usage of Azure and Microsoft Dynamics.

Microsoft's carbon accounting extends across all three scopes of emissions with a methodology validated by Stanford University in 2018. It uses consistent and accurate carbon accounting to quantify the impact of Microsoft cloud services on customers' environmental footprint. Microsoft is the only cloud provider to provide this level of transparency to customers while compiling reports for voluntary or statutory reporting requirements.

## Prerequisites

To install the Emissions Impact Dashboard for Azure in Power BI and connect it to your data, make sure you have the following before installing the app:

- A Power BI Pro license. If you don't have a Power BI Pro license, [get a free trial now](#).
- An admin role with read/write permissions on your Azure tenant. If you don't have these permissions, contact an Azure admin who has read/write permissions.

The Emissions Impact Dashboard for Azure is supported for EA Direct, MCA, and MPA accounts with direct billing relationships with Microsoft.

- If you have an EA Direct account, you must be a Billing Account Administrator (formerly known as an Enrollment Administrator) with either read or write permissions and have your company's [billing account ID](#) (formerly known as the enrollment number).
- If you have an MCA or MPA and direct billing relationship with Microsoft, you must be a Billing Account Administrator with a role as Billing Account Reader/Contributor/Owner and have your company's [billing account ID](#).

## ⓘ Important

Cloud solution providers (CSPs) are supported. Customers who purchase Azure from a CSP aren't supported and must work directly with their CSP partner to learn about their cloud emissions. Legacy accounts and China enrollments aren't supported.

## Install the app

1. Select the following link to get to the app: [Emissions Impact Dashboard template app ↗](#).
2. On the AppSource page for the app, select **GET IT NOW**.

The screenshot shows the Microsoft AppSource page for the "Emissions Impact Dashboard". The page includes the following elements:

- Header:** Microsoft | AppSource More ▾, Search, Help, ?
- Breadcrumb:** Apps > Emissions Impact Dashboard
- App Card:** Emissions Impact Dashboard by Microsoft Corporation (Power BI apps). A red box highlights the "Get it now" button.
- Overview Tab:** Overview (selected), Ratings + reviews, Details + support
- Emissions Impact Dashboard Section:** Calculate your carbon emissions from cloud-based computing. It describes the app's purpose: providing transparency into greenhouse gas emissions associated with cloud usage. It also lists requirements for getting started, which include having an EA enrollment number or Billing Account ID with Admin access, and connecting to the Power BI Pro trial.
- Data for your organization's carbon accounting Section:** This section lists benefits such as gaining visibility, tracking emissions changes, measuring environmental impact, and calculating how to reduce emissions.
- Methodology Note:** Microsoft's emissions reports are determined via a methodology validated by Stanford University in 2018, aligning with ISO standards for measuring greenhouse gas emissions.
- Get access now Section:** This section provides instructions for accessing the app, mentioning credential validation and the Power BI Pro trial.
- At a glance:** A grid of four screenshots showing different views of the Power BI dashboard, including charts and data tables.

You can also search for the app in Power BI.

3. When prompted, select **Install**.

## Install this Power BI app?



Emissions Impact Dashboard  
by Microsoft Corporation  
[View in AppSource](#)

Apps may contain security or privacy risks. Only install apps from trusted authors and sources.  
[Learn more](#)

By installing this app I agree to the publisher's [privacy policy](#) and [terms of service](#)

**Install**

**Cancel**

- When the app finishes installing, it will appear on your Power BI Apps page. Select the app and open it.

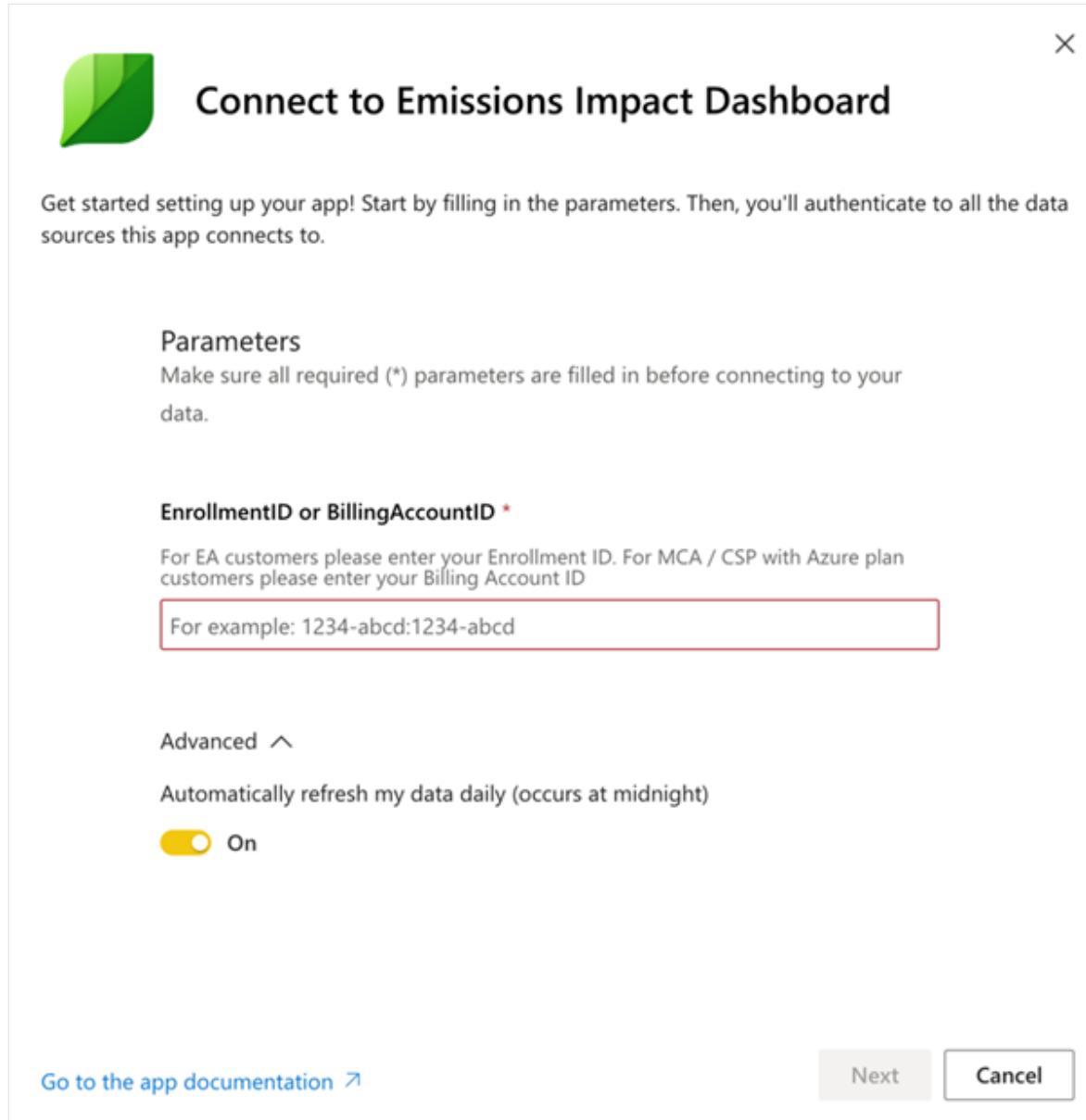
The screenshot shows the Microsoft Power BI Apps interface. On the left, there is a sidebar with various navigation options: Home, Favorites, Recent, Create, Datasets, Goals, Apps (which is highlighted with a red box), Shared with me, Deployment pipelines, Learn, Workspaces (with My workspace dropdown), and Log out. The main area is titled 'Apps' and contains a sub-instruction: 'Apps are collections of dashboards and reports in one easy-to-find place.' Below this, a table lists several apps, each with a small icon, name, publisher (redacted), published date, and app type. One row, 'Emissions Impact Dashboard', is highlighted with a red box. The table columns are Name, Publisher, Published, and App type.

Name	Publisher	Published	App type
Github	[Redacted]	5/23/2021, 12:00:25 PM	Template app
Analyze Popular Stocks with Power BI	[Redacted]	5/21/2021, 6:30:51 PM	Template app
Inventory Tracker	[Redacted]	6/13/2021, 11:51:10 PM	Template app
Premium Capacity Utilization And Metrics	[Redacted]	8/25/2021, 10:55:53 AM	Template app
<b>Emissions Impact Dashboard</b>	[Redacted]	10/1/2021, 9:37:55 AM	Template app

- Select Connect your data.

The screenshot shows the 'Emissions Impact Dashboard' app interface. At the top, there is a header bar with the Microsoft logo, 'Power BI', and the dashboard title 'Emissions Impact Dashboard 10/1/2021, 9:37:55 AM'. Below the header, there is a toolbar with File, Export, Chat in Teams, Get insights, Subscribe, and other options. A yellow banner at the top states 'You're viewing this app with sample data. [Connect your data](#)'. The main content area has tabs for Dashboard, Emissions details, Emissions savings, and GHG Preparation. The 'Dashboard' tab is selected. It displays the message 'Microsoft carbon emissions from my company cloud usage'. On the left, there is a sidebar with a link to 'Microsoft Sustainability Calcul...' and a 'Subscription name' dropdown set to 'All'. On the right, there is a section titled 'Microsoft carbon emissions from my company usage : scope 1, 2 and 3 (MTCO2e)' with a 'Total emissions' and 'Remaining year projections at E...' option.

6. In the Connect to Emissions Impact Dashboard dialog that appears, under **EnrollmentID or BillingAccountID**, enter either your billing account ID (formerly known as the enrollment number) for EA Direct customers or billing account ID for MCA/MPA.



When done, select **Next**.

7. Connect your account:

- For **Authentication method**, select **OAuth2**.
- For **Privacy level setting for this data source**, select **Organizational**.
- When done, select **Sign in and connect**.

X



## Connect to Emissions Impact Dashboard

You are connecting to

Url

<https://gw.us-il301.gateway.prod.island.powerapp.com>



Authentication method

OAuth2

Privacy level setting for this data source [Learn more](#)

Organizational

[Go to the app documentation](#)

Back

Sign in and connect

Cancel

8. Select the user account. Be sure to sign in with the credentials that have access to the enrollmentID/Billing AccountID with valid permissions as explained in the [prerequisites](#).



### Pick an account



Megan Bowen  
mbowen@contoso.com  
Connected to Windows

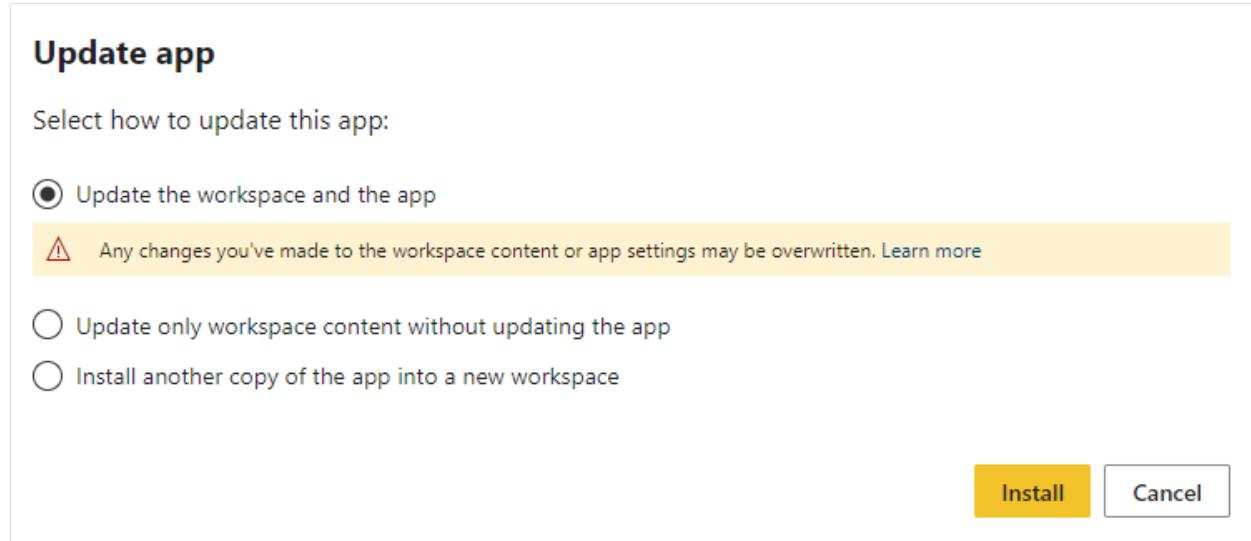


Use another account

9. Wait for the view to build. This can take up to 24 hours. Refresh the dataset after 24 hours.

## Update the app

Periodically you may receive update notifications from Appssource/Power BI about a new version of the app. When you install the new version, the following options are available:



Select **Update the workspace and the app**, and then select **Install**. This will install the update, overwriting the existing/installed workspace and app.

## Issues

If there are any issues with the dataset refresh/app update during the updating process, validate these steps and refresh the dataset.

Follow the steps below to make sure your dataset configurations are set correctly:

1. Go to the workspace panel and open the app workspace.
2. Open the **Scheduled Refresh** option in the dataset settings and make sure the [billing account ID](#) is correct.

All	Content	Datasets + dataflows
	Name	Type
	Microsoft Sustainability Calculator (GA)-SS	Report
	Microsoft Sustainability Calculato...	Dataset

[Schedule refresh](#)

3. Open the **Parameters** section and configure the data source again in the **Data Source** section with the credentials that have access to the **Enrollment ID / Billing Account ID** with valid permissions, mentioned in the [prerequisites](#).

▶ Data source credentials

[Web](#)   [Edit credentials](#)   [Show in lineage view](#)

▶ Sensitivity label

▶ Parameters

EnrollmentIDOrBillingAccountID  
For EA customers please enter your Enrollment ID. For MCA / CSP customers please enter your Billing Account ID

9cda4f18-f144-5194-bc6a-e3169e7768de:0547e0a2-8792-

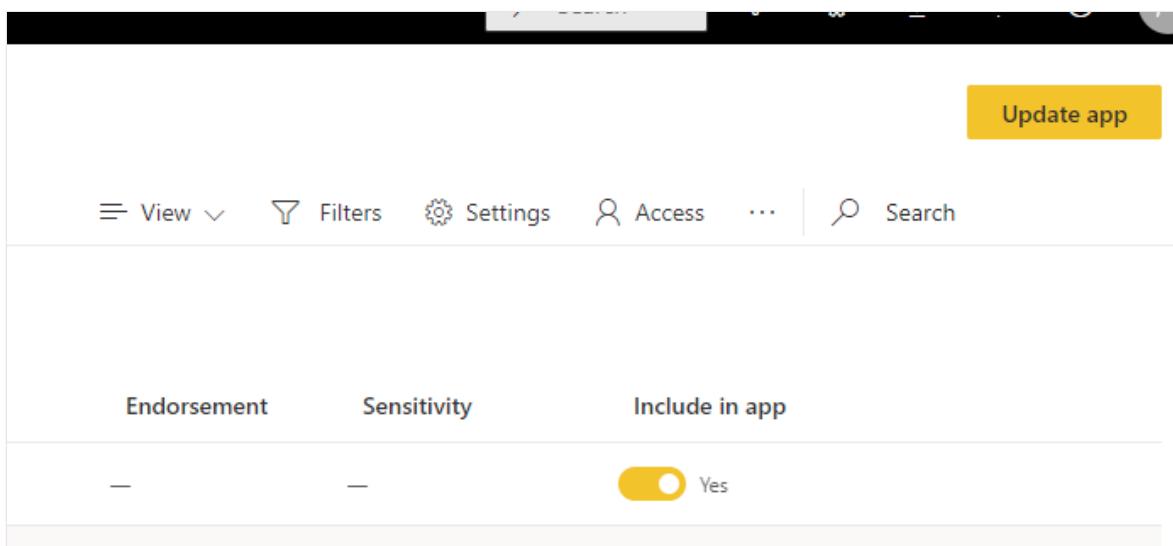
Apply
Discard

4. After the above steps are validated, go back to the app workspace and select the **Refresh** option.

Name	Type
 Microsoft Sustainability Calculator (GA)-SS	Report
 Microsoft Sustainability Calculato...	Dataset

[Refresh now](#)

5. After the dataset has refreshed successfully, select the **Update App** option at the top-right corner of the app workspace.



The screenshot shows the Microsoft Power BI app workspace. At the top right, there is a yellow button labeled "Update app". Below the header, there are several navigation links: "View", "Filters", "Settings", "Access", and "Search". In the main content area, there are three columns: "Endorsement", "Sensitivity", and "Include in app". Under "Include in app", there is a toggle switch set to "Yes".

## Additional resources

- [How-to video ↗](#)
- [The carbon benefits of cloud computing: A study on the Microsoft Cloud in partnership with WSP ↗](#)

## Finding your company's billing account ID

Follow the steps below to find your company's billing account ID, or ask your organization's Azure administrator.

1. In the [Azure portal ↗](#), navigate to **Cost Management + Billing**.
2. In the **Billing Scopes** menu, select your billing account.

3. Under **Settings**, select **Properties**. Your billing account ID will display under **Billing account**.

The screenshot shows the 'Properties' section of the Azure Cost Management interface for a 'Billing account'. On the left, there's a sidebar with navigation links like Overview, Access control (IAM), Cost management, Billing, and Settings. The 'Properties' link is selected. The main area displays account details:

- BILLING ACCOUNT ID:** <billing account id>
- BILLING ACCOUNT TYPE:** Microsoft Customer Agreement (highlighted with a red box)
- COMPANY:** Contoso
- ADDRESS:** Microsoft Corporation  
One Microsoft way  
Redmond WA 98029
- MY ROLE:** Billing account owner

## FAQs

### App setup

I'm receiving an error at the time of connecting my data with the dashboard. What can I do?

First, check Azure Cost Management and verify that you have Admin privileges. If you don't, request this access from your administrator. Next, ensure you're using the correct billing account ID or enrollment number.

I entered my enrollment number/billing account ID, but my company data isn't loading. What's the issue?

The Emissions Impact Dashboard for Azure may take up to 24 hours to load your data. Return after 24 hours and select the Refresh button in Power BI.

Is Microsoft trying to shift responsibility for emissions from Microsoft to me?

No. Carbon emissions from Azure services are reported as Microsoft's scope 1 and 2 emissions, consistent with the industry-standard [Greenhouse Gas \(GHG\) Protocol](#). The GHG Protocol defines scope 3 emissions as emissions another entity emits on your

behalf, and are inherently double-counted. The Emissions Impact Dashboard for Azure provides new transparency to your scope 3 emissions associated with the use of Azure services, specifically Scope 3 Category 1 "Purchased goods and services".

### Why are my emissions from use of the Microsoft cloud so much lower than they would be if I were using an on-premises solution?

Microsoft conducted [a study, published in 2018](#) that evaluated the difference between the Microsoft cloud and on-premises or traditional datacenters. The results show that Azure Compute and Storage are between 52 and 79 percent more energy-efficient than traditional enterprise datacenters, depending on the specific comparison to low, medium, or high efficiency on-premises alternative being made. When taking into account our renewable energy purchases, Azure is between 79 and 98 percent more carbon efficient. These savings are due to four key features of the Microsoft Cloud: IT operational efficiency, IT equipment efficiency, datacenter infrastructure efficiency, and renewable electricity.

### If Microsoft's operations are carbon neutral and powered by renewables, why aren't customer emissions from Azure services zero?

There are two primary reasons why customer emissions from Microsoft aren't zero. The first is related to GHG accounting practices, and the second has to do with the boundary of this analysis. To achieve carbon neutral operations, Microsoft uses carbon offsets to reduce certain emission sources such as onsite fuel combustion for backup generators, refrigerants, and vehicle fleets. These reduce Microsoft's *net* emissions to zero. The dashboard reports *gross* GHG emissions before the application of these offsets, though the volume of offsets applied and net emissions is reported in the **GHG Reporting** tab for further transparency. The second reason is that in addition to the energy and emissions associated with the operation of Microsoft's datacenters, the emissions footprint includes the energy used by Internet Service Providers outside of Microsoft's operational boundary to transmit data between Microsoft datacenters and Azure customers.

### How am I supposed to use this data, and where do I report it?

Your emissions can be reported as part of your company's Scope 3 indirect carbon emissions. Scope 3 emissions are often disclosed in sustainability reports, CDP climate change, and other reporting outlets. In addition to the emissions totals, the emissions savings provide a clear example of how your company's decision to use Microsoft Azure services is contributing to global emissions reductions. To contextualize, the app indicates the equivalent vehicle miles avoided corresponding to the reduction in GHG emissions, based on EPA's equivalency calculator factors as of January 2020.

## What can I do to reduce emissions further?

Being resource and cost efficient in Azure will reduce the environmental impact from your use of Azure. As an example, unused virtual machines are wasteful whether in the cloud or on-premises. Right-sizing virtual machines to improve compute utilization factors (CUF) decreases energy use per useful output, just as it does with physical servers. [Azure Cost Management](#) gives you the tools to plan for, analyze and reduce your spending to maximize your cloud investment. The [sustainability guidance within the Azure Well-Architected Framework \(WAF\)](#) is also designed to help you optimize your cloud workloads and reduce your operational footprint.

## My company contract renewal process is underway and we'll have a new account number. Will I lose my historical emissions data?

Yes, you will. Before your renewal, be sure to download all historical data and reports you'll need for your records.

## Can I export emissions data to Microsoft Excel?

You can export data from the **GHG Preparation** report, **Usage** report, and dashboard page on a per-visualization level. You can't export the overall report's data from the **Export** button on the top header.

### ⓘ Note

Export to Excel is limited to 150,000 rows, and export to CSV is limited to 30,000 rows.

## Methodology

### What is the methodology behind the tool?

The Emissions Impact Dashboard for Azure reflects the specific cloud services consumed and the associated energy requirements, efficiency of the datacenters providing those services, electricity fuel mixes in the regions in which those datacenters operate, and Microsoft's purchases of renewable energy. As part of the app's development, the methodology and its implementation went through third-party verification to ensure that it aligns to the World Resources Institute (WRI)/World Business Council for Sustainable Development (WBCSD) Greenhouse Gas (GHG) Protocol Corporate Accounting and Reporting Standard. The scope of the verification, conducted in accordance with ISO 14064-3: Greenhouse gases--Part 3: Specification with guidance for the validation and verification of greenhouse gas assertions, included the estimation of

emissions from Azure services, but excluded the estimation of on-premises emissions given the counterfactual nature of that estimate. A more detailed description of the carbon calculation is documented in the **Calculation Methodology** tab in the tool.

### What data is required to calculate the Azure carbon footprint? Do you access my company's data?

The estimated carbon calculations are performed based on consumption of Azure services accessed using Azure Consumed Revenue. The dashboard does not access any of your stored customer data. The consumption data is combined with Microsoft's energy and carbon tracking data to compute the estimated emissions associated with your consumption of Azure services based on the datacenters that provide those services.

### Does this calculation include all Azure services and all Azure regions?

The estimates include all Azure services in all Azure regions associated with the tenant ID provided during setup.

## Characterizing on-premises emissions

### Where does the Emissions Impact Dashboard for Azure obtain data about my on-premises emissions and operations?

The Emissions Impact Dashboard for Azure doesn't obtain any information specifically about your on-premises datacenters *except* what you provide. As described in subsequent FAQs, the Emissions Impact Dashboard for Azure relies on industry research and user inputs about the efficiency and energy mix of on-premises alternatives to develop an estimate of on-premises emissions.

### What are the assumptions regarding on-premises estimations? Are efficiency savings just from improvements in Power Usage Effectiveness (PUE)?

Efficiencies associated with Microsoft cloud services include far more than improved PUE. While Microsoft datacenters strive to optimize PUE, the primary efficiency improvements come from IT operational efficiency (dynamic provisioning, multitenancy, server utilization) and IT equipment efficiency (tailoring hardware to services ensuring more energy goes towards useful output), in addition to datacenter infrastructure efficiency (PUE improvements). Our [2018 study](#) quantifies these savings compared to a range of on-premises alternatives ranging from low-efficiency to high-efficiency datacenters. These findings are used to estimate the energy use required for a corresponding on-premises datacenter to provide the same services that each customer consumes on the Microsoft cloud.

## What is the assumed energy mix for the on-premises infrastructure?

By default, the Emissions Impact Dashboard for Azure estimates on-premises emissions based on the mix of renewables and non-renewables on the grid. It is assumed that the on-premises datacenter would be located on the same grid as Microsoft's datacenters. However, for customers who purchase renewable electricity *in addition to* what's on the grid (for example, through Power Purchase Agreements), users can select the percentage of renewable electricity, and the Emissions Impact Dashboard for Azure will adjust on-premises emissions accordingly.

## When should I choose Low, Medium, or High for the efficiency of the on-premises infrastructure?

Users should select the efficiency most representative of the on-premises deployment they would like to compare against, based on the equipment and datacenter characteristics here:

- **Low:** Physical servers and direct attached storage in small localized datacenter (500-1,999 square feet)
- **Medium:** Mix of physical/virtualized servers and attached/dedicated storage in mid-tier internal datacenter (2,000-19,999 square feet)
- **High:** Virtualized servers and dedicated storage in high-end internal datacenter (>20,000 square feet)

# Connect to the Emissions Impact Dashboard for Microsoft 365

Article • 04/06/2023

Calculate emissions from your tenant's usage of Microsoft 365 with the Emissions Impact Dashboard for Microsoft 365.

Accurate carbon accounting requires good information from partners, vendors, and suppliers. The Emissions Impact Dashboard for Microsoft 365 gives you transparency on the carbon emissions generated by your organization's usage of Microsoft 365. Microsoft's carbon accounting extends across all three scopes of emissions with a 3rd party-validated methodology.

## Prerequisites

To install the Emissions Impact Dashboard for Microsoft 365 in Power BI and connect it to your data, make sure you have the following:

- A business, enterprise, or education subscription for Microsoft 365 or Office 365.
- A Power BI Pro license. If you don't have a Power BI Pro license, get a [free trial now](#).
- One of the following Microsoft 365 admin roles:
  - Global admin
  - Exchange admin
  - Skype for Business admin
  - SharePoint admin
  - Global reader
  - Report reader

### Note

In order to ensure that the report successfully refreshes over time, the Microsoft 365 admin credentials of the user who connects the application to your organization's data must persist over time. If that user has the admin role removed after the connection is established, the report will render demo data upon the next refresh.

### Note

The Emissions Impact Dashboard for Microsoft 365 is currently not supported for national/regional cloud deployments including but not limited to Microsoft's US Government clouds and Office 365 operated by 21Vianet.

## Included Microsoft 365 applications

The Emissions Impact Dashboard for Microsoft 365 reports datacenter emissions and active usage associated with the following applications:

- Exchange Online
- SharePoint
- OneDrive
- Microsoft Teams
- Word
- Excel
- PowerPoint
- Outlook

## Install the app

1. Click the following link to get to the app: [Emissions Impact Dashboard for Microsoft 365 ↗](#).
2. On the AppSource page for the app, select **GET IT NOW**. You can also search for the app in Power BI.
3. When prompted, click **Install**.
4. When the app finishes installing, it will appear on your Power BI Apps page. Click on the app and open it.

The screenshot shows the Microsoft Power BI Apps interface. On the left, there's a navigation sidebar with options like Home, Favorites, Recent, Create, Datasets, Goals, and Apps (which is selected and highlighted with a red box). Below that are Shared with me, Deployment pipelines, and Learn. The main area is titled 'Apps' and contains a description: 'Apps are collections of dashboards and reports in one easy-to-find place.' A table lists several apps, including 'Github', 'Analyze Popular Stocks with Power BI', 'Inventory Tracker', 'Premium Capacity Utilization And Metrics', and 'Emissions Impact Dashboard for Microsoft 365'. The 'Emissions Impact Dashboard for Microsoft 365' row is also highlighted with a red box.

## 5. Select Connect your data.

The screenshot shows the 'Overview' tab of the 'Emissions Impact Dashboard for Microsoft 365' app. At the top, there's a warning message: '⚠ You're viewing this app with sample data.' Next to it is a blue button labeled 'Connect your data' (which is highlighted with a red box). Below the header are tabs for Overview, Carbon Intensity, Emissions Savings, GHG Preparation Report, Usage Report, and Calculation Methods. The Overview section displays a chart titled 'Total emissions per month (MTCO2e)' showing monthly emissions from January to December. It includes a legend for Scope 1 (blue), Scope 2 (orange), and Scope 3 (green). The chart shows a general upward trend with some fluctuations. To the right of the chart is a table titled 'Emissions per Microsoft 365 region (MTCO2e)' showing emissions by region: United States (56), Australia (30), Asia Pacific (5), India (3), EMEA (2), and Japan (1).

## 6. In the Connect to Emissions Impact Dashboard for Microsoft 365 dialog, enter your Microsoft 365 tenant ID. For help finding your tenant ID, see [Find your Microsoft 365 tenant ID](#).



## Connect to Emissions Impact Dashboard for Microsoft 365

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

### Parameters

Make sure all required (\*) parameters are filled in before connecting to your data.

#### TenantID \*

Please provide your TenantID here.

For example: 1234-5698-8964-4682

Advanced ▾

When done, click **Next**.

7. Connect your account:

- For **Authentication method**, choose **OAuth2**.
- For **Privacy level setting for this data source**, choose **Organizational**.
- When done, click **Sign in and connect**.



## Connect to Emissions Impact Dashboard for Microsoft 365

You are connecting to

Url

<https://iks-powerbi.ideas.microsoft.com/api/conn>



Authentication method

OAuth2



Privacy level setting for this data source [Learn more](#)

Organizational



[Go to the app documentation ↗](#)

Back

Sign in and connect

Cancel

8. Select the user account. Make sure to sign in with the credentials that have appropriate admin access to your Microsoft 365 tenant (see [list of approved roles above](#)).

Microsoft

Pick an account

---

Megan Bowen  
mbowen@contoso.com  
Connected to Windows

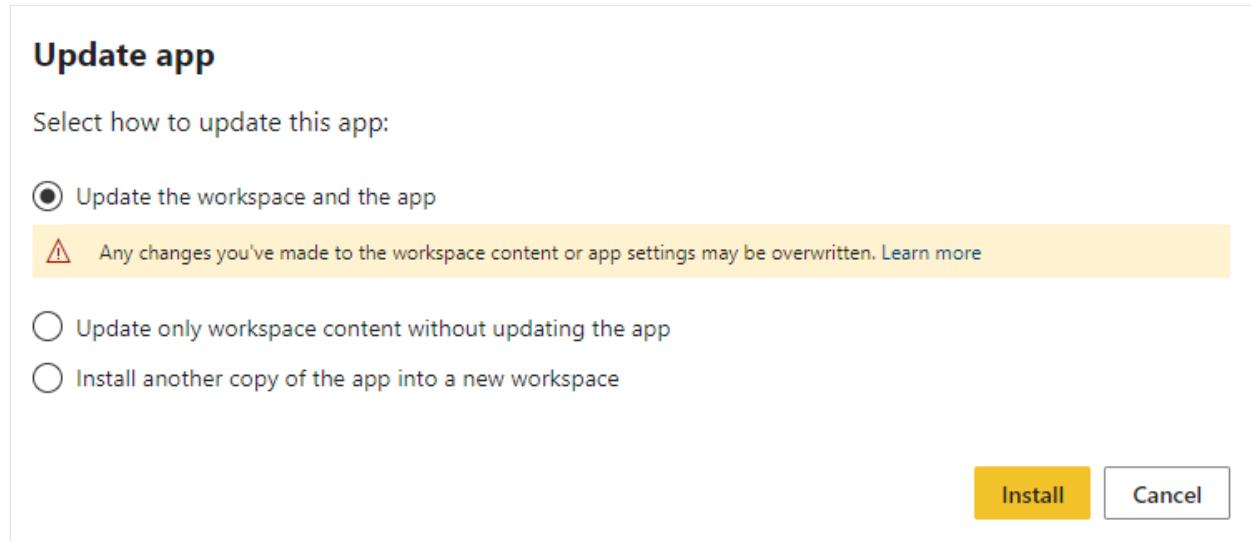
---

Use another account

Wait for the view to build. This can take 24-48 hours. Refresh the dataset after 24 hours.

## Update the app

Periodically you may receive update notifications from Appssource/Power BI about a new version of the app. When you install the new version, the following options are available:



Choose **Update the workspace and the app**, then click **Install**. This will install the update, overwriting the existing/installed workspace and app. After installation is complete, repeat steps 5-8 in the [Install the app](#) section above to re-establish the connection for the new app version.

## Issues

If there are any issues with the dataset refresh/app update during the updating process, validate the following steps and refresh the dataset.

Follow the steps below to make sure your dataset configurations are set correctly:

1. Go to the workspace panel and open the app workspace.
2. Open the **Scheduled Refresh** option in the dataset settings.

All	Content	Datasets + dataflows
Name	Type	
 Emissions Impact Dashboard for Microsoft 365 (Pr...	Report	
 Emissions Impact Dashboard for ...	Dataset	<span>↻</span> <span>🔗</span> <span>⋮</span> <span>Schedule refresh</span>

3. Open the **Parameters** section and configure the data source once again in the **Data Source** section with the credentials with which you have access to the **Tenant ID** with valid permissions, mentioned in the [prerequisites](#) section above.

▲ Data source credentials

OData [Edit credentials](#) [Show in lineage view](#) □

► Sensitivity label

▲ Parameters

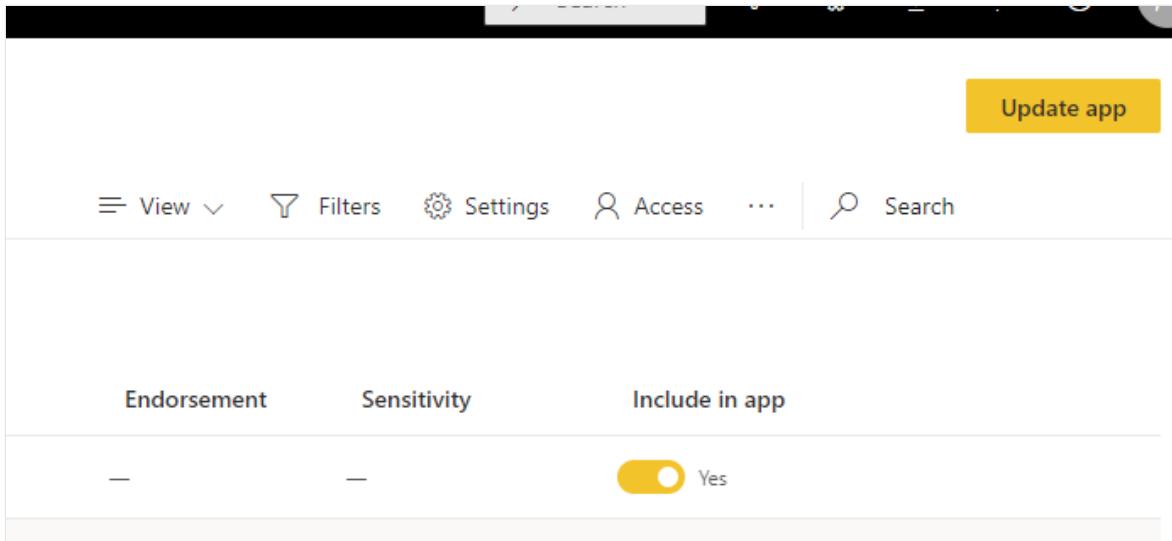
TenantID  
Please provide your TenantID here.  
123-456-789-000

Apply Discard

4. Once the above steps are validated, go back to the app workspace and click on the **Refresh** option.

All	Content	Datasets + dataflows
Name	Type	
 Emissions Impact Dashboard for Microsoft 365 (Pr...	Report	
 Emissions Impact Dashboard for ...	Dataset	<span>↻</span> <span>🔗</span> <span>⋮</span> <span>Refresh now</span>

- Once the dataset has refreshed successfully, click the **Update App** option at the top-right corner of the app workspace.



## Opting out

To opt out of the Emissions Impact Dashboard for Microsoft 365, go to the **Opt out** tab in the app and follow the prompts. This step must be performed by a user with one of the roles listed in the [prerequisites](#) section above.

Within 48 hours of submitting your opt out request, go back to Power BI and delete the Emissions Impact Dashboard for Microsoft 365 app. If you don't perform this step, and the app does a data refresh more than 48 hours after your opt out request, your tenant will be opted back in to emissions processing.

If you opt out of the Emissions Impact Dashboard for Microsoft 365, you can always opt back in by re-installing and re-connecting to the app, but historical data will not initially be available.

## Data schemas

*Data schema: GHG Preparation Report*

If you choose to export data from the **GHG Preparation Report** tab, use the following data schema to determine the definition of each output column.

Column	Type	Description	Example
Year	Integer	The year in which the emissions are being reported.	2023
Quarter	String	The quarter in which the emissions are being reported.	Qtr1

Column	Type	Description	Example
Month	String	The month in which the emissions are being reported.	January
Region	String	The Microsoft 365 region with which the emissions are associated.	Asia Pacific
Scope	String	Indicates whether the emissions value is associated with Microsoft's Scope 1, Scope 2, or Scope 3.	Scope1
Carbon Emissions (mtCO2e)	Double	Volume of carbon dioxide equivalent, measured in metric tons.	1.019

### *Data schema: Usage Report*

If you choose to export data from the **GHG Preparation Report** tab, use the following data schema to determine the definition of each output column.

Column	Type	Description	Example
Year	Integer	The year in which the active usage is being reported.	2023
Quarter	String	The quarter in which the active usage is being reported.	Qtr1
Month	String	The month in which the active usage is being reported.	January
Microsoft 365 active users	Integer	Count of unique active users across the <a href="#">Microsoft 365 apps currently included in the report</a> . If a given user has usage of multiple applications, they are only counted once.	5000

## Accessing the data via the Cloud for Sustainability API (preview)

The same underlying data that powers the Emissions Impact Dashboard for Microsoft 365 can be accessed programmatically through the Cloud for Sustainability API (preview). [Learn more here](#).

## Additional resources

- [The role of embodied carbon in cloud emissions: Assessing the scale and sources of Microsoft 365 emissions, and what organizations can do to help reduce them \(PDF Download\)](#).

- The carbon benefits of cloud computing: A study on the Microsoft Cloud in partnership with WSP ↗
- Microsoft Sustainability webpage ↗

## Finding your company's Microsoft 365 Tenant ID

Follow the steps on [this page](#) to find your tenant ID.

## FAQs

### App setup

I'm receiving an error at the time of connecting my data with the dashboard. What can I do?

First, check in the Microsoft 365 Admin Center that you have one of the roles listed in the [prerequisites](#) section above. If you don't, request this access from your administrator. Next, ensure you're using the correct [Microsoft 365 tenant ID](#).

I entered my tenant ID, but my company data isn't loading. What's the issue?

The Emissions Impact Dashboard for Microsoft 365 may take 24-48 hours to load your data after completing the connection process. Return after 24 hours and select the **Refresh** button in Power BI in the app workspace, as shown in the [Issues](#) section above.

How do I know that I have the latest version of the template app installed?

Microsoft may periodically release a new version of the Emissions Impact Dashboard for Microsoft 365 in order to deliver new features or update text and visuals in the report. Microsoft will notify your organization of new version releases via the [Microsoft 365 Message center](#). In addition, the user in your organization who installed the application will receive a [notification](#) in Power BI requesting that they update to the latest version of the report. That user should follow the steps [above](#) to update to the latest version of the app.

### Emissions data

I successfully connected to my tenant's data. Why do I still see demo data in the report, even after waiting 48 hours?

This could indicate that you don't have one of the Microsoft 365 admin roles listed in the [prerequisites](#) section above. Make sure that one of these roles is assigned to your

profile, then open the app workspace and refresh the dataset (see further instructions in the [Issues](#) section above).

### I successfully connected to my tenant's data. Why do I see blank data in the report?

This likely indicates that your tenant's emissions volumes are very small. The charts in this report do not display emissions values that are lower than 0.001 MTCO2e.

### Why can't I see emissions data for the previous month?

Emissions data for a given month will be available by the 14th day after the end of that month (including non-business days). Ensure that your dataset is scheduled to refresh automatically on a daily or weekly basis so that you always have access to the latest information.

### Why can't I see usage information prior to June 2022 in the Carbon Intensity tab?

Currently, information on Microsoft 365 usage is only available in the report from June 2022 onwards. For more usage history, see the [Usage report](#) in the Microsoft 365 Admin Center.

### Why do I see emissions from regions outside of my Microsoft 365 data location?

Microsoft 365 emissions are allocated based on data storage as well as active usage and compute associated with Microsoft 365 applications. Emissions allocated based on compute may occur in regions outside your Microsoft 365 data location. Learn more about Microsoft 365 data residency [here](#).

### Where is the data used to produce this report stored?

It is stored in the United States.

### How can I influence my tenant's Microsoft 365 emissions numbers?

Refer to [this white paper \(PDF Download\)](#) for guidance on how to interpret and act on the information reported in the Emissions Impact Dashboard for Microsoft 365.

### Why can I only see emissions data for the past 12 months?

Currently, the Emissions Impact Dashboard for Microsoft 365 only reports on emissions from the prior 12 months, and it is not possible for app users to access emissions data prior to this window. If in the future this timeframe will be extended, it will be made available via a new app version release and announced via the [Microsoft 365 Message center](#).

**In the 'carbon intensity' tab, why do I see my emissions grow even though the count of Microsoft 365 users in our tenant stayed flat or shrunk (or vice versa) from one month to the next?**

It is possible for your Microsoft 365 emissions figure to move in the opposite direction of your count of unique active users in Microsoft 365. This most often occurs when an organization that already uses one or multiple Microsoft 365 applications begins the process of onboarding to a new application. In this scenario, the organization would start getting allocated emissions associated with the new application, but the overall unique active user count displayed in the 'carbon intensity' tab of the dashboard might remain constant or even decline. This is because the 'user count' figure only counts each user once, even if they use multiple applications. So, for example, if an organization has 500 users of Exchange in June (and doesn't use any other Microsoft 365 applications that month) and subsequently onboards all 500 of those users to SharePoint in July, their overall emissions figure would go up in July but their unique active user count would remain the same. The Emissions Impact Dashboard only displays a single unique active user count in the 'carbon intensity' tab; to parse out usage of specific Microsoft 365 applications, organizations can visit the [Apps usage report in the Microsoft 365 Admin Center](#).

## Methodology

### What is the methodology behind the tool?

The Emissions Impact Dashboard for Microsoft 365 reflects the specific cloud services consumed and the associated energy requirements, efficiency of the datacenters providing those services, electricity fuel mixes in the regions in which those datacenters operate, and Microsoft's purchases of renewable energy. As part of the app's development, the methodology and its implementation went through third-party verification to ensure that it aligns to the World Resources Institute (WRI)/World Business Council for Sustainable Development (WBCSD) Greenhouse Gas (GHG) Protocol Corporate Accounting and Reporting Standard. The scope of the verification, conducted in accordance with ISO 14064-3: Greenhouse gases--Part 3: Specification with guidance for the validation and verification of greenhouse gas assertions, included the estimation of emissions from Microsoft 365 services, but excluded the estimation of on-premises emissions given the counterfactual nature of that estimate. A more detailed description of the carbon calculation is documented in the **Calculation Methodology** tab in the tool.

## Characterizing on-premises emissions

## **Where does the Emissions Impact Dashboard for Microsoft 365 obtain data about my on-premises emissions and operations?**

The Emissions Impact Dashboard for Microsoft 365 doesn't obtain any information specifically about your on-premises datacenters except what you provide. As described in subsequent FAQs, the Emissions Impact Dashboard for Microsoft 365 relies on industry research and user inputs about the efficiency and energy mix of on-premises alternatives to develop an estimate of on-premises emissions.

## **What are the assumptions regarding on-premises estimations? Are efficiency savings just from improvements in Power Usage Effectiveness (PUE)?**

Efficiencies associated with Microsoft cloud services include far more than improved PUE. While Microsoft datacenters strive to optimize PUE, the primary efficiency improvements come from IT operational efficiency (dynamic provisioning, multitenancy, server utilization) and IT equipment efficiency (tailoring hardware to services ensuring more energy goes towards useful output), in addition to datacenter infrastructure efficiency (PUE improvements). Our [2018 study](#) quantifies these savings compared to a range of on-premises alternatives ranging from low-efficiency to high-efficiency datacenters. These findings are used to estimate the energy use required for a corresponding on-premises datacenter to provide the same services that each customer consumes on the Microsoft cloud.

## **What is the assumed energy mix for the on-premises infrastructure?**

By default, the Emissions Impact Dashboard for Microsoft 365 estimates on-premises emissions based on the mix of renewables and non-renewables on the grid. It is assumed that the on-premises datacenter would be located on the same grid as Microsoft's datacenters. However, for customers who purchase renewable electricity *in addition* to what's on the grid (for example, through Power Purchase Agreements), users can select the percentage of renewable electricity, and the Emissions Impact Dashboard for Microsoft 365 will adjust on-premises emissions accordingly.

## **When should I choose Low, Medium, or High for the efficiency of the on-premises infrastructure?**

Users should select the efficiency most representative of the on-premises deployment they would like to compare against, based on the equipment and datacenter characteristics here:

- **Low:** Physical servers and direct attached storage in small localized datacenter (500-1,999 square feet).

- **Medium:** Mix of physical/virtualized servers and attached/dedicated storage in mid-tier internal datacenter (2,000-19,999 square feet).
- **High:** Virtualized servers and dedicated storage in high-end internal datacenter (>20,000 square feet).

# Connect to Office365Mon with Power BI

Article • 08/01/2023

Analyzing your Office 365 outages and health performance data is easy with Power BI and the Office365Mon template app. Power BI retrieves your data, including outages and health probes, and then builds an out-of-the-box dashboard and reports based on that data.

Connect to the [Office365Mon template app](#) for Power BI.

## ⓘ Note

You need an Office365Mon admin account to connect to and load the Power BI template app.

## How to connect

1. Select **Connect your data** at the top of the screen:



2. In the Connect to Office365 Power BI Template Pack window, select **Next**:



## Connect to Office365 PowerBI Template Pack

X

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

### Parameters

Make sure all required (\*) parameters are filled in before connecting to your data.



This app doesn't have any parameters.

Advanced ▾

Next

Cancel

3. In the **Authentication method** box, select **OAuth2**. You can change the privacy level if you want. For more information, select **Learn more** in the window. When you're done, select **Sign in and connect**.



## Connect to Office365 PowerBI Template Pack

X

You are connecting to

ExtensionDataSourceKind

Office365Mon2



ExtensionDataSourcePath

Office365Mon2

Authentication method

OAuth2

Privacy level setting for this data source [Learn more](#)

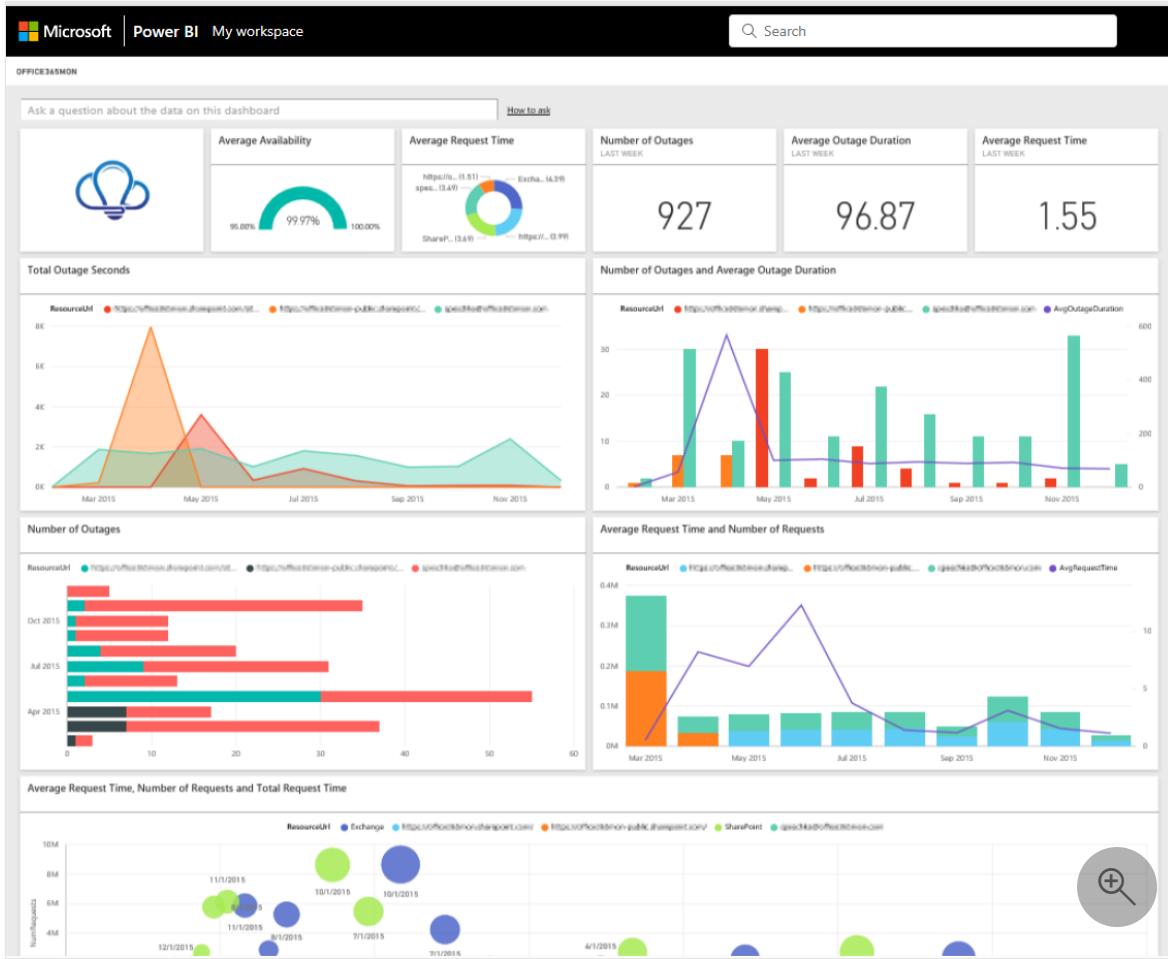
None

Back

Sign in and connect

Cancel

4. When prompted, enter your Office365Mon admin credentials and complete the authentication process.
5. After Power BI imports the data, you see a new dashboard, report, and dataset in your workspace. Select **Office365Mon**.



## What now?

- Try [asking a question in the Q&A box](#) at the top of the dashboard.
- [Change the tiles](#) in the dashboard.
- [Select a tile](#) to open the underlying report.
- Change the refresh schedule. The dataset is scheduled to refresh daily. You can change the schedule, or refresh it on demand by selecting **Refresh now** in the workspace.

## Troubleshooting

If you see a **Need admin approval** error when you try to sign in to Office365Mon, the account you're using doesn't have permissions to retrieve the data. You need to use an Office365Mon admin account when you sign in.

## Next steps

- [What is Power BI?](#)
- [Get data for Power BI](#)

# Connect to Project Web App with Power BI

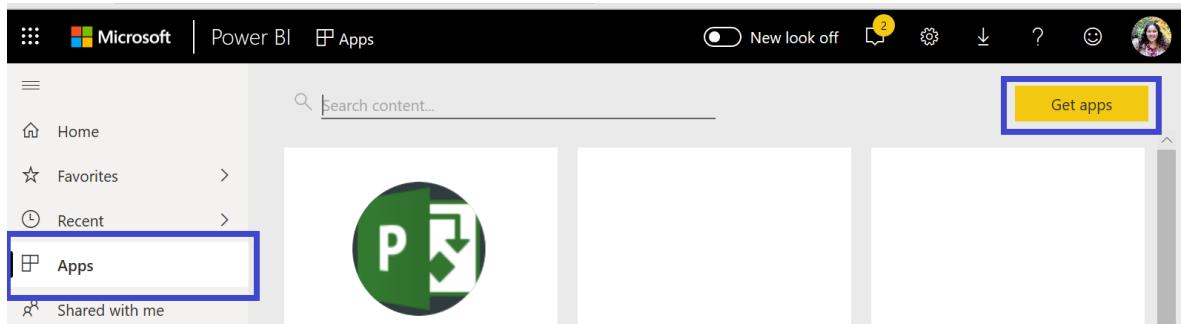
Article • 08/18/2022

Microsoft Project Web App is a flexible online solution for project portfolio management (PPM) and everyday work. Project Web App enables organizations to get started, prioritize project portfolio investments and deliver the intended business value. The Project Web App Template App for Power BI allows you to unlock insight from Project Web App to help manage projects, portfolios and resources.

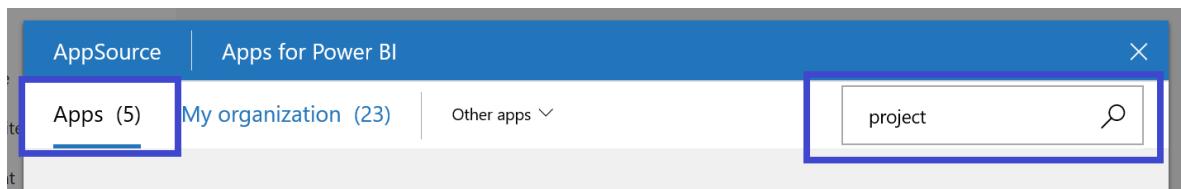
Connect to the [Project Web App Template App](#) for Power BI.

## How to connect

1. Select Apps in the nav pane > select Get apps in the upper right corner.

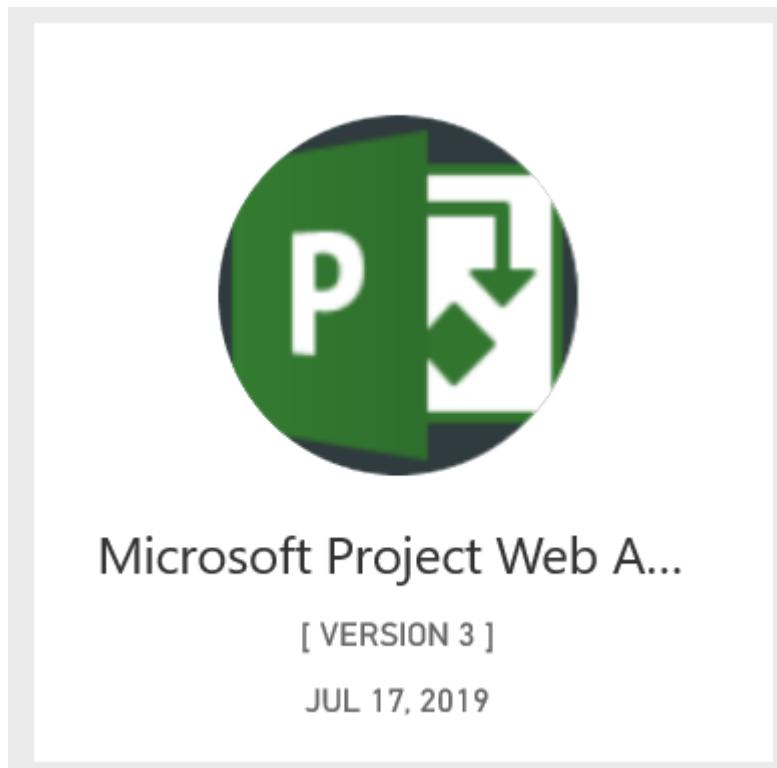


2. In the Services box, select Get.



3. In AppSource, select the Apps tab, and search/select Microsoft Project Web App.

4. You will get a message saying - Install this Power BI App? select Install.



5. In the Apps pane, select the Microsoft Project Web App tile.

## Get started with your new app

Explore your app with sample data, go to the workspace to customize as needed and share with your organization, or connect your data to get up and running.

**Connect your data**  
Connect to a data source to view your new app with your own data.

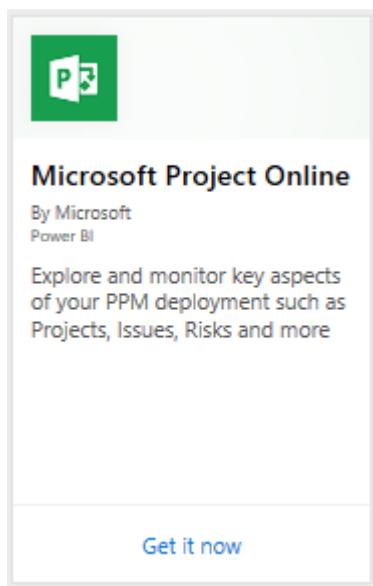
**Explore with sample data**  
Open your new app to start exploring with sample data.

**Customize and share**  
Your app comes with a workspace, so you can customize and share it, just like an app you built yourself.

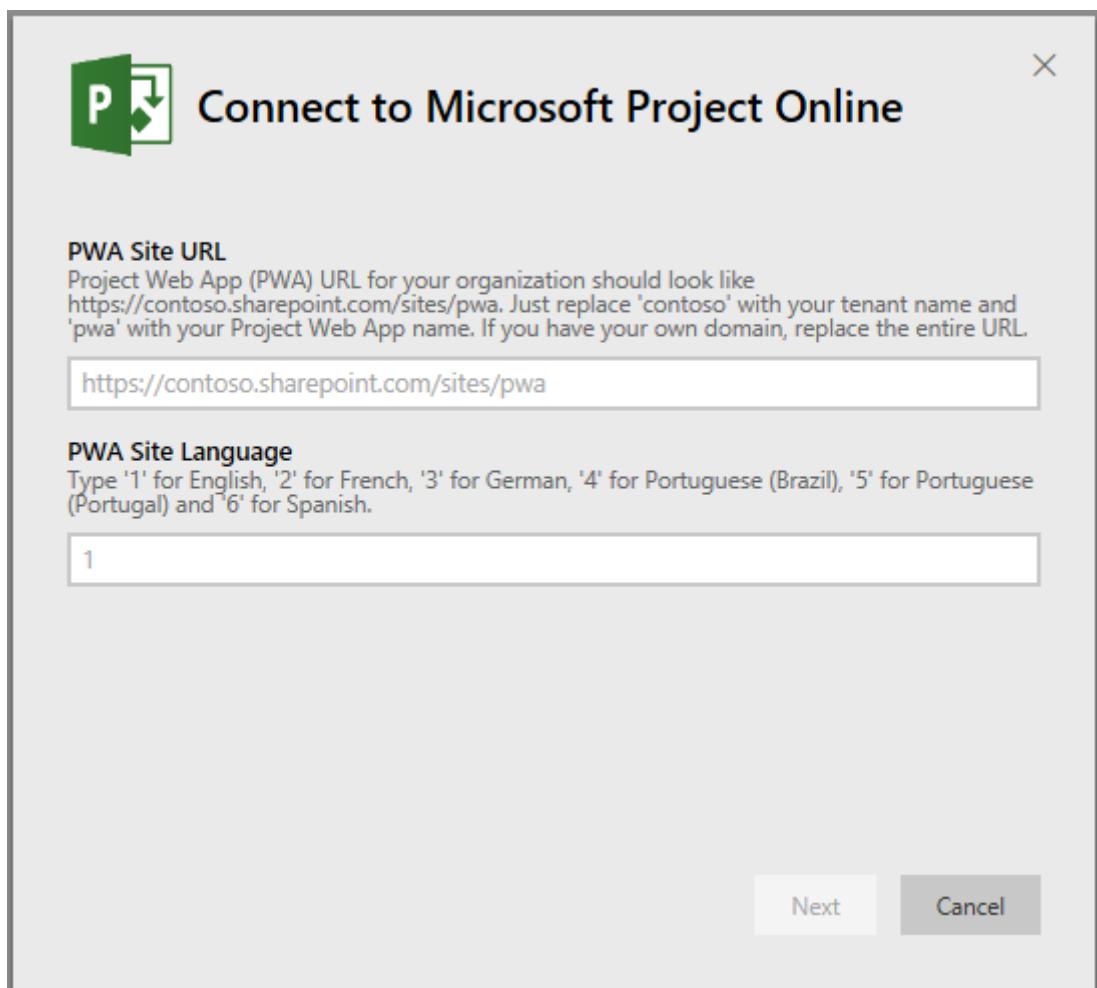
**Connect**      **Explore app**      **Edit workspace**

[Don't show this again](#)

6. In **Get started with your new app**, select **Connect data**.



7. In the **Project Web App URL** text box, enter the URL for the Project Web App (PWA) you want to connect to. Note this may differ from the example if you have a custom domain. In the **PWA Site Language** text box, type the number that corresponds to your PWA site language. Type the single digit '1' for English, '2' for French, '3' for German, '4' for Portuguese (Brazil), '5' for Portuguese (Portugal) and '6' for Spanish.



8. For Authentication Method, select **oAuth2 > Sign In**. When prompted, enter your Project Web App credentials and follow the authentication process.

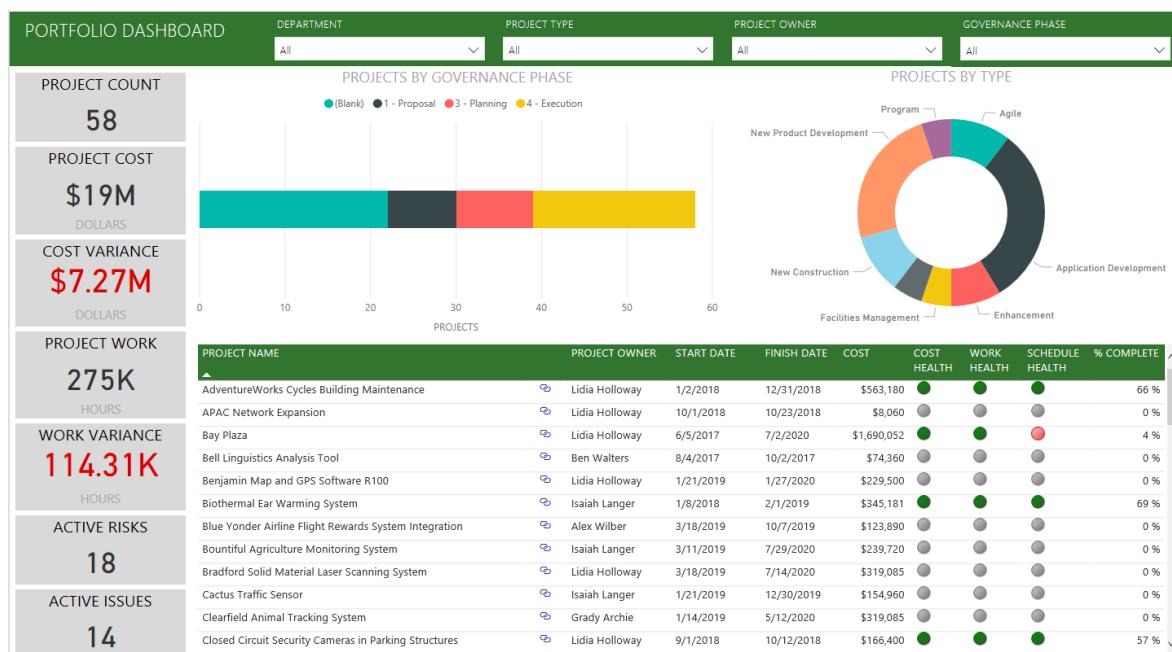
**ⓘ Note**

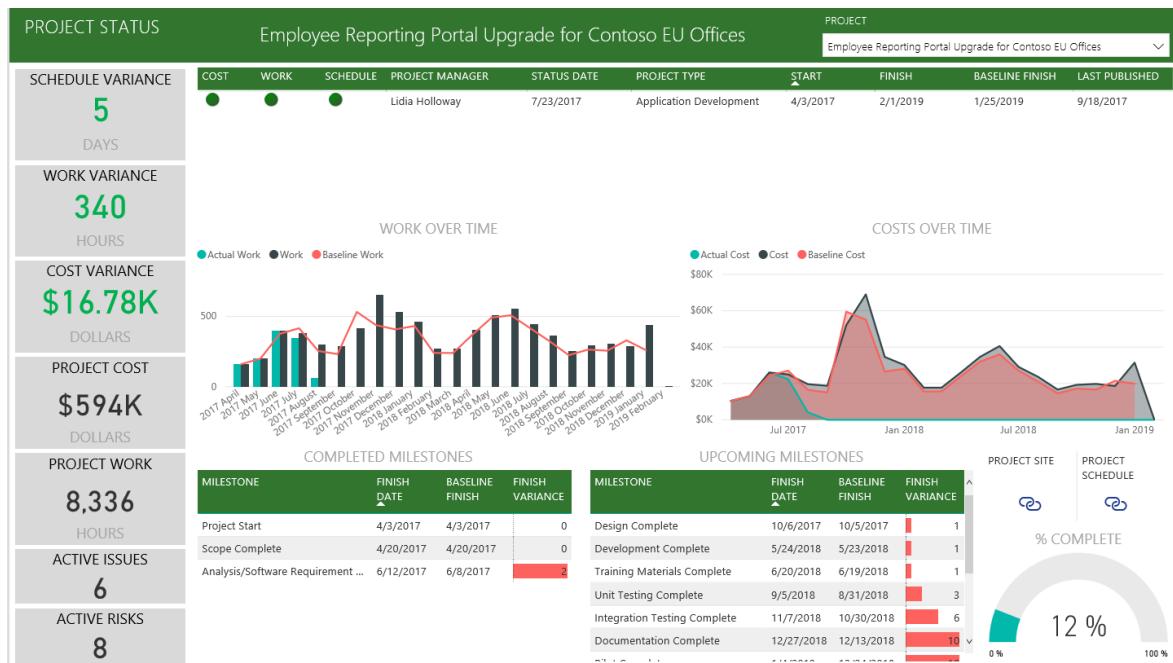
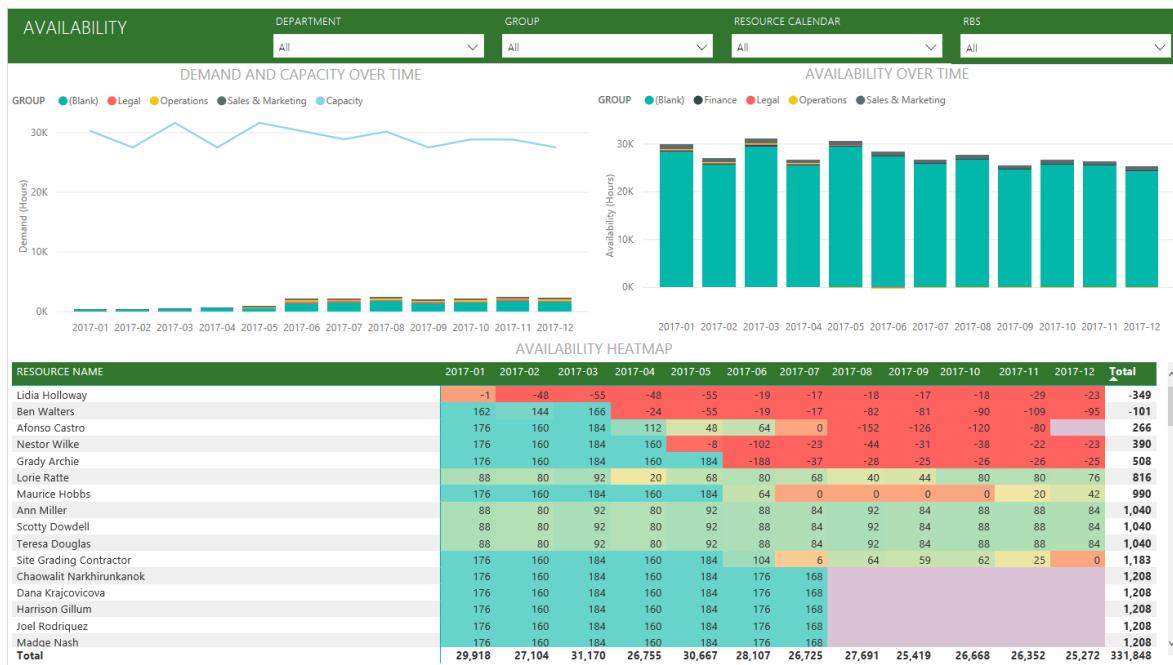
You need to have Portfolio Viewer, Portfolio Manager, or Administrator permissions for the Project Web App you are connecting to.

9. You'll see a notification indicating your data is loading. Depending on the size of your account this may take some time. After Power BI imports the data, you will see the contents of your new workspace. You may need to refresh the dataset to get the latest updates.

After Power BI imports the data you will see the report with 13 pages and dataset in the nav pane.

10. Once your reports are ready, go ahead and start exploring your Project Web App data! The Template App comes with 13 rich and detailed reports for the Portfolio Overview (6 report pages), Resource Overview (5 report pages) and Project Status (2 report pages).





## What now?

- While your dataset will be scheduled to refresh daily, you can change the refresh schedule or try refreshing it on demand using Refresh Now.

## Expand the Template App

Download the [GitHub PBIT file](#) to further customize and update the template app.

## Next steps

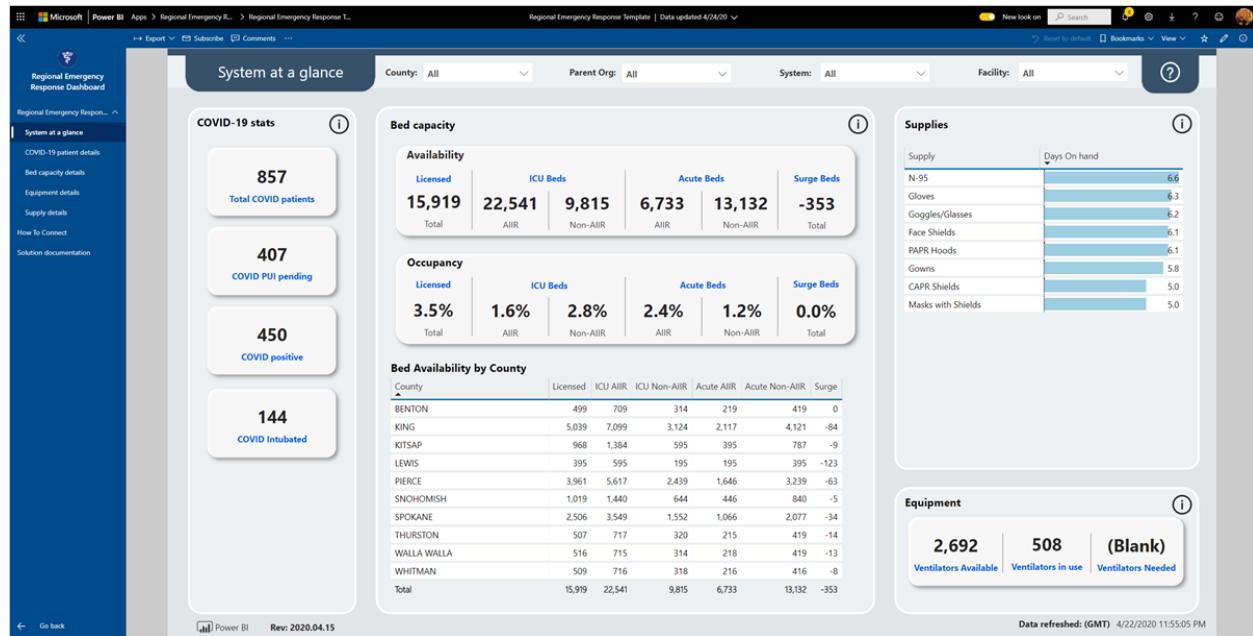
[Get started in Power BI](#)

[Get data in Power BI](#)

# Connect to the Regional Emergency Response Dashboard

Article • 11/30/2020

The Regional Emergency Response Dashboard is the reporting component of the [Microsoft Power Platform Regional Emergency Response solution](#). Regional organization admins can view the dashboard in their Power BI tenant, enabling them to quickly view important data and metrics that will help them make efficient decisions.



This article tells you how to install the Regional Emergency Response app using the Regional Emergency Response Dashboard template app, and how to connect to the data sources.

For detailed information about what is presented in the dashboard, see [Get insights](#).

After you've installed the template app and connected to the data sources, you can customize the report as per your needs. You can then distribute it as an app to colleagues in your organization.

## Prerequisites

Before installing this template app, you must first install and set up the [Regional Emergency Response solution](#). Installing this solution creates the datasource references necessary to populate the app with data.

When installing Regional Emergency Response solution, take note of the [URL of your Common Data Service environment instance](#). You will need it to connect the template

app to the data.

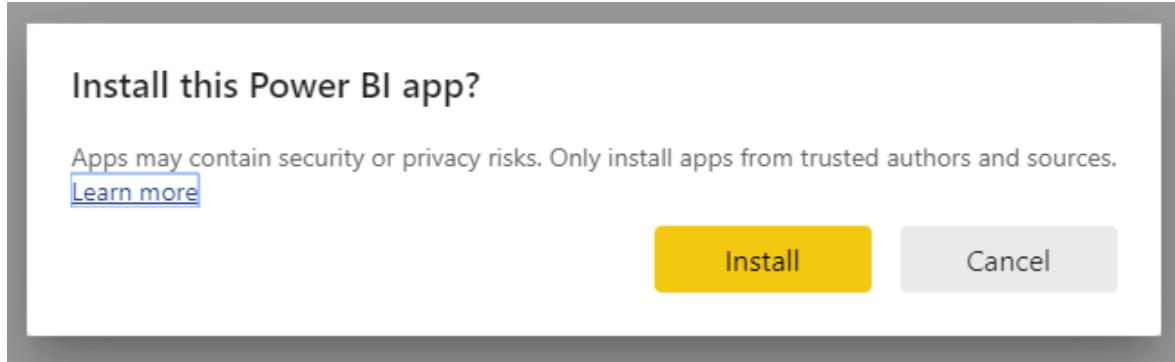
## Install the app

1. Click the following link to get to the app: [Regional Emergency Response Dashboard template app ↗](#)

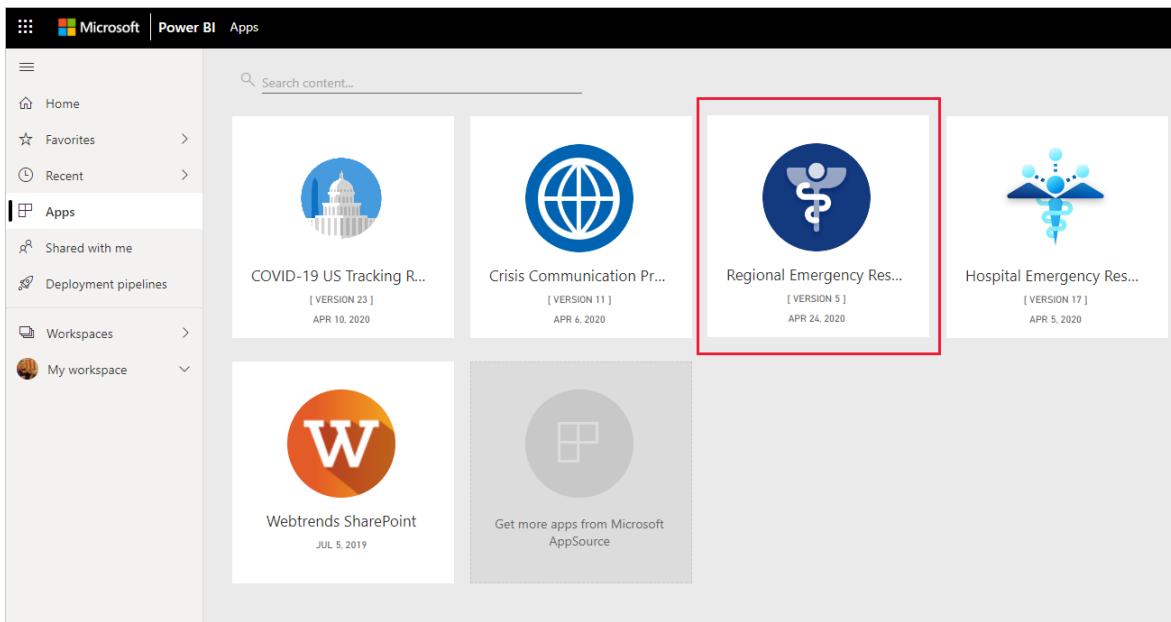
2. On the AppSource page for the app, select [GET IT NOW ↗](#).

The screenshot shows the Microsoft AppSource interface. In the top navigation bar, 'Microsoft' and 'AppSource' are visible, along with a search bar and user profile icons. Below the navigation, the breadcrumb path 'Apps > Regional emergency response' is shown. The main content area features a large thumbnail image of the app icon (a stylized caduceus-like symbol) and the app's name 'Regional emergency response'. Below the name, it says 'Microsoft Corporation' and has a 'Write a review' button. There are two tabs: 'Overview' (which is selected) and 'Reviews'. A prominent blue button labeled 'GET IT NOW' is highlighted with a red box. To the right of the button, there is a brief description: 'Get insight at a glance to manage and maximize hospital & healthcare resources.' Below this, there is a note about the app being a sample and its intended use for reference information only. To the right of the description, there are three smaller screenshots showing different views of the app's interface, including a dashboard with various metrics and tables.

3. Select **Install**.



Once the app has installed, you see it on your Apps page.



## Connect to data sources

1. Select the icon on your Apps page to open the app.
2. On the splash screen, select **Explore**.

**Get started with your new app**

Explore your app with sample data, go to the workspace to customize as needed and share with your organization, or connect your data to get up and running.

**Connect your data**  
Connect to a data source to view your new app with your own data.

**Explore with sample data**  
Open your new app to start exploring with sample data.

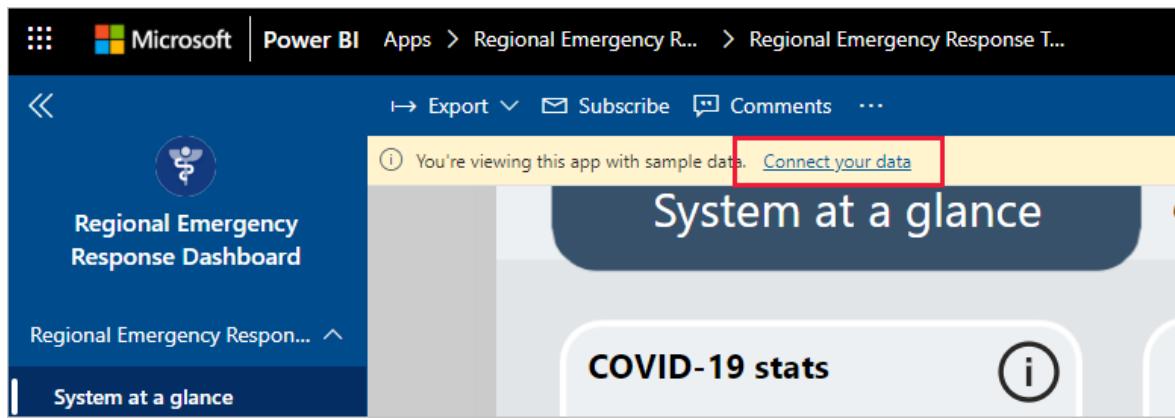
**Customize and share**  
Your app comes with a workspace, so you can customize and share it, just like an app you built yourself.

**Explore app**

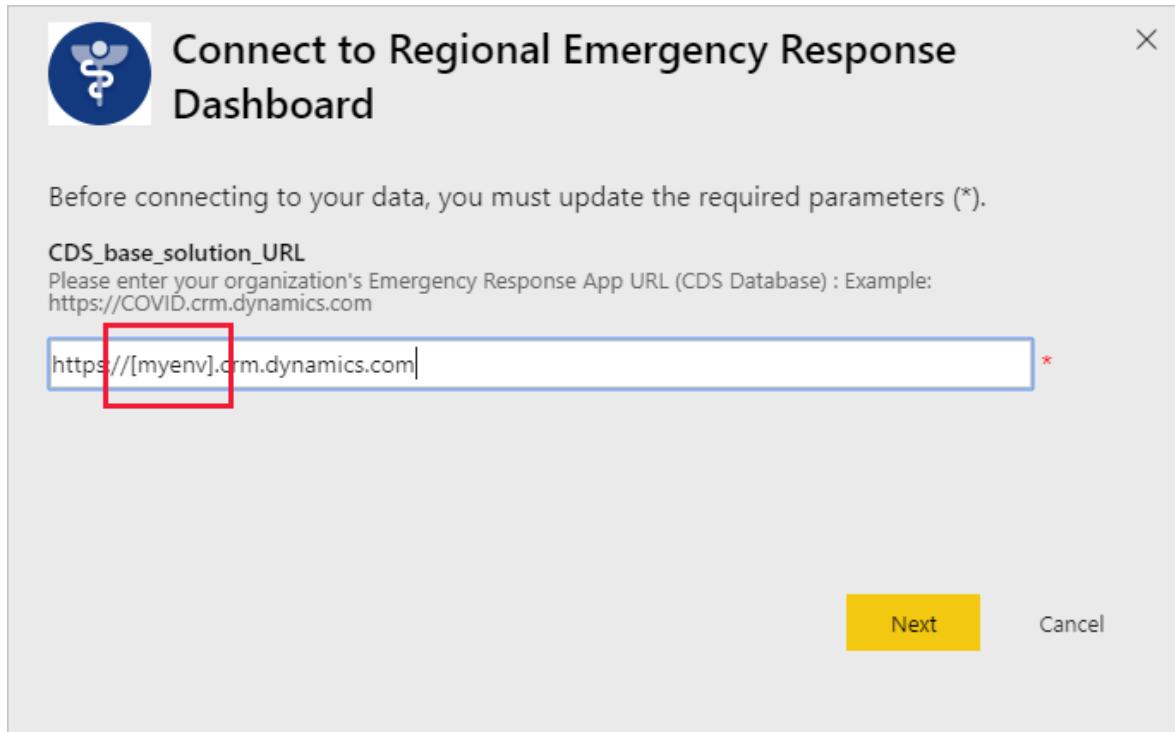
Don't show this again

The app opens, showing sample data.

3. Select the **Connect your data** link on the banner at the top of the page.



4. In the dialog box that appears, type the [URL of your Common Data Service environment instance](#). For example: [https://\[myenv\].crm.dynamics.com](https://[myenv].crm.dynamics.com). When done, click **Next**.



5. In the next dialog that appears, set the authentication method to **OAuth2**. You don't have to do anything to the privacy level setting.

Select **Sign in**.



## Connect to Regional Emergency Response Dashboard

X

url

Authentication method

Privacy level setting for this data source

Cancel

6. At the Microsoft sign-in screen, sign in to Power BI.



### Pick an account

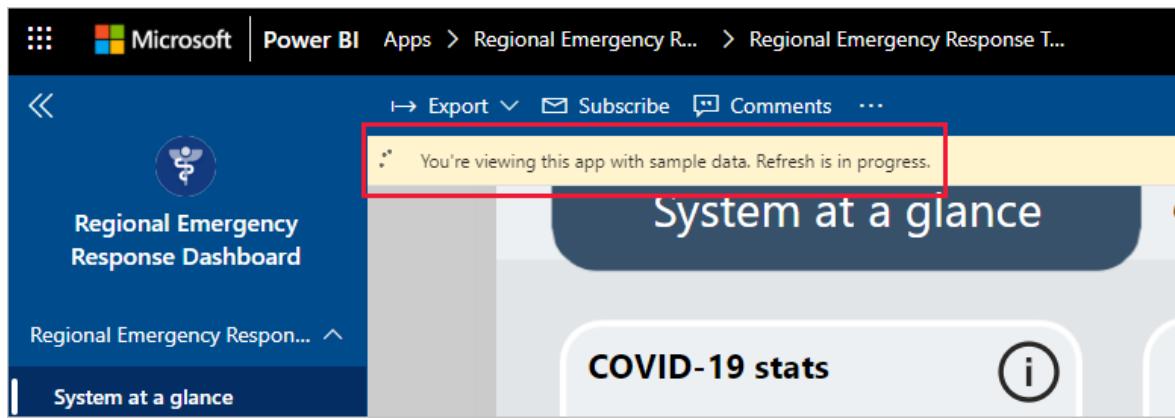


Megan Bowen  
meganb@contoso.com  
Connected to Windows



Use another account

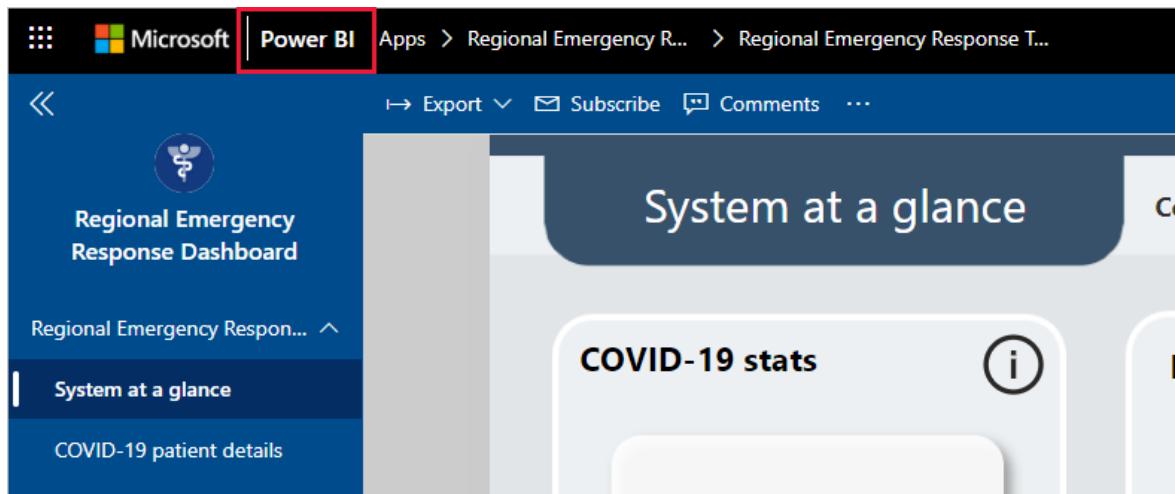
After you've signed in, the report connects to the data sources and is populated with up-to-date data. During this time, the activity monitor turns.



## Schedule report refresh

When the data refresh has completed, [set up a refresh schedule](#) to keep the report data up to date.

1. In the top header bar, select **Power BI**.



2. In the left navigation pane, look for the Regional Emergency Response Dashboard workspace under **Workspaces**, and follow the instructions described in the [Configure scheduled refresh](#) article.

## Customize and share

See [Customize and share the app](#) for details. Be sure to review the [report disclaimers](#) before publishing or distributing the app.

## Next steps

- Understanding the Regional Emergency Response dashboard
- Set up and learn about the Crisis Communication sample template in Power Apps

- Questions? [Try asking the Power BI Community](#)↗
- What are Power BI template apps?
- Install and distribute template apps in your organization

# Salesforce Analytics for Sales Managers

Article • 03/07/2022

The Salesforce Analytics for Sales Managers template app includes visuals and insights for analyzing your marketing effort.

The app's out-of-the-box dashboard provides key metrics, such as your sales pipeline, best accounts, and KPI's. You can drill down into the report for more details on each aspect. Fully interactive visuals help you explore your data further.

In this article we will walk through the app using sample data to give you an idea of how you can use the app to gain key insights into your sales data.

## Prerequisites

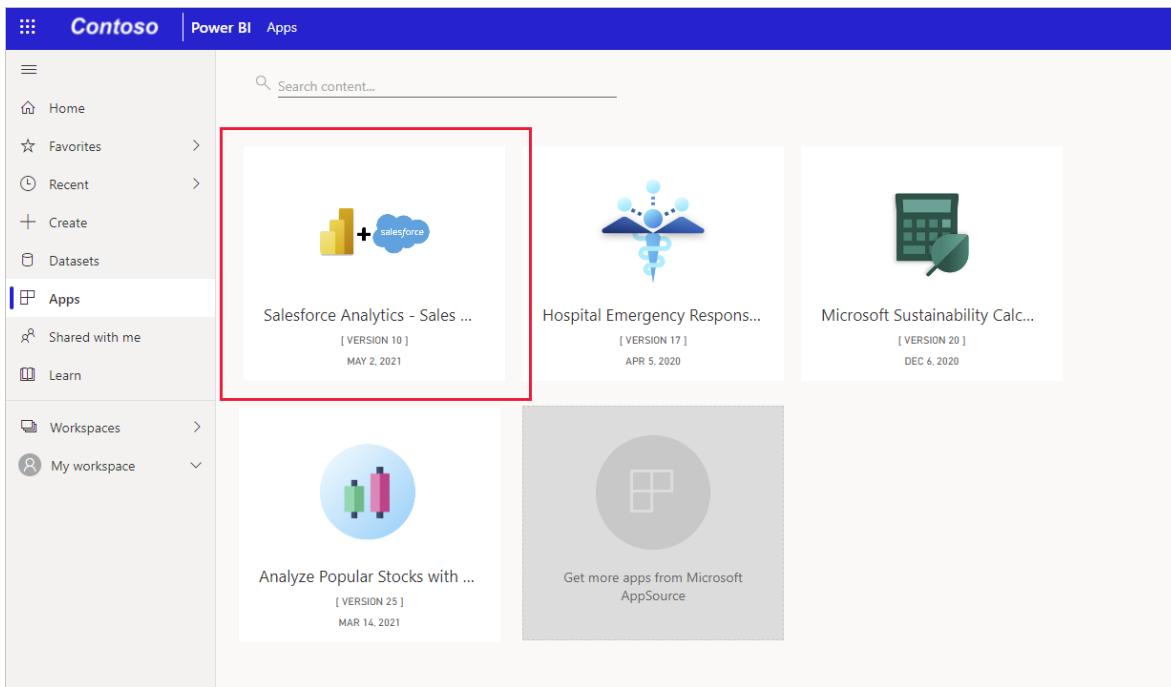
- Power BI Pro
- A marketing, developer, or admin Salesforce subscription

## Install the app

1. Click the following link to get to the app: [Salesforce Analytics for Sales Managers template app.](#)
2. Once you're on the App's AppSource page, click [GET IT NOW](#).



3. When prompted , click **Install**. Once the app has installed, you will see it on your Apps page.



## Connect to data sources

1. Click the icon on your Apps page to open the app. The app opens, showing sample data.
2. Select the **Connect your data** link on the banner at the top of the page.

The screenshot shows the Power BI app interface for the 'Salesforce Analytics - Sales Manager' app. At the top, there's a banner with the message 'You're viewing this app with sample data.' followed by a blue link 'Connect your data'. This banner is highlighted with a red box. Below the banner, the main content area displays a KPI card with the number '13' and the text 'Overall view of the sales activities that is going on in...'. On the left, there's a sidebar with a tree view showing 'SalesManager' expanded, with 'KPI' and 'Sales Manager' listed under it. The top navigation bar includes icons for File, Export, Chat in Teams, Comment, and Subscription.

3. The parameters dialog will appear. There are no required parameters. Click **Next**.



## Connect to Salesforce Analytics - Sales Manager

X

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

### Parameters

Make sure all required (\*) parameters are filled in before connecting to your data.



This app doesn't have any parameters.

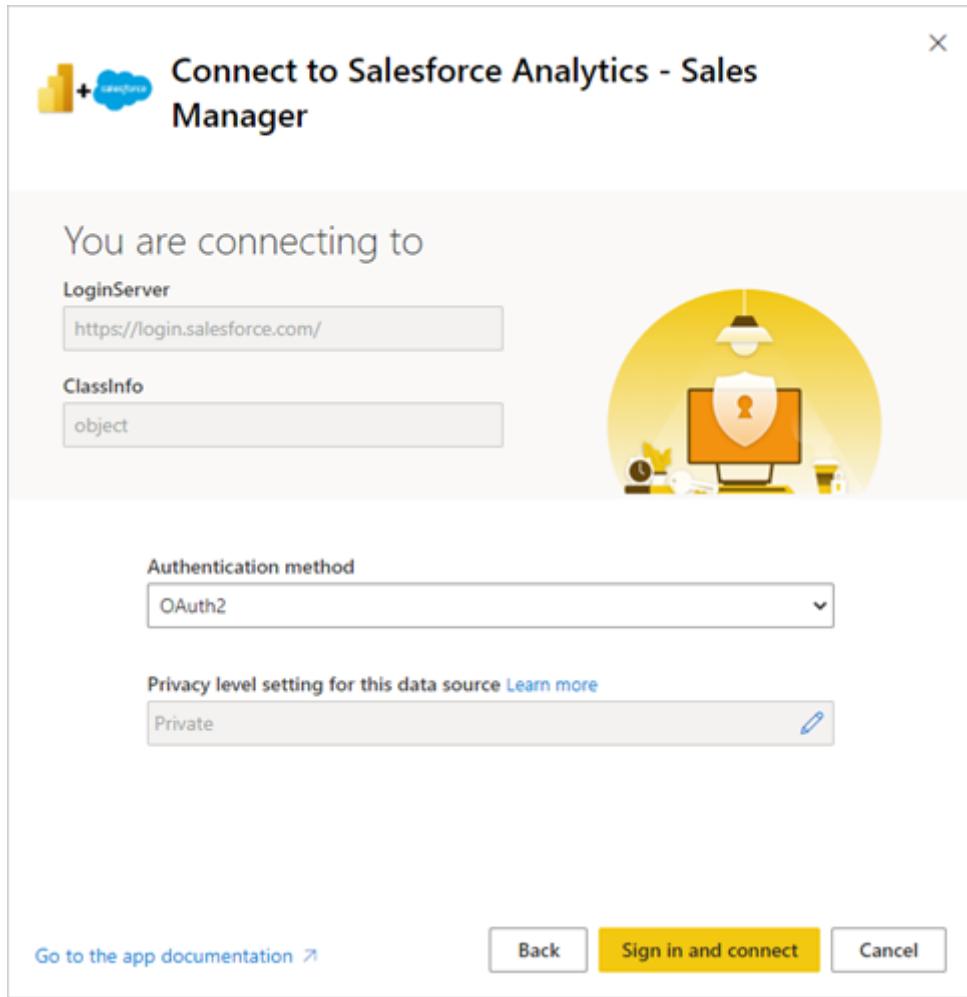
Advanced ▾

[Go to the app documentation ↗](#)

Next

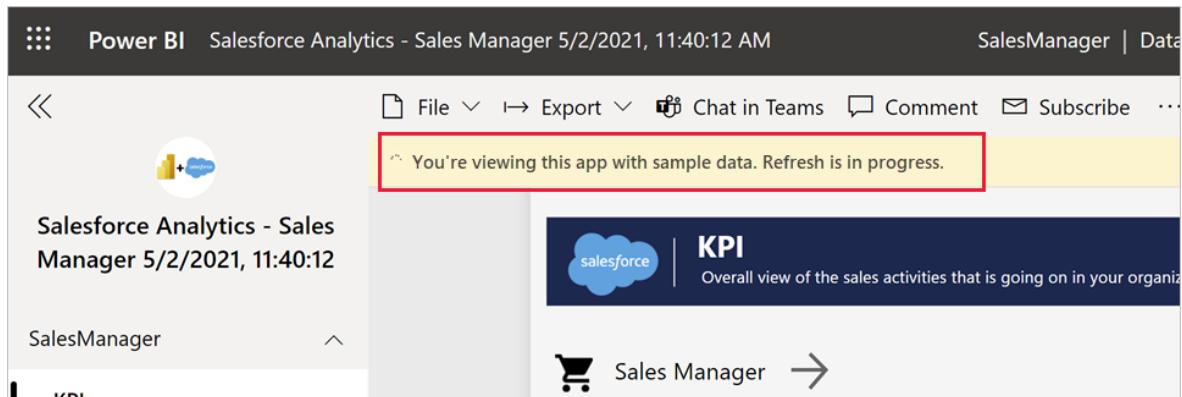
Cancel

4. The authentication method dialog will appear. Recommended values are prepopulated. Don't change these unless you have specific knowledge of different values. Click **Sign in and connect**.



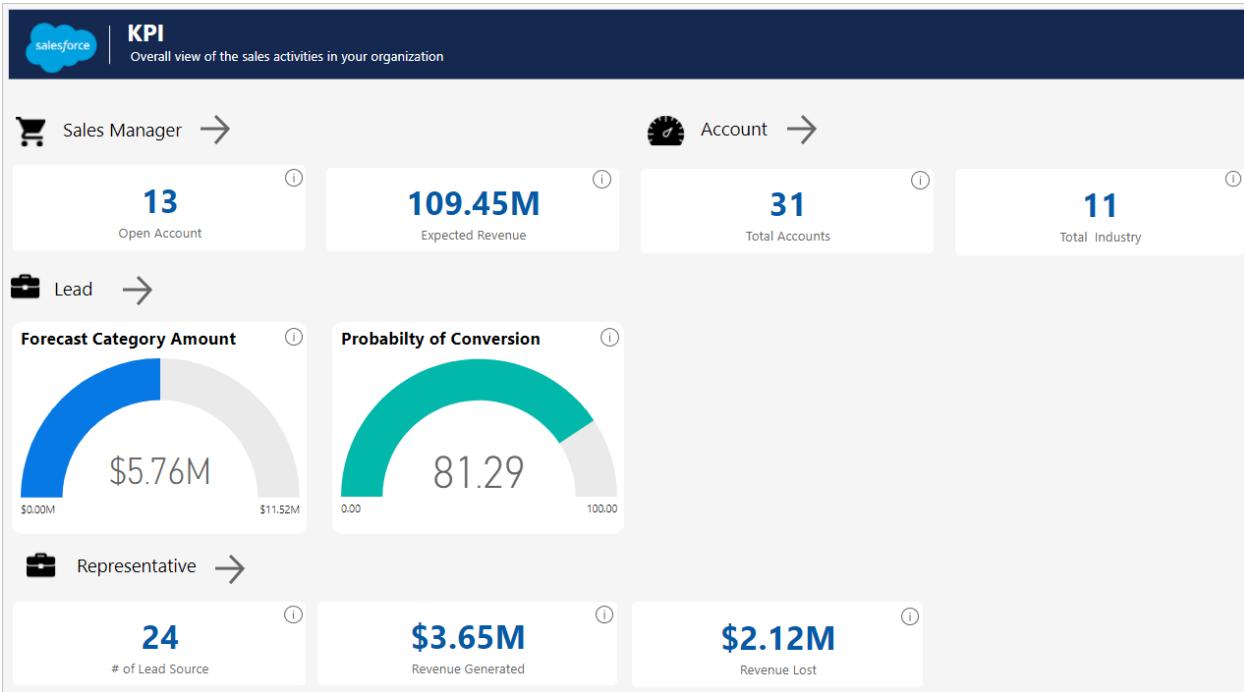
## 5. When prompted, sign into Salesforce.

The report will connect to the data sources and be populated with up-to-date data. During this time you will see sample data and that refresh is in progress.



## What does the KPI dashboard tell us?

When you open the app, you'll see the KPI dashboard. The KPI dashboard shows us an overall view of the key metrics from all the dashboards. Click the arrows to get to the individual dashboards.



## What does the Sales Manager dashboard tell us?

The Sales Manager dashboard and underlying report focus on a typical sales challenge: providing a total sales analysis over a certain period.

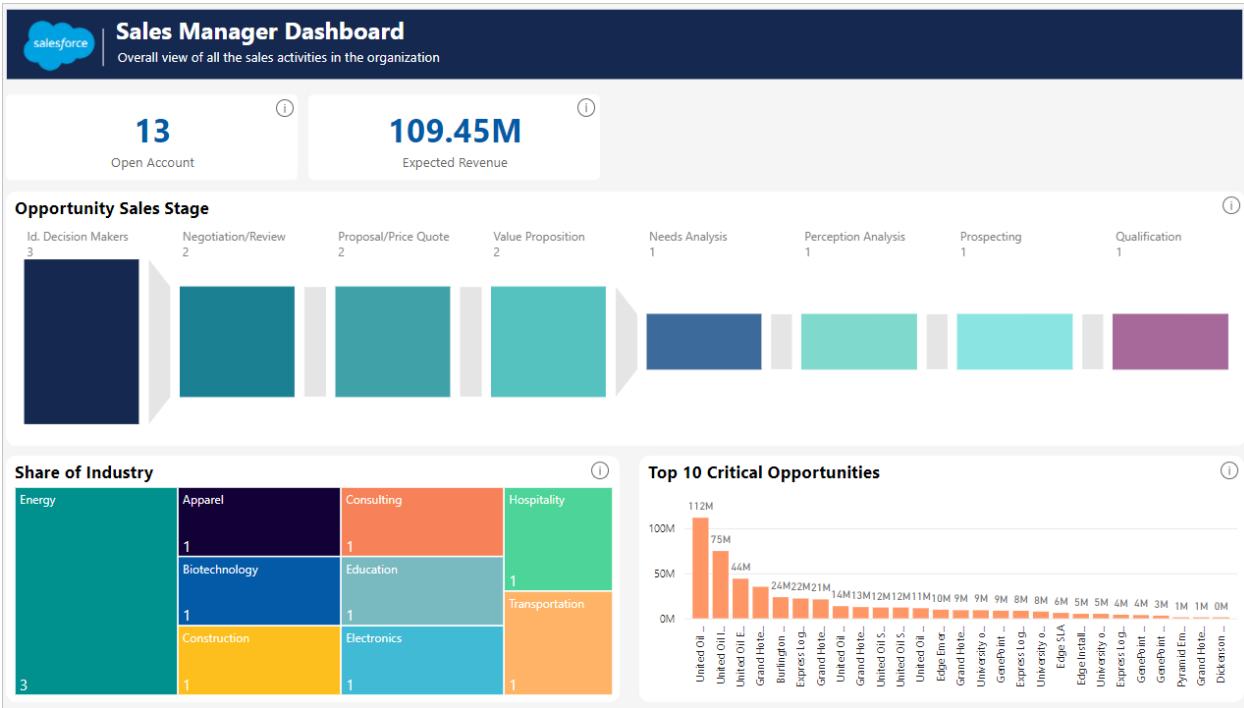
Using the dashboard to look at the sample data, we're going see how to

- Understand how much we can generate from the total open opportunities that a Sales company has.
- Identify which areas we should focus on in the total sales life cycle.
- Spot loopholes we should look into.

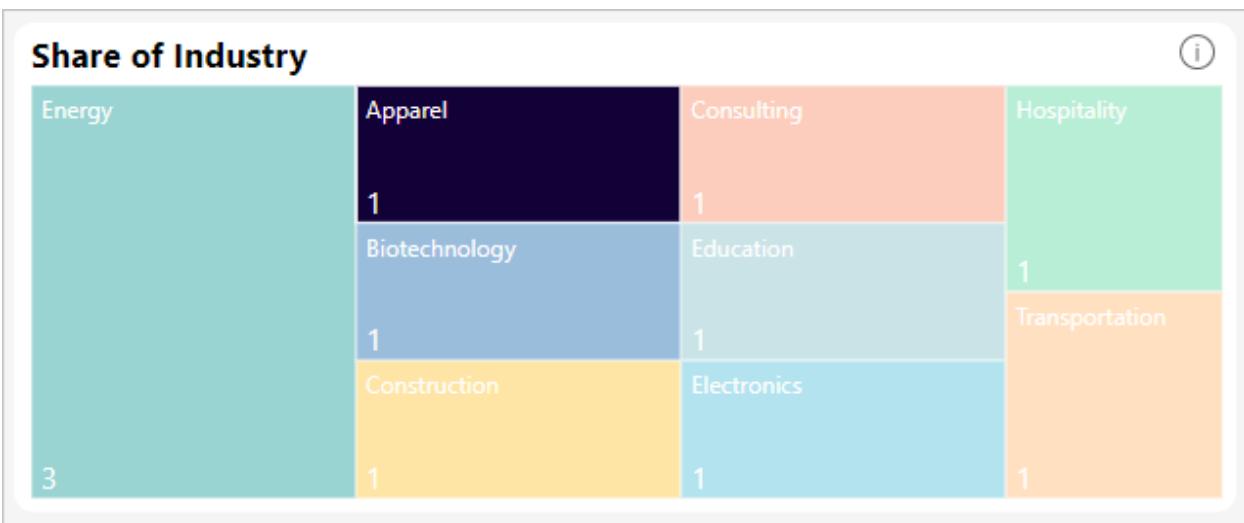
Now let's look at the various components of the dashboard.

The top two visuals show us the total number of open opportunities we have and the revenue we can expect to generate from them.

The Opportunity Sales Stage visual shows the position of the opportunities in the sales pipeline. Select any stage on the sales pipeline stage to see its impact on the whole sales process. Now we can analyze the revenue that corresponds with that stage.

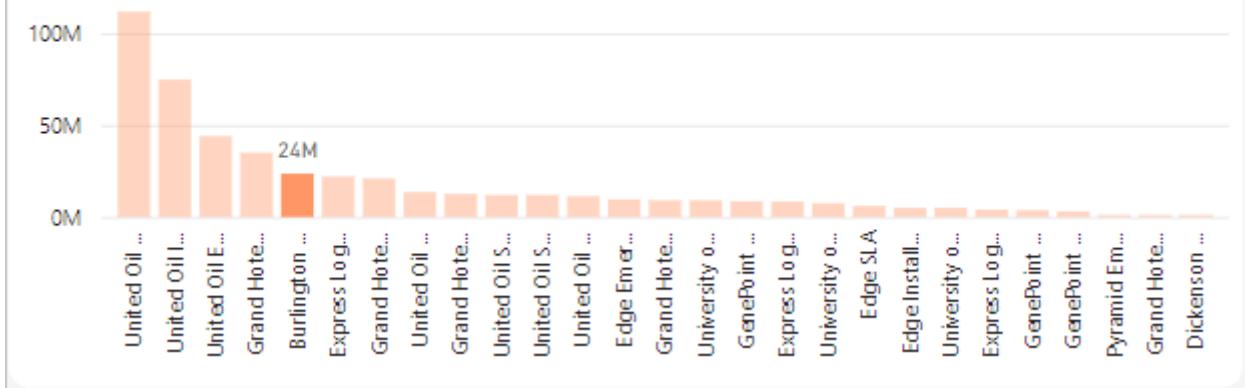


If you click on any one of the industries in the Share of Industry visual, it shows the share of the industry your sales team is working on. Let's click on "Apparel" for example.



When we click on "Apparel", we see that the account we have in this sector is one of the top ten critical opportunities with respect to revenue. We can conclude that this is an opportunity that deserves our focus on a priority basis, since its expected revenue is among the highest as compared to other accounts.

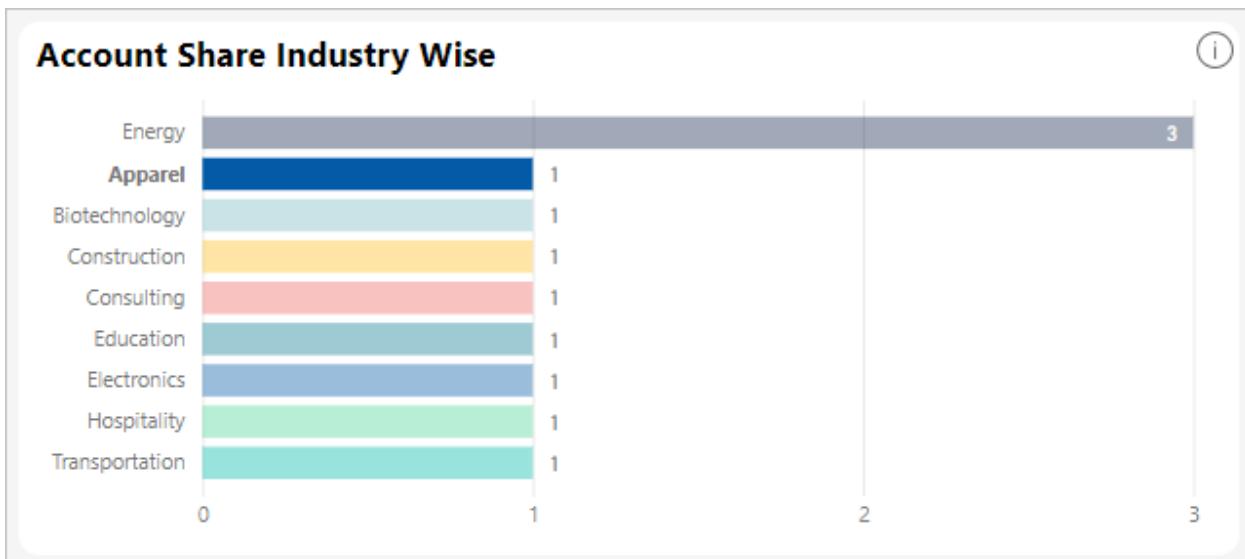
## Top 10 Critical Opportunities



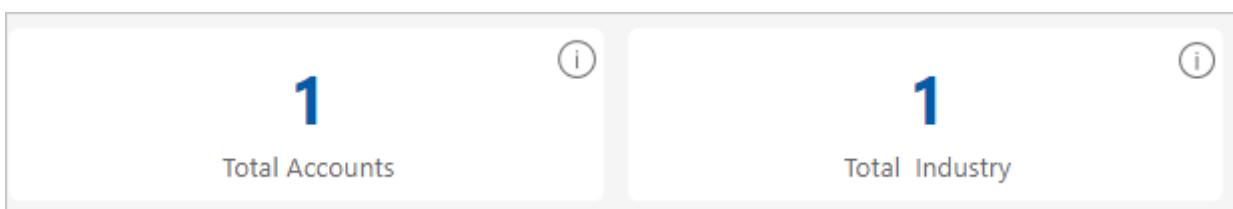
## What does the Account dashboard tell us?

The Account dashboard lets you oversee how you are performing in all your accounts. It tells you which are your most profitable accounts. Let's analyze the accounts for apparel.

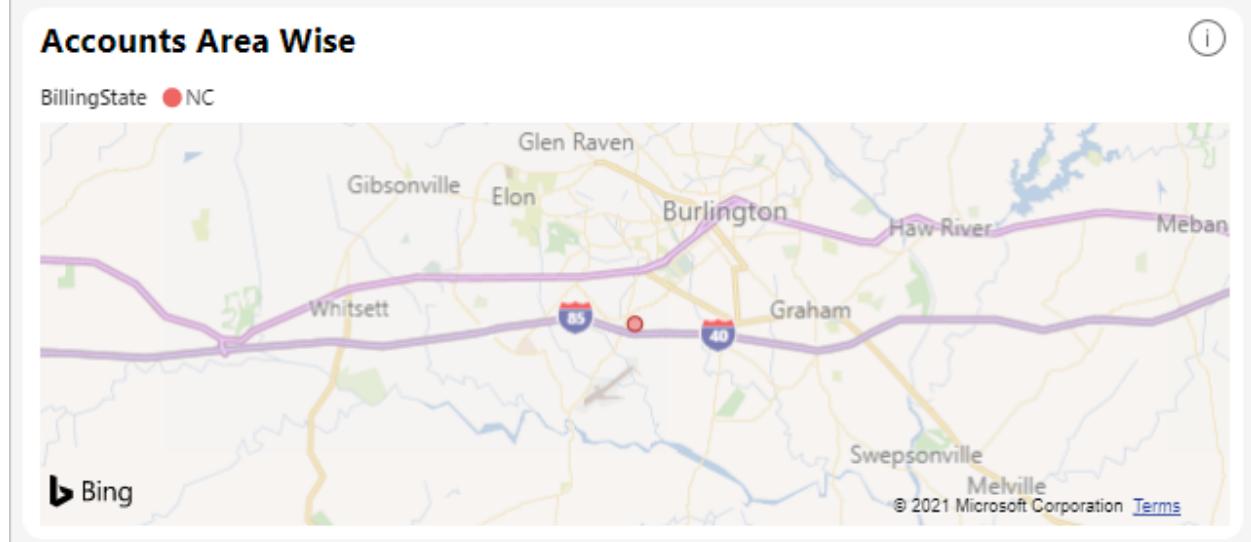
On the dashboard, select "Apparel" on the Account dashboard's Account Share Industry Wise visual.



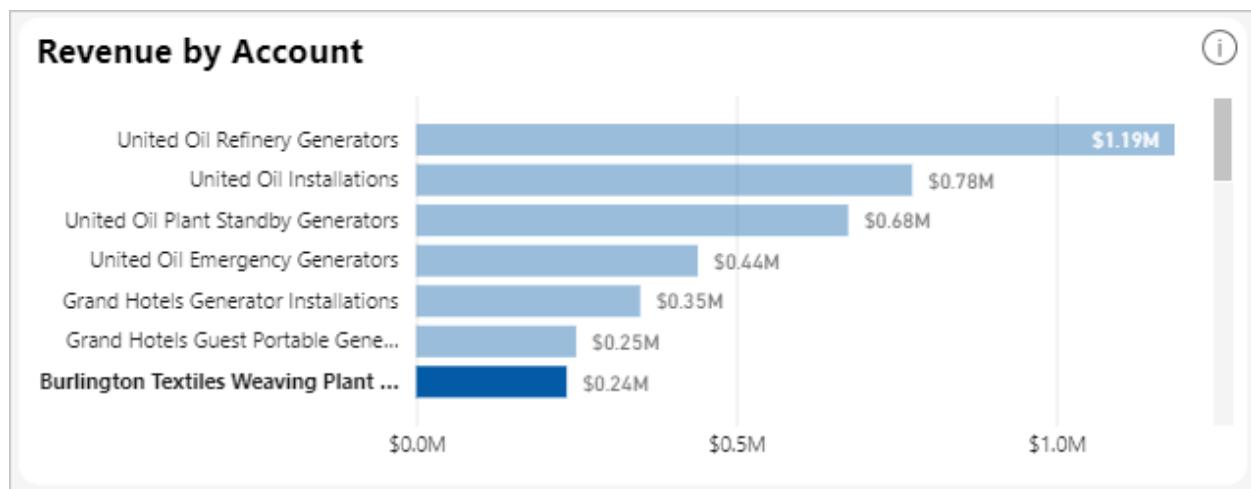
You'll see the tiles get updated. Notice that there is one industry in apparel, for which we have 1 account.



If we look at the Account Area Wise map visual, you'll see the area where we have the apparel account.



Let's look at the revenue for the account under apparel. In the Revenue by Account visual you can see that the account for apparel is highlighted, and shows the revenue being generated.



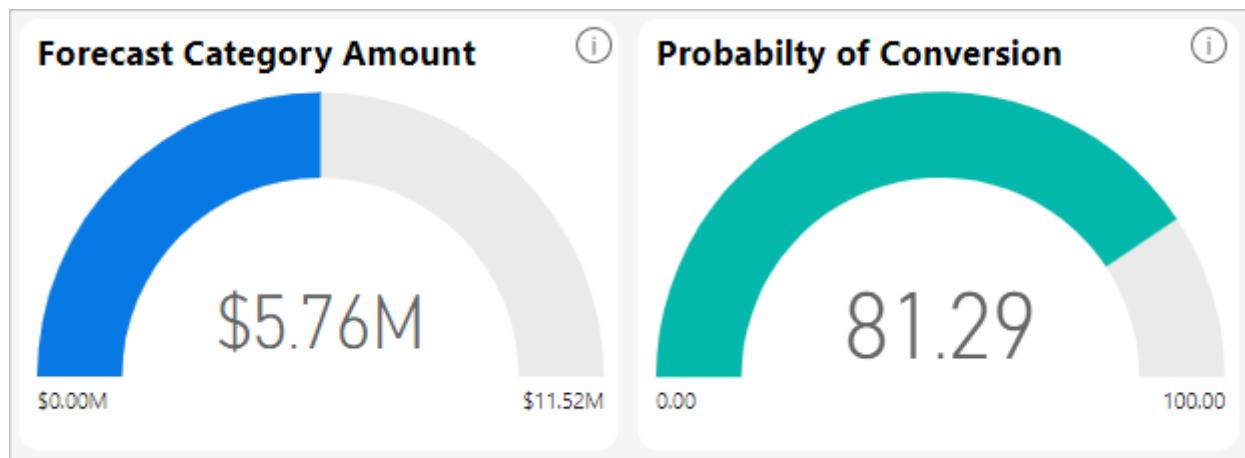
We can also see a comparison of revenue won vs lost. If we hover over a bar, we see the exact revenue won out of the total revenue.



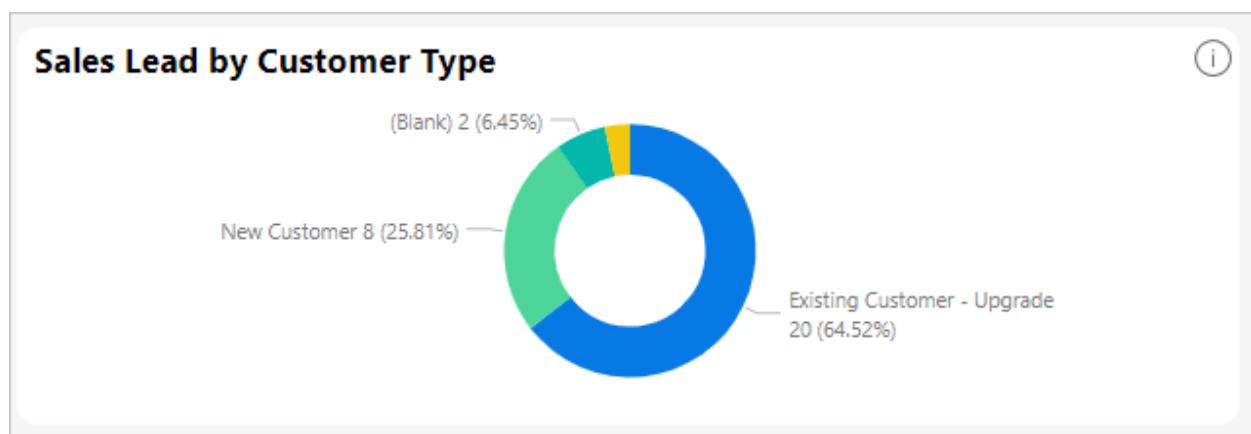
# What does the Lead dashboard tell us?

The Lead dashboard lets you see what the sources of your leads are. It tells you which are your most profitable sources of lead.

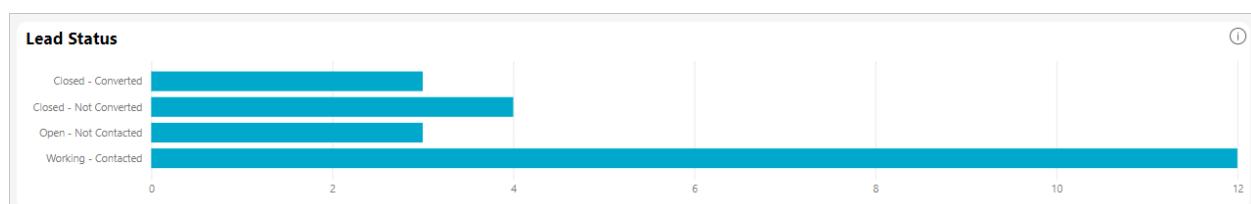
Look at the Probability of Conversion visual to examine what the probability is that the lead from a source is going to be converted. For example, selecting "External Referral" in the Leads by Source visual shows that the probability of conversion is 90.00%, while its total forecasted amount is 1.65 million dollars.



Similarly, you can also see the distribution of customers in the sales lead by looking at the Sales Lead by Customer Type visual. You can see it for a single lead by hovering over it. So, for "External Referral", you can see that there is one new customer belonging to the "New Business" category.



We can also see the overall number of lead statuses that we have, and in what stage they are.

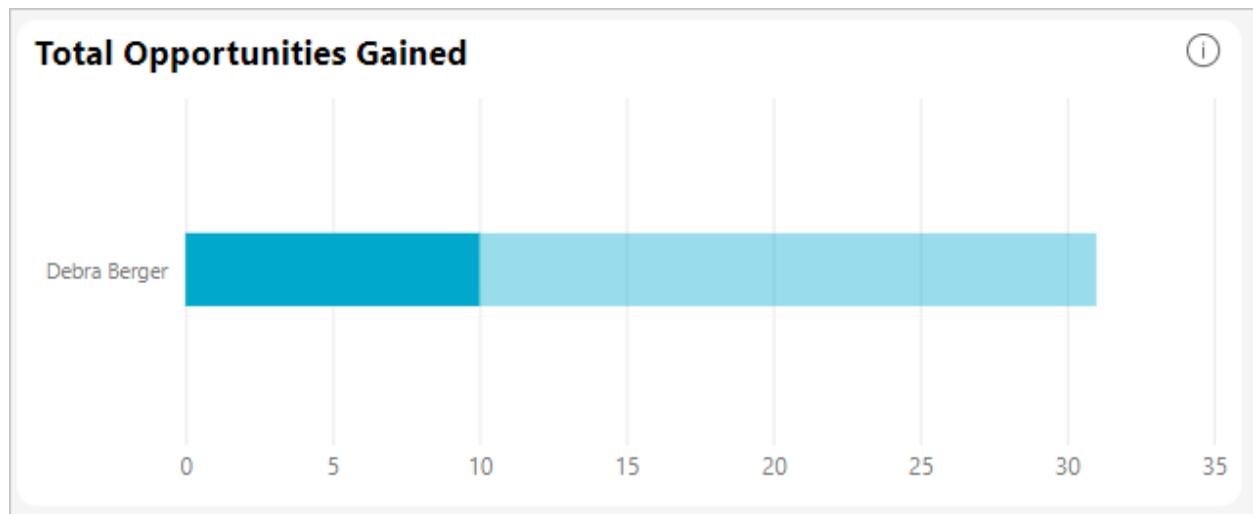


# What does Representative dashboard tell us?

The Representative dashboard lets you measure the performance of the sales representatives by a number of matrices.

We can see all the industries a sales representative works in, and all the representatives of an industry.

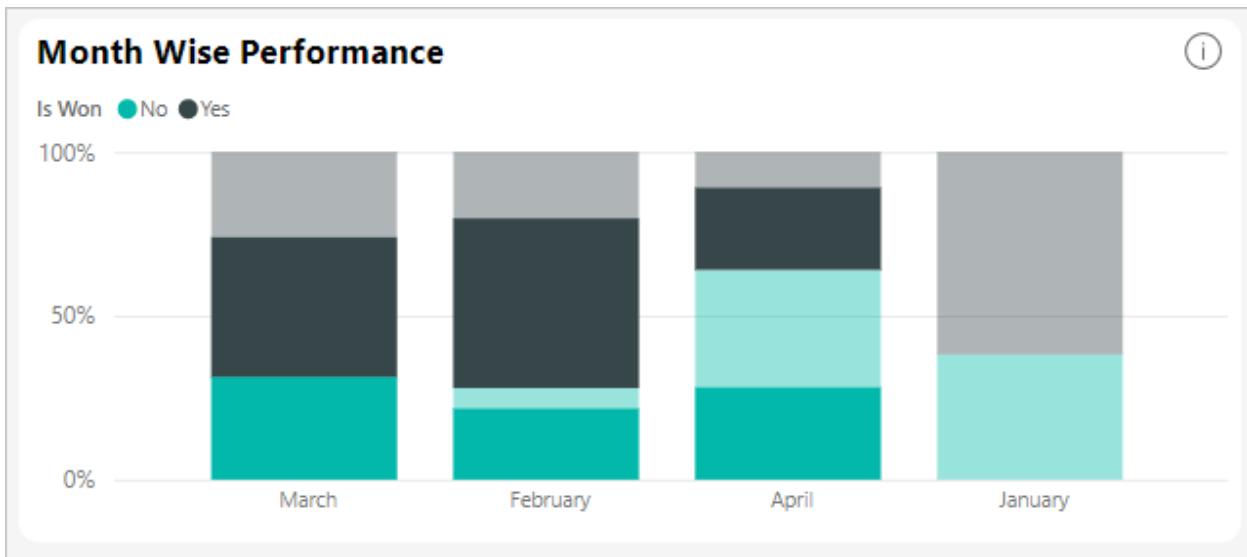
To see the performance of the sales representatives, we selected "Energy" as the industry from the Share of Industry visual. Then, in the Total Opportunities Gained visual, we can see the name of the sales representative and the number of opportunities which belong to the selected industry.



The top three visuals show us the number of leads generated by the sales representatives and the revenue they have generated and lost.



At same time, we can use the Month Wise Performance visual to measure the month-wise performance of the sales representatives.



## System requirements and considerations

- Connected with a production Salesforce account that has API access enabled.
- Permission granted to the Power BI app during sign in.
- The account has sufficient API calls available to pull and refresh the data.
- A valid authentication token is required for refresh. Salesforce has a limit of five authentication tokens per application so make sure you've five or less Salesforce data sets imported.
- The Salesforce Reports API has a restriction that supports up to 2,000 rows of data.

## Troubleshooting

- If you encounter any errors, review the requirements above.
- Signing in to a custom or sandbox domain isn't currently supported.
- [Salesforce connector reference](#)

## "Unable to connect to the remote server" message

If you get an "Unable to connect to the remote server" message when trying to connect to your Salesforce account, see this solution on the following forum: [Salesforce Connector sign in Error Message: Unable to connect to the remote server ↗](#)

## Next steps

- [What are Power BI template apps](#)
- [Create a template app in Power BI](#)
- [Install and distribute template apps in your organization](#)

- Questions? Try asking the Power BI Community 

# Connect to Smartsheet with Power BI

Article • 06/19/2023

This article walks you through pulling your data from your Smartsheet account with a Power BI template app. Smartsheet offers an easy platform for collaboration and file sharing. The Smartsheet template app for Power BI provides a dashboard, reports, and dataset that show an overview of your Smartsheet account. You can also use [Power BI Desktop](#) to connect directly to individual sheets in your account.

After you've installed the template app, you can change the dashboard and report. Then you can distribute it as an app to colleagues in your organization.

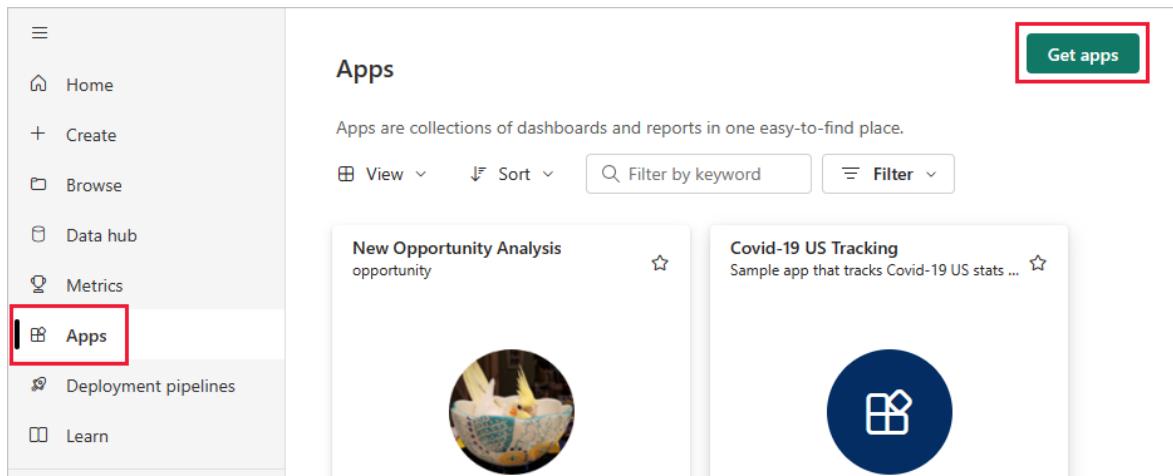
Connect to the [Smartsheet template app](#) for Power BI.

## ⓘ Note

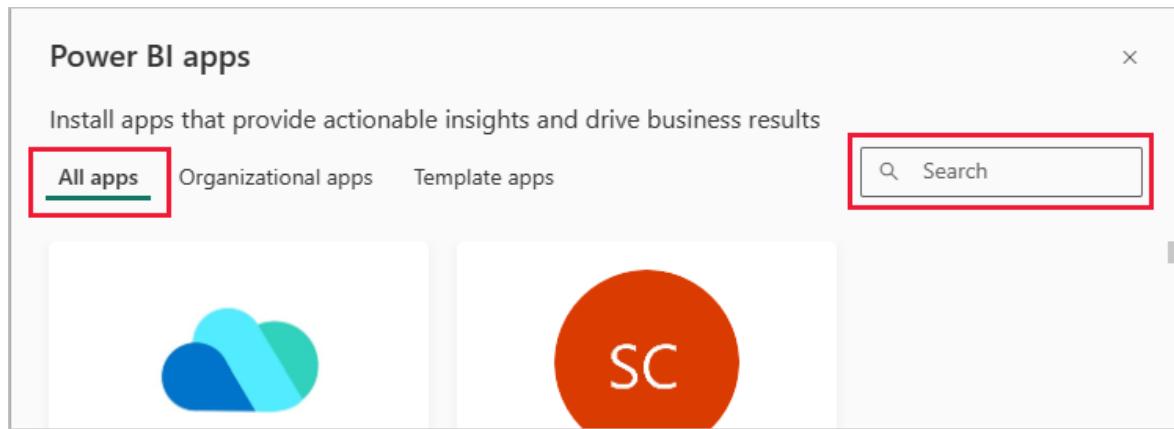
A Smartsheet admin account is preferred for connecting and loading the Power BI template app as it has additional access.

## Install the app

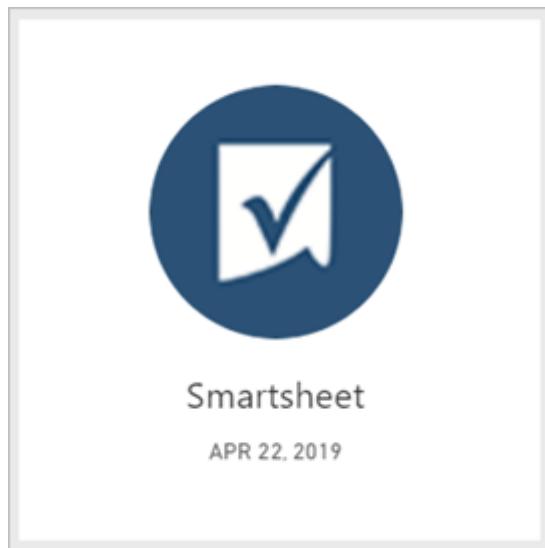
1. Select Apps in the navigation pane, then choose Get apps in the upper-right corner.



2. In Power BI apps, select the Apps tab, and search for the service you want.



1. Select **Smartsheet > Get it now.**
2. In **Install this Power BI App?** select **Install**.
3. In the **Apps** pane, select the **Smartsheet** tile.



## Connect to your Smartsheet data source

1. Select the Smartsheet tile on your Apps page to open the app. The app opens, showing sample data.
2. Select the **Connect your data** link on the banner at the top of the page.
3. For Authentication Method, select **oAuth2 > Sign In**.

When prompted, enter your Smartsheet credentials and follow the authentication process.

# Connect to Smartsheet



To start using your Smartsheet data in Power BI, follow the prompts below.

## Authentication method

oAuth2

Sign in

Cancel

A screenshot of a web browser window titled "Log In | Smartsheet - Google Chrome". The address bar shows the URL: "Smartsheet.com, Inc. [US] https://app.smartsheet.com/b/authorize?response\_type=code&state=app.powerbi.com&client\_id=bqezobx83hxji". The page content displays a blue sidebar on the left with the text "This app requires you to log in to Smartsheet" and an icon of a bar chart. To the right is the main login form with the Smartsheet logo, fields for "Email" and "Continue", a "Forgot password?" link, and social login options for Google and Facebook. At the bottom, there's a "Don't have a login? Sign up" link and a "Privacy Policy" link.

This app requires you to log in to Smartsheet

Power BI  
Power BI allows you to easily explore and visualize your Smartsheet data.  
[Learn More](#)

smartsheet

Email

Continue

Forgot password?

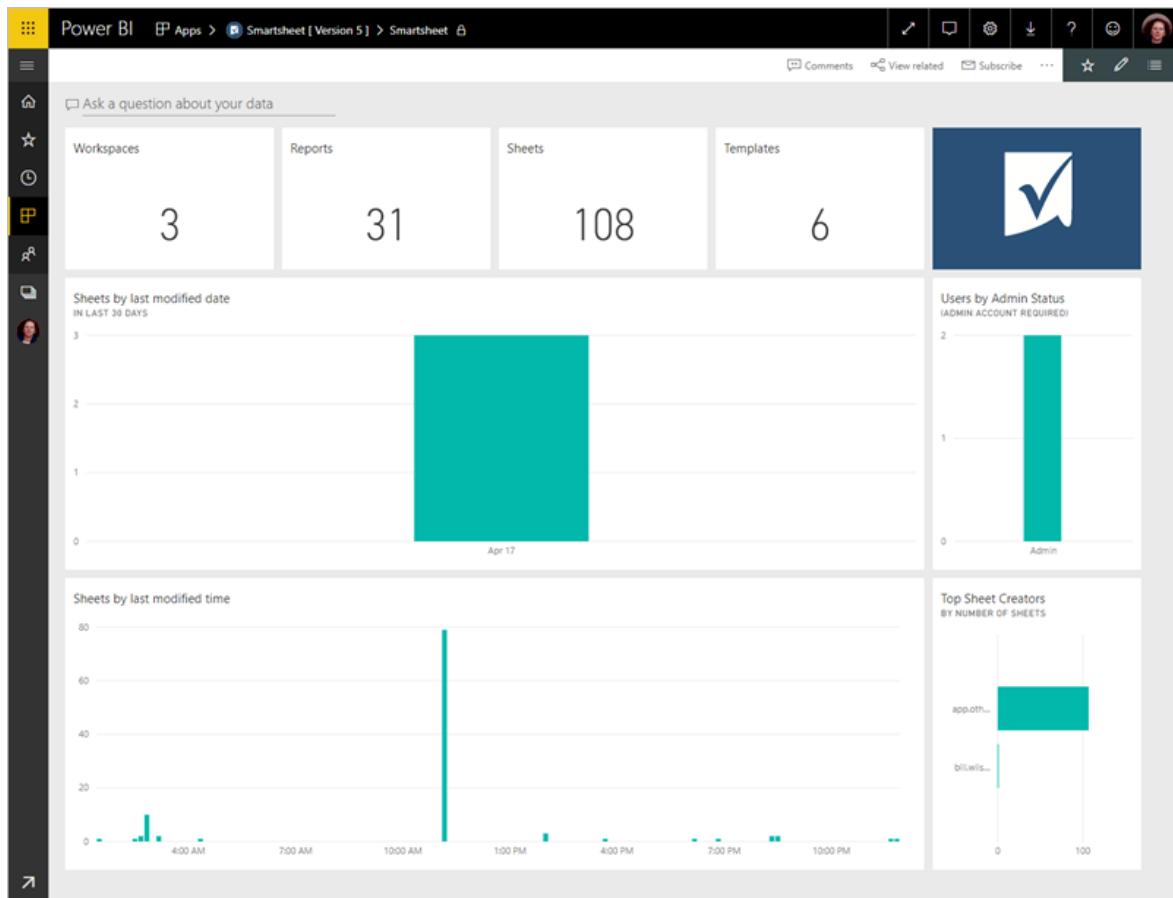
Or log in with

g f

Don't have a login? [Sign up](#)

Privacy Policy

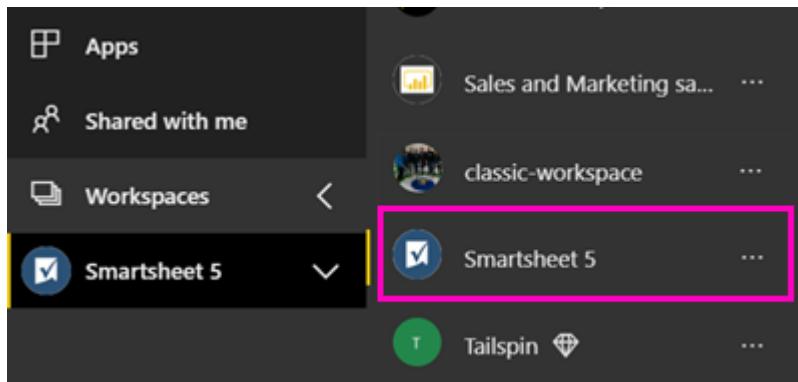
4. After Power BI imports the data, the Smartsheet dashboard opens.



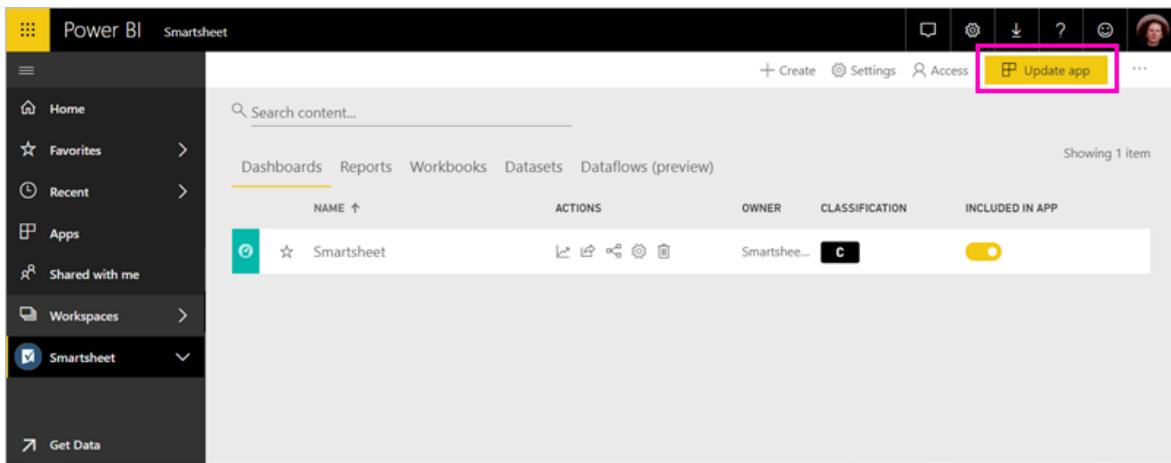
## Modify and distribute your app

You've installed the Smartsheet template app. That means you've also created the Smartsheet workspace. In the workspace, you can change the report and dashboard, and then distribute it as an *app* to colleagues in your organization.

1. To view all the contents of your new Smartsheet workspace, in the nav pane, select **Workspaces > Smartsheet**.



This view is the content list for the workspace. In the upper-right corner, you see **Update app**. When you're ready to distribute your app to your colleagues, that's where you'll start.



2. Select **Reports** and **Datasets** to see the other elements in the workspace.

Read about [distributing apps](#) to your colleagues.

## What's included

The Smartsheet template app for Power BI includes an overview of your Smartsheet account, such as the number of workspaces, reports, and sheets you have, when they're modified etc. Admin users also see some information around the users in their system, such as top sheet creators.

To connect directly to individual sheets in your account, you can use the Smartsheet connector in the [Power BI Desktop](#).

## Next steps

- [Create workspaces in Power BI](#)
- [Install and use apps in Power BI](#)
- [Connect to Power BI apps for external services](#)
- Questions? [Try asking the Power BI Community](#)

# Connect to Zendesk with Power BI

Article • 07/25/2023

This article walks you through pulling your data from your Zendesk account with a Power BI template app. The Zendesk app offers a Power BI dashboard and a set of Power BI reports that provide insights about your ticket volumes and agent performance. The data is refreshed automatically once a day.

After you've installed the template app, you can customize the dashboard and report to highlight the information you care about most. Then you can distribute it as an app to colleagues in your organization.

Connect to the [Zendesk template app](#) or read more about the [Zendesk integration](#) with Power BI.

After you've installed the template app, you can change the dashboard and report. Then you can distribute it as an app to colleagues in your organization.

## Note

You need a Zendesk Admin account to connect. More details on requirements below.

## Warning

Before Oct 15, 2019, the Zendesk Support Search API allowed for a total of 200,000 results to be received through pagination of large queries. To align search usage with its intended scope, Zendesk now limits the maximum number of results returned to 1,000 total results, with a maximum of 100 results per page. However, the current Power BI Zendesk connector can still create API calls that exceed these new limits, resulting in possibly misleading results.

## Install the app

1. Select **Apps** in the navigation pane, then choose **Get apps** in the upper-right corner.

The screenshot shows the Power BI Apps interface. On the left, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, and Apps (which is highlighted with a red box). The main area is titled 'Apps' and contains a brief description: 'Apps are collections of dashboards and reports in one easy-to-find place.' Below this are filter options ('View', 'Filter by keyword', 'Filter') and a table of apps. The first app listed is 'Sales and Marketing sample' by MOD Administrator, published on 7/9/23 at 1:07:50 PM, categorized as a 'Template app'. A large green 'Get apps' button is located in the top right corner.

2. In Power BI apps, select the Apps tab, and search for the service you want.

The screenshot shows the 'Power BI apps' search interface. It features a search bar at the top right, a tab navigation bar with 'All apps', 'Organizational apps', and 'Template apps' (which is highlighted with a red box), and a 'Sort by: Popularity' dropdown. Below this is a section titled 'Browse available apps from AppSource' with a sub-instruction: 'Save time by connecting your own data to a prebuilt report that you can personalize and share.' To the right is a diagram illustrating data connectivity. At the bottom, there are three app tiles: a bar chart tile, a Microsoft 365 tile, and a GitHub tile.

1. Select Zendesk > Get it now.

2. When prompted, select Install. Once the app has installed, you'll see it listed on your Apps page.

## Connect to your Zendesk data source

1. Select the Zendesk tile on your Apps page to open the app. The app opens, showing sample data.
2. Select the **Connect your data** link on the banner at the top of the page.
3. Provide the URL associated with your account. The URL has the form <https://company.zendesk.com>. See details on [finding these parameters](#) below.



## Connect to Zendesk

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

### Parameters

Make sure all required (\*) parameters are filled in before connecting to your data.

#### ZendeskURL \*

Enter your zendesk cloud url: https://[Your zendesk cloud].zendesk.com

For example: For example: "https://[Your zendesk cloud].zendesk.com"

Advanced ▾

[Go to the app documentation ↗](#)

Next

Cancel

4. When prompted, enter your Zendesk credentials. Select **oAuth 2** as the Authentication Mechanism and select **Sign In**. Follow the Zendesk authentication flow. (If you're already signed in to Zendesk in your browser, you may not be prompted for credentials.)

**Note**

This template app requires that you connect with a Zendesk Admin account.



## Connect to Zendesk

X

You are connecting to

ExtensionDataSourceKind

zendesk

ExtensionDataSourcePath

<https://myzendeskcloud.zendesk.com/>



Authentication method

OAuth2

Privacy level setting for this data source [Learn more](#)

Private



[Go to the app documentation](#)

Back

Sign in and connect

Cancel

5. Select Allow to allow Power BI to access your Zendesk data.

zendesk



Power BI  
by Microsoft

**Allow Power BI to access your Zendesk account?**

Power BI is a complete self-service business intelligence (BI) solution, allowing users to connect, explore and monitor their data.

**This application would be able to:**

- Read all data.
- Write all data.

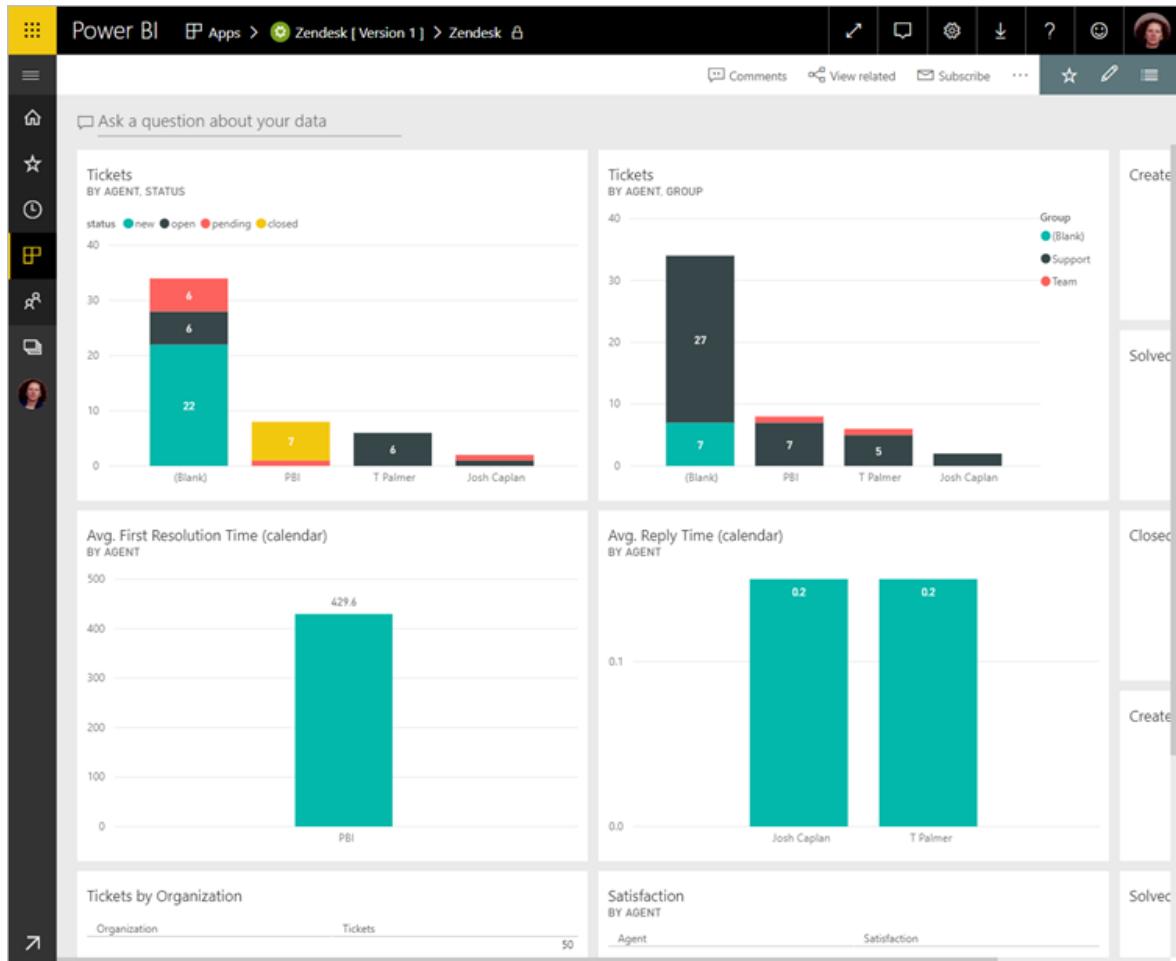
Deny

Allow

6. Select Connect to begin the import process.

7. After Power BI imports the data, you see the content list for your Zendesk app: a new dashboard, report, and dataset.

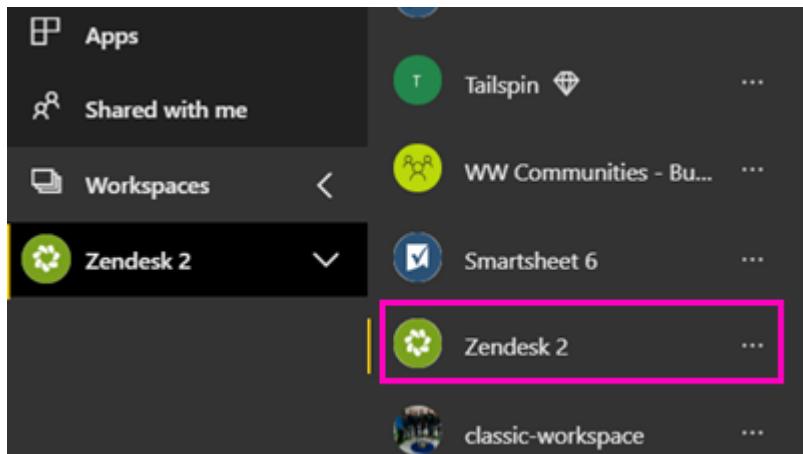
8. Select the dashboard to start the exploration process.



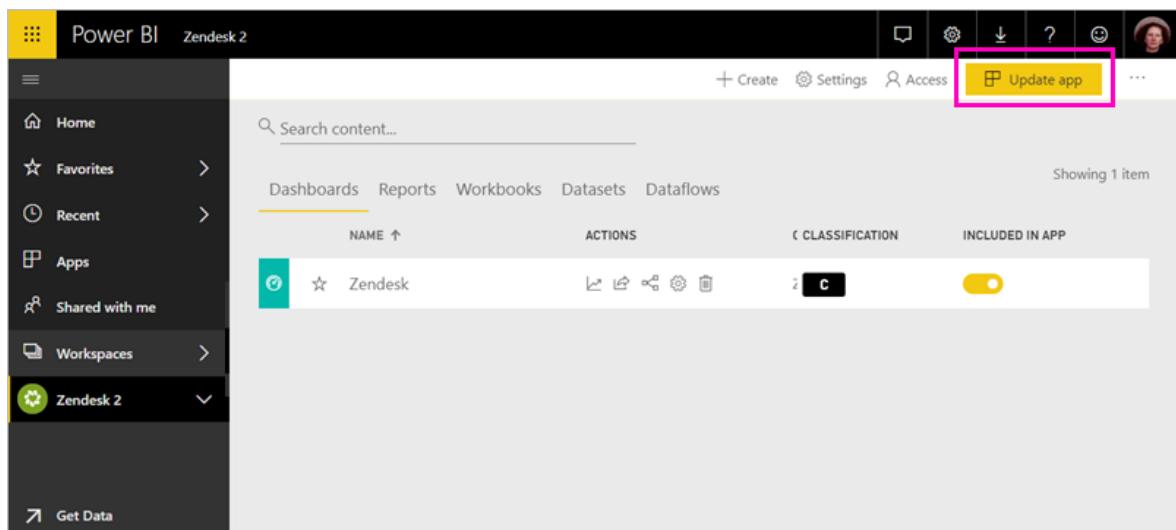
## Modify and distribute your app

You've installed the Zendesk template app. That means you've also created the Zendesk workspace. In the workspace, you can change the report and dashboard, and then distribute it as an *app* to colleagues in your organization.

1. To view all the contents of your new Zendesk workspace, in the nav pane, select **Workspaces > Zendesk**.



This view is the content list for the workspace. In the upper-right corner, you see **Update app**. When you're ready to distribute your app to your colleagues, that's where you'll start.



2. Select **Reports** and **Datasets** to see the other elements in the workspace.

Read about [distributing apps](#) to your colleagues.

## System requirements

A Zendesk Administrator account is required to access the Zendesk template app. If you're an agent or an end user and are interested in viewing your Zendesk data, add a suggestion and review the Zendesk connector in the [Power BI Desktop](#).

## Finding parameters

Your Zendesk URL will be the same as the URL you use to sign into your Zendesk account. If you're not sure of your Zendesk URL, you can use the Zendesk [login help](#).

# Troubleshooting

If you're having issues connecting, check your Zendesk URL and confirm you're using a Zendesk administrator account.

## Next steps

- [Create workspaces in Power BI](#)
- [Install and use apps in Power BI](#)
- [Connect to Power BI apps for external services](#)
- Questions? [Try asking the Power BI Community](#)↗

# Guidance for deploying a data gateway for the Power BI service

Article • 01/17/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

This article provides guidance and considerations for deploying a data gateway for the Power BI service in your network environment.

For information about how to download, install, configure, and manage the on-premises data gateway, see [What is an on-premises data gateway?](#). You can also find out more about the on-premises data gateway and Power BI by visiting the [Microsoft Power BI blog](#) and the [Microsoft Power BI Community](#) site.

## Installation considerations for the on-premises data gateway

Before you install the on-premises data gateway for your Power BI cloud service, there are some considerations to keep in mind. The following sections describe these considerations.

### Number of users

The number of users who consume a report that uses the gateway is an important metric in your decision about where to install the gateway. Here are some questions to consider:

- Do users use these reports at different times of the day?
- What types of connections do they use: DirectQuery or Import?
- Do all users use the same report?

If all the users access a given report at the same time each day, make sure that you install the gateway on a machine that's capable of handling all those requests. See the

following sections for performance counters and minimum requirements that can help you determine whether a machine is adequate.

A constraint in the Power BI service allows only *one* gateway per *report*. Even if a report is based on multiple data sources, all such data sources must go through a single gateway. If a dashboard is based on *multiple* reports, you can use a dedicated gateway for each contributing report. In this way, you distribute the gateway load among the multiple reports that contribute to the single dashboard.

## Connection type

The Power BI service offers two types of connections: DirectQuery and Import. Not all data sources support both connection types. Many factors might contribute to your choice of one over the other, such as security requirements, performance, data limits, and data model sizes. To learn more about connection types and supported data sources, see the [list of available data source types](#).

Depending on which type of connection is used, gateway usage can be different. For example, try to separate DirectQuery data sources from scheduled refresh data sources whenever possible. The assumption is that they're in different reports and can be separated. Separating sources prevents the gateway from having thousands of DirectQuery requests queued up at the same time as the morning's scheduled refresh of a large-size data model that's used for the company's main dashboard.

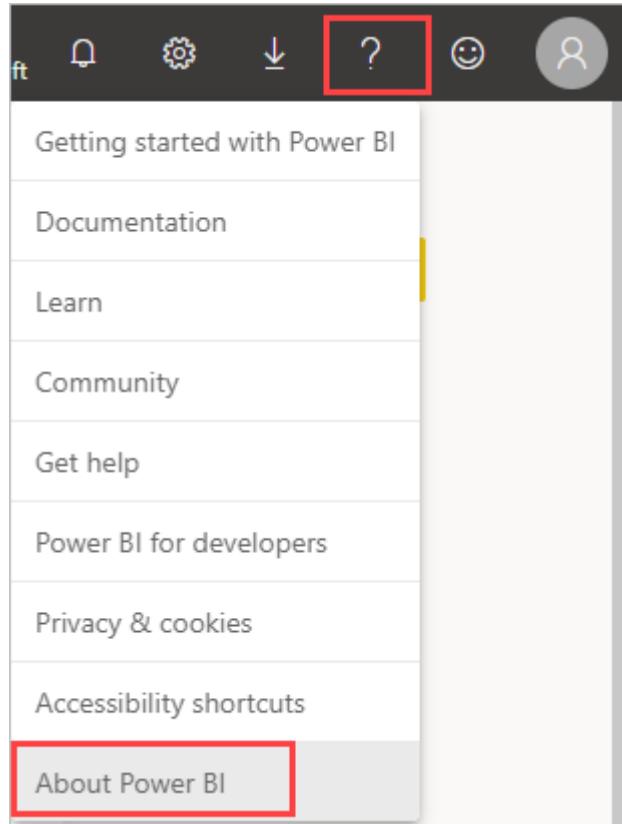
Here's what to consider for each option:

- **Scheduled refresh:** Depending on your query size and the number of refreshes that occur per day, you can choose to stay with the recommended minimum hardware requirements or upgrade to a higher performance machine. If a given query isn't folded, transformations occur on the gateway machine. As a result, the gateway machine benefits from having more available RAM.
- **DirectQuery:** A query is sent each time any user opens the report or looks at data. If you expect more than 1,000 users to access the data concurrently, make sure your computer has robust and capable hardware components. More CPU cores result in better throughput for a DirectQuery connection.

For the machine installation requirements, see the [on-premises data gateway installation requirements](#).

## Location

The location of the gateway installation can have significant effect on your query performance. Try to make sure that your gateway, data source locations, and the Power BI tenant are as close as possible to each other to minimize network latency. To determine your Power BI tenant location, in the Power BI service select the question mark (?) icon in the upper-right corner. Then select **About Power BI**.



If you intend to use the Power BI service gateway with Azure Analysis Services, be sure that the data regions in both match. For more information about how to set data regions for multiple services, watch [this video](#).

## Optimizing performance

By default, the gateway spools data before returning it to the dataset, potentially causing slower performance during data load and refresh operations. The default behavior can be overridden.

1. In the *C:\Program Files\On-Premises data*

*gateway\Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config* file, set the `StreamBeforeRequestCompletes` property to `True`, and then save.

JSON

```
<setting name="StreamBeforeRequestCompletes" serializeAs="String">
 <value>True</value>
</setting>
```

2. In **On-premises data gateway** > **Service Settings**, restart the gateway.

If installing the gateway on an Azure Virtual Machine, ensure optimal networking performance by configuring accelerated networking. To learn more, see [Create a Windows VM with accelerated networking](#).

## Next steps

- [Configure proxy settings](#)
- [Troubleshoot gateways - Power BI](#)
- [On-premises data gateway FAQ - Power BI](#)

More questions? Try the [Power BI Community](#).

# On-premises data gateway in-depth

Article • 01/17/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

We moved the information from this article to several articles across the Power BI and general docs. Follow the links under each heading to find the relevant content.

## How the gateway works

See [On-premises data gateway architecture](#).

## List of available data source types

See [Add or remove a gateway data source](#).

## Authentication to on-premises data sources

See [Authentication to on-premises data sources](#).

## Authentication to a live Analysis Services data source

See [Authentication to a live Analysis Services data source](#).

## Role-based security

See [Role-based security](#).

## Row-level security

See [Row-level security](#).

## What about Azure Active Directory?

See [Azure Active Directory](#).

## How do I tell what my UPN is?

See [How do I tell what my UPN is?](#).

## Map user names for Analysis Services data sources

See [Map user names for Analysis Services data sources](#).

## Synchronize an on-premises Active Directory with Azure Active Directory

See [Synchronize an on-premises Active Directory with Azure Active Directory](#).

## What to do next?

See the articles on data sources:

- [Add or remove a gateway data source](#)
- [Manage your data source - Analysis Services](#)
- [Manage your data source - SAP HANA](#)
- [Manage your data source - SQL Server](#)
- [Manage your data source - Oracle](#)
- [Manage your data source - Import/Scheduled refresh](#)

## Where things can go wrong

See [Troubleshoot the on-premises data gateway](#) and [Troubleshoot gateways - Power BI](#).

## Sign in account

See [Sign in account](#).

# Windows Service account

See [Change the on-premises data gateway service account](#).

## Ports

See [Ports](#).

## Forcing HTTPS communication with Azure Service Bus

See [Force HTTPS communication with Azure Service Bus](#).

## Support for TLS 1.2

See [TLS 1.2 for gateway traffic](#).

## How to restart the gateway

See [Restart a gateway](#).

## Next steps

- [What is the on-premises data gateway?](#)
- More questions? [Try the Power BI Community](#) ↗

# Use a personal gateway in Power BI

Article • 01/17/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

The on-premises data gateway (personal mode) is a version of the on-premises data gateway that works only with Power BI. You can use a personal gateway to install a gateway on your own computer and get access to on-premises data.

## ⓘ Note

Each Power BI user can have only one personal mode gateway running. If the same user installs another personal mode gateway, even on a different computer, the most recent installation replaces the existing previous installation.

## On-premises data gateway vs. on-premises data gateway (personal mode)

The following table describes differences between an on-premises data gateway and an on-premises data gateway (personal mode).

	On-premises data gateway	On-premises data gateway (personal mode)
<b>Supports cloud services:</b>	Power BI, PowerApps, Azure Logic Apps, Power Automate, Azure Analysis Services, dataflows	None
<b>Runs under credentials:</b>	As configured by users who have access to the gateway	Your credentials for Windows authentication, or credentials you configure for other authentication types
<b>Can install only as computer admin</b>	Yes	No

	<b>On-premises data gateway</b>	<b>On-premises data gateway (personal mode)</b>
<b>Centralized gateway and data source management</b>	Yes	No
<b>Can import data and schedule refresh</b>	Yes	Yes
<b>DirectQuery support</b>	Yes	No
<b>LiveConnect support for Analysis Services</b>	Yes	No

## Install the on-premises data gateway (personal mode)

To install the on-premises data gateway (personal mode):

1. Download the on-premises data gateway [↗](#).
2. Open the installer, and select Next.
3. Select On-premises data gateway (personal mode), and then select Next.



## On-premises data gateway installer

X

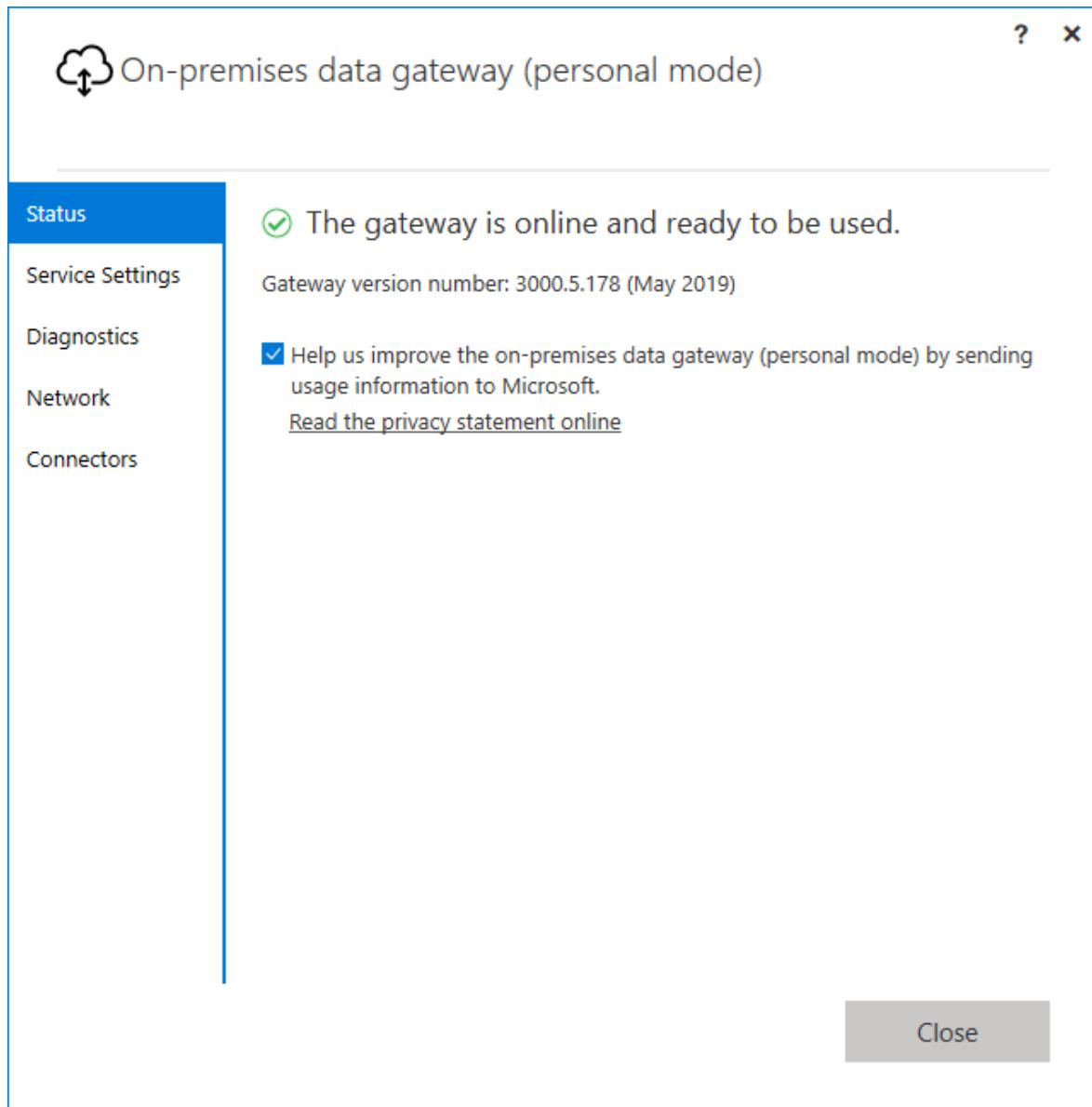
Choose the type of gateway you need.

- On-premises data gateway (recommended)
  - Can be shared and reused by multiple users
  - Can be used by Power BI, PowerApps, Logic Apps, and Microsoft Flow
  - Supports schedule refresh and live query for Power BI[Learn more](#)
- On-premises data gateway (personal mode)
  - Can only be used by you
  - Can only be used in Power BI
  - Only schedule refresh is supported[Learn more](#)

Next

Cancel

4. On the next screen, review the minimum requirements, verify or edit the installation path, and select the checkbox to accept the terms of use and privacy statement. Then select **Install**.
5. After the installation completes successfully, enter your email address under **Email address to use with this gateway**, and select **Sign in**.
6. After you sign in, the following screen appears. Select **Close** to close the installer.



## Use Fast Combine with the personal gateway

Fast Combine on a personal gateway helps you ignore specified privacy levels when you run queries. To enable Fast Combine for the on-premises data gateway (personal mode):

1. Use Windows File Explorer to open the file `<localappdata>\Microsoft\On-premises data gateway (personal mode)\Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config`.
2. At the end of the file, before `</Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.GatewayCoreSettings>`, add the following code, and then save the file.

XML

```
<setting name="EnableFastCombine" serializeAs="String">
 <value>true</value>
</setting>
```

3. The setting takes effect in approximately one minute. To confirm that Fast Combine is working properly, try an on-demand refresh in the Power BI service.

## Frequently asked questions (FAQ)

- **Question:** Can you run the on-premises data gateway (personal mode) side-by-side with the on-premises data gateway that used to be called the Enterprise gateway?

**Answer:** Yes, both gateways can run simultaneously.

- **Question:** Can you run the on-premises data gateway (personal mode) as a service?

**Answer:** No. The on-premises data gateway (personal mode) can run only as an application. To run a gateway as a service or in admin mode, use the [on-premises data gateway](#), which used to be called the Enterprise gateway.

- **Question:** How often does the on-premises data gateway (personal mode) update?

**Answer:** The personal gateway updates monthly.

- **Question:** Why does the personal gateway ask you to update your credentials?

**Answer:** Many situations can trigger a request for credentials. The most common scenario is that you reinstalled the on-premises data gateway (personal mode) on a different machine than your original Power BI personal gateway. There could also be an issue in the data source, or Power BI failed to make a test connection, or a timeout or system error occurred.

To update your credentials in the Power BI service, select the gear icon in the header and then choose **Settings**. On the **Datasets** tab, select the dataset, and then choose **Data source credentials**.

- **Question:** How long is a personal gateway offline during an upgrade?

**Answer:** Upgrading the personal gateway to a new version takes only few minutes.

- **Question:** Does the personal gateway support R and Python scripts?

**Answer:** Yes, personal mode supports R and Python scripts.

## Next steps

- Add or remove a gateway data source
- Configure proxy settings for an on-premises data gateway

More questions? Try the [Power BI Community](#).

# On-premises data gateway FAQ - Power BI

FAQ

## Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

## Do I need to upgrade the on-premises data gateway (personal mode)?

No, you can keep using the on-premises data gateway (personal mode) for Power BI.

## Are any special permissions required to install the gateway and manage it in the Power BI service?

No special permissions are required. You need to sign in with either a work or school email account.

## Can I upload Excel workbooks with Power Pivot data models that connect to on-premises data sources, and do I need a gateway for this scenario?

Yes, you can upload the workbook. No, you don't need a gateway. But, because the data will reside in the Excel data model, reports in Power BI based on the Excel workbook won't be live. To refresh reports in Power BI, you have to reupload an updated workbook each time. Or, use the gateway with scheduled refresh.

# If users share dashboards with a DirectQuery connection, will other users see the data even though they might not have the same permissions?

For a dashboard connected to Analysis Services, users will see only the data they have access to. If the users don't have the same permissions, they won't be able to see any data. For other data sources, all users will share the credentials entered by the admin for that data source.

# Why can't I connect to my Oracle server?

You might need to install the Oracle client and configure the *tnsnames.ora* file with the proper server information to connect to your Oracle server. The oracle client is a separate installation outside of the gateway. For more information, see [Install the Oracle client](#).

# Are R scripts supported?

R scripts are supported only for personal mode.

# Can I use msmdpump.dll to create custom effective username mappings for Analysis Services?

No. This use isn't supported.

# Can I use the gateway to connect to a multidimensional (OLAP) instance?

Yes. The on-premises data gateway supports live connections to both Analysis Services Tabular and Multidimensional models.

# What if I install the gateway on a computer in a different domain from my on-premises server that uses Windows authentication?

No guarantees. It depends on the trust relationship between the two domains. If two different domains are in a trusted domain model, the gateway might be able to connect to the Analysis Services server, and the effective username can be resolved. If not, you might encounter a sign-in failure.

# How can I find out what effective username is being passed to my on-premises Analysis Services server?

See [Troubleshoot gateways - Power BI](#).

## Next steps

- [Troubleshoot the on-premises data gateway](#)

More questions? Ask the [Power BI Community](#)↗.

# Add or remove a gateway data source

Article • 01/17/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

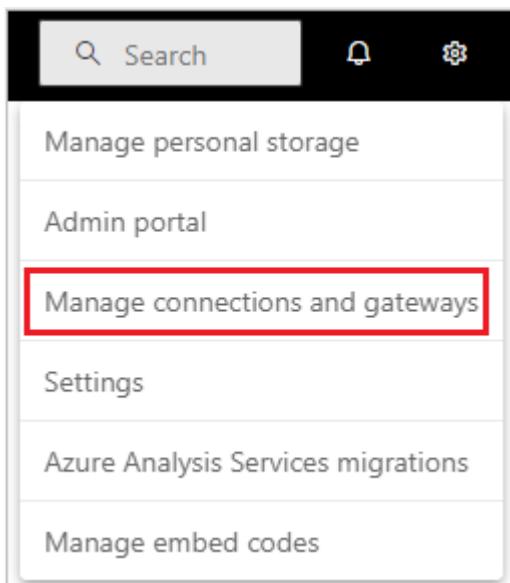
Power BI supports many [on-premises data sources](#), and each source has its own requirements. You can use a gateway for a single data source or multiple data sources. For this example, you learn how to add SQL Server as a data source. The steps are similar for other data sources.

You can also do most data sources management operations by using APIs. For more information, see [REST APIs \(Gateways\)](#).

If you don't have a gateway installed, [install an on-premises data gateway](#) to get started.

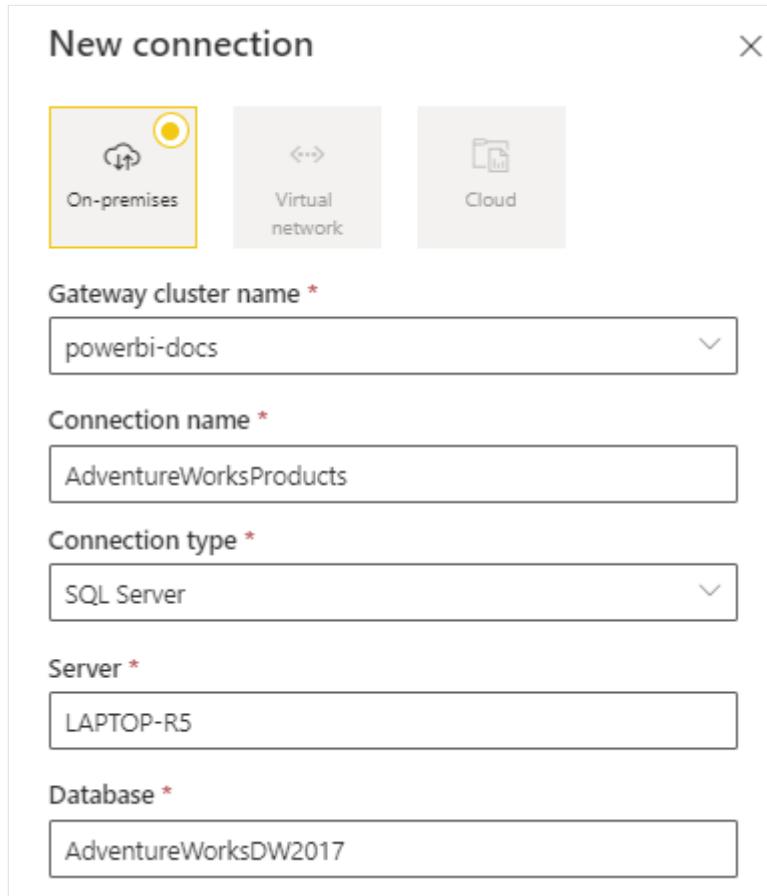
## Add a data source

1. From the page header in the Power BI service, select the **Settings** icon, and then select **Manage connections and gateways**.



2. Select **New** at the top of the screen to add a new data source.

3. On the **New connection** screen, select **On-premises**, provide the **Gateway cluster name** you want to create the connection on, provide a **Connection name**, and select the **Data Source Type**. For this example, choose **SQL Server**.
4. Enter information about the data source. For SQL Server, provide the **Server** and **Database**.



**ⓘ Note**

To use the data source for Power BI reports and dashboards, the server and database names must match between Power BI Desktop and the data source you add to the gateway.

5. Select an **Authentication Method** to use when connecting to the data source, **Basic**, **Windows**, or **OAuth2**. For SQL Server, choose **Windows** or **Basic (SQL Authentication)**. Enter the credentials for your data source.

**Authentication** ⓘ

**Authentication method \***

Windows

**Username \***

**Password \***

Skip test connection

If you selected **OAuth2** authentication method:

- Any query that runs longer than the OAuth token expiration policy may fail.
- Cross-tenant Azure Active Directory (Azure AD) accounts aren't supported.

If you selected **Windows** authentication method, make sure that account has access on the machine. If you're not sure, make sure to add *NT-AUTHORITY\Authenticated Users (S-1-5-11)* to the local machine **Users** group.

6. Optionally, under **Single sign-on**, you can configure [single sign-on \(SSO\)](#) for your data source.

**Single sign-on**

Use SSO via Kerberos for DirectQuery queries ⓘ

Use SSO via Kerberos for DirectQuery and Import queries ⓘ

Use SSO via Azure AD for DirectQuery queries ⓘ

Depending on your organization settings, for DirectQuery-based reports, you can configure **Use SSO via Kerberos for DirectQuery queries**, **Use SSO via Kerberos for DirectQuery And Import queries** or **Use SSO via Azure AD for DirectQuery queries**. You can configure **Use SSO via Kerberos for DirectQuery And Import queries** for refresh-based reports.

If you use **Use SSO via Kerberos for DirectQuery queries** and use this data source for a DirectQuery-based report, the report uses the credentials of the user that signs in to the Power BI service. A refresh-based report uses the credentials that you enter in the **Username** and **Password** fields and the **Authentication** method you choose.

When you use **Use SSO via Kerberos for DirectQuery And Import queries**, you don't need to provide any credentials. If this data source is used for DirectQuery-based reports, the report uses the user that's mapped to the Azure AD user that signs in to the Power BI service. A refresh-based report uses the dataset owner's security context.

For more information about **Use SSO via Kerberos for DirectQuery queries** and **Use SSO via Kerberos for DirectQuery And Import queries**, see [Overview of single sign-on \(SSO\) for gateways in Power BI](#).

If you use **Use SSO via Azure AD for DirectQuery queries** and use this data source for a DirectQuery-based report, the report uses the Azure AD token of the user who signs into the Power BI service. A refresh-based report uses the credentials that you enter in the **Username** and **Password** fields and the **Authentication** method you choose. The **Use SSO via Azure AD for DirectQuery queries** option is available only if the tenant admin allows Azure AD SSO via the on-premises data gateway, and for the following data sources:

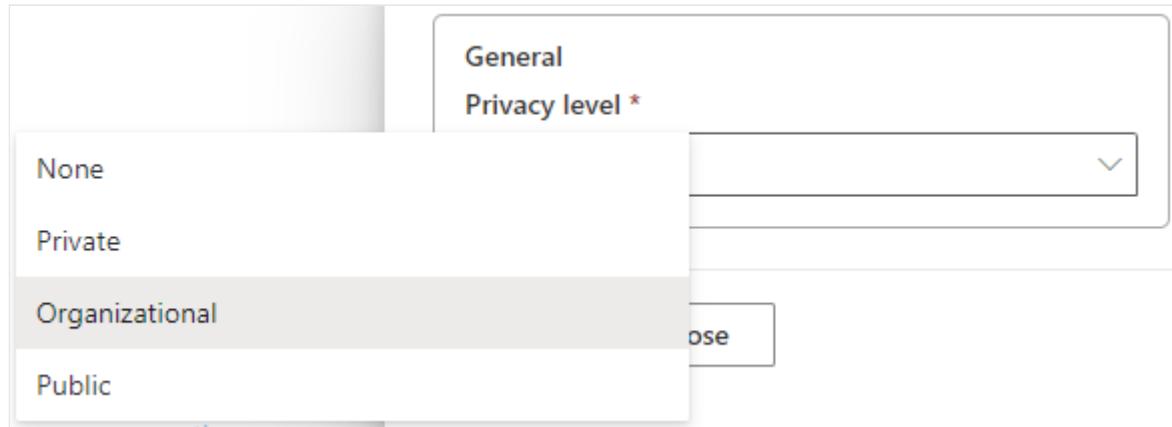
- SQL Server
- Azure Data Explorer
- Snowflake

For more information about **Use SSO via Azure AD for DirectQuery queries**, see [Azure AD single sign-on \(SSO\) for Gateway](#).

 **Note**

SSO for Import queries is available only for the SSO data sources that use **Kerberos constrained delegation**.

7. Under **General > Privacy level**, optionally configure a [privacy level](#) for your data source. This setting doesn't apply to **DirectQuery**.



8. Select **Create**. Under **Settings**, you see **Created new connection** if the process succeeds.



You can now use this data source to include data from SQL Server in your Power BI dashboards and reports.

## Remove a data source

You can remove a data source if you no longer use it. If you remove a data source, any dashboards and reports that rely on that data source no longer work.

To remove a data source, select the data source from the **Data (preview)** screen in **Manage connections and gateways**, and then select **Remove** from the top ribbon.

A screenshot of the 'Data (preview)' screen. At the top, there are several buttons: 'Settings', 'Manage users', a redboxed 'Remove' button, and 'Get help'. Below the buttons, the title 'Data (preview)' is displayed. Underneath, there are three tabs: 'Data sources' (which is underlined), 'On-premises data gateways', and 'Virtual network data gateways'. A descriptive text below the tabs reads: 'Power BI data sources for DirectQuery and Import datasets and dataflows, via cloud and the on-premises data gateway.' A table follows, listing data sources. The first row shows 'Name ↑', 'Data source type', and 'Users'. The second row lists 'bies-dmts-gw02\_AdventureWorksD...', 'SQL Server', and 'Megan Bowen'. The third row, which is highlighted with a gray background, lists 'c:\users\mideboer.europe\docume...', 'File', and 'Megan Bowen'. A checkmark icon is next to the first column of this row.

## Use the data source for scheduled refresh or DirectQuery

After you create the data source, it's available to use with DirectQuery connections or through scheduled refresh. You can learn more about setting up scheduled refresh in [Configure scheduled refresh](#).

The link between your dataset and the data source in the gateway is based on your server name and database name. These names must match. For example, if you supply

an IP address for the server name in Power BI Desktop, you must use the IP address for the data source in the gateway configuration. If you use SERVER\INSTANCE in Power BI Desktop, you must use the same format in the data source you configure for the gateway.

If you're listed in the **Users** tab of the data source configured in the gateway, and the server and database name match, you see the gateway listed as **Running** under **Gateway connections** in the **Settings** for your data source. You can select **Scheduled refresh** to set up scheduled refresh for the data source.

► **Gateway connection**

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#).

Use a data gateway

On

Gateway	Department	Contact information	Status	Actions
<input type="radio"/> Personal Gateway			Running on ContosoSQL01	
<input checked="" type="radio"/> Contoso Gateway		john@contoso.com	Running on ContosoSQL01	

**Apply** **Discard**

▷ Data source credentials  
▷ Sensitivity label  
▷ Parameters  
▷ Scheduled refresh

**Important**

If your dataset contains multiple data sources, each data source must be added in the gateway. If one or more data sources aren't added to the gateway, you won't see the gateway as available for scheduled refresh.

## Manage users

After you add a data source to a gateway, you give users and security groups access to the specific data source, not the entire gateway. The access list for the data source controls only who is allowed to publish reports that include data from the data source. Report owners can create dashboards and apps, and then share those items with other users.

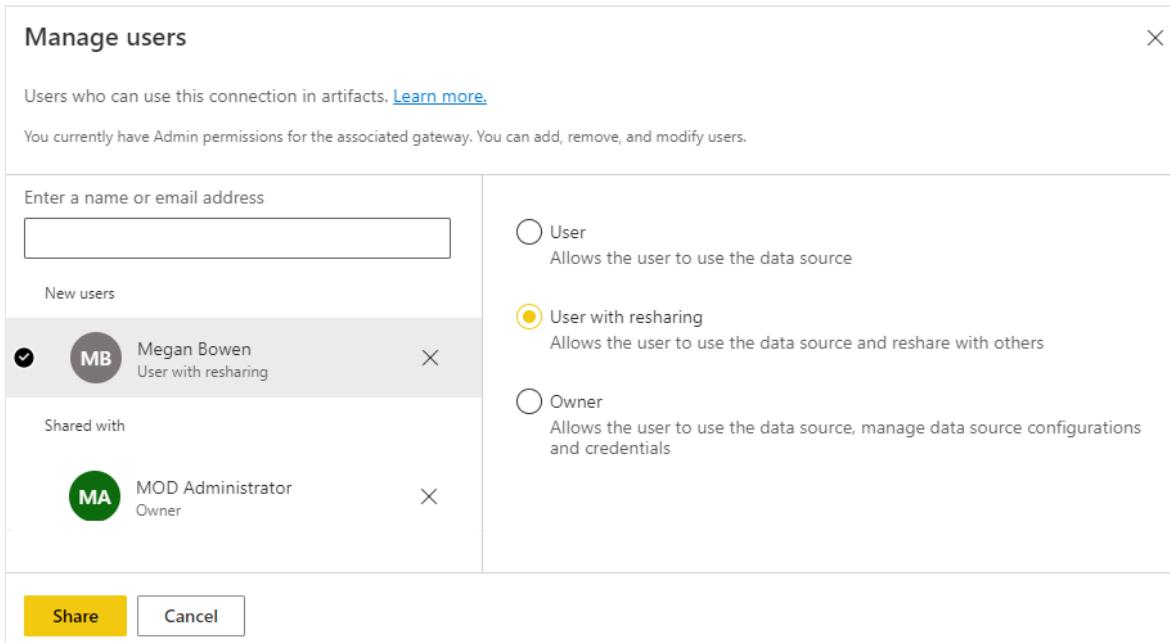
You can also give users and security groups administrative access to the gateway.

### Note

Users with access to the data source can associate datasets to the data source, and connect, based on either the stored credentials or SSO you selected while creating a data source.

## Add users to a data source

1. From the page header in the Power BI service, select the **Settings** icon, and then select **Manage connections and gateways**.
2. Select the data source where you want to add users.
3. Select **Manage users** from the top ribbon
4. On the **Manage users** screen, enter the users and/or security groups from your organization who can access the selected data source.
5. Select the new user name, and select the role to assign: **User**, **User with resharing**, or **Owner**.
6. Select **Share**, and the added member's name is added to the list of people who can publish reports that use this data source.



The screenshot shows the 'Manage users' dialog box. At the top, it says 'Manage users' and has a close button. Below that, it says 'Users who can use this connection in artifacts. [Learn more.](#)' and 'You currently have Admin permissions for the associated gateway. You can add, remove, and modify users.' A search bar says 'Enter a name or email address'. On the left, there's a 'New users' section with a checked checkbox next to 'MB' (Megan Bowen) and the role 'User with resharing'. Below that is a 'Shared with' section with 'MOD Administrator' (Owner). On the right, there are three radio button options: 'User' (Allows the user to use the data source), 'User with resharing' (selected, Allows the user to use the data source and reshare with others), and 'Owner' (Allows the user to use the data source, manage data source configurations and credentials). At the bottom are 'Share' and 'Cancel' buttons.

Remember that you need to add users to each data source that you want to grant access to. Each data source has a separate list of users. Add users to each data source separately.

## Remove users from a data source

On the **Manage Users** tab for the data source, you can remove users and security groups that use this data source.

## Store encrypted credentials in the cloud

When you add a data source to the gateway, you must provide credentials for that data source. All queries to the data source run by using these credentials. The credentials are encrypted securely with symmetric encryption, so that they can't be decrypted in the cloud. The credentials are sent to the machine that runs the on-premises gateway, where they're decrypted when the data sources are accessed.

## List of available data source types

For information about which data sources the on-premises data gateway supports, see [Power BI data sources](#).

## Next steps

- [Manage your data source - Analysis Services](#)
- [Manage your data source - SAP HANA](#)
- [Manage your data source - SQL Server](#)
- [Manage your data source - Oracle](#)
- [Manage your data source - Import/scheduled refresh](#)
- [Guidance for deploying a data gateway](#)

More questions? Try the [Power BI Community](#).

# Manage SQL Server Analysis Services data sources

Article • 03/10/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

After you [install an on-premises data gateway](#), you can [add data sources](#) to use with the gateway. This article describes how to add a SQL Server Analysis Services (SSAS) data source to your on-premises gateway to use for scheduled refresh or for live connections.

To learn more about how to set up a live connection to SSAS, watch this [Power BI Walkthrough: Analysis Services Live Connect](#) video.

## ⓘ Note

If you have an Analysis Services data source, you need to install the gateway on a computer joined to the same forest or domain as your Analysis Services server.

## ⓘ Note

The gateway supports only Windows authentication for Analysis Services.

## Add a data source

To connect to either a multidimensional or tabular Analysis Services data source:

1. On the **New connection** screen for your on-premises data gateway, select **Analysis Services** for **Connection type**. For more information about how to add a data source, see [Add a data source](#).

## New connection

X



Gateway cluster name \*

powerbi-docs

Connection name \*

New SSAS

Connection type \*

Analysis Services

2. Fill in the information for the data source, which includes **Server** and **Database**.

The gateway uses the information you enter for **Username** and **Password** to connect to the Analysis Services instance.

Server \*

SSAServer

Database \*

AdventureWorks

Authentication ⓘ

Authentication method \*

Windows

Username \*

CONTOSO\contosouser

Password \*

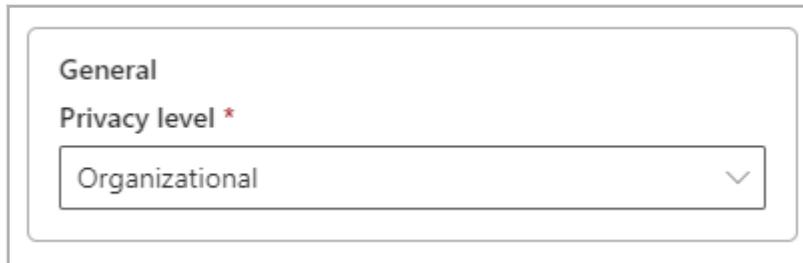
.....

### ⓘ Note

The Windows account you enter must be a member of the Server Administrator role on the Analysis Services instance you're connecting to. If

this account's password is set to expire, users get a connection error unless you update the data source password. For more information about how credentials are stored, see [Store encrypted credentials in the cloud](#).

3. Configure the **Privacy level** for your data source. This setting controls how data can be combined for scheduled refresh. The privacy-level setting doesn't apply to live connections. To learn more about privacy levels for your data source, see [Set privacy levels \(Power Query\)](#).



4. Optionally, you can configure user name mapping now. For instructions, see [Manual user name remapping](#).
5. After you complete all the fields, select **Create**.

You can now use this data source for scheduled refresh or live connections against an on-premises Analysis Services instance.

## User names for Analysis Services

To learn about authentication with Analysis Services live connections in Power BI, watch this video:

 **Note**

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



Each time a user interacts with a report connected to Analysis Services, the effective user name passes to the gateway and then passes on to your on-premises Analysis Services server. The email address that you use to sign in to Power BI passes to Analysis Services as the effective user in the [EffectiveUserName](#) connection property.

The email address must match a defined user principal name (UPN) within the local Active Directory (AD) domain. The UPN is a property of an AD account. The Windows account must be present in an Analysis Services role. If a match can't be found in AD, the sign-in isn't successful. To learn more about AD and user naming, see [User naming attributes](#).

## Map user names for Analysis Services data sources

You can also map your Power BI sign-in name to a local directory UPN. To learn about UPN mapping in Power BI, watch this video:

**Note**

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



Power BI allows mapping user names for Analysis Services data sources. You can configure rules to map a Power BI sign-in user name to an `EffectiveUserName` that passes to the Analysis Services connection. This feature is a great workaround when your Azure Active Directory (Azure AD) user name doesn't match a UPN in your local Active Directory instance. For example, if your email address is `meganb@contoso.onmicrosoft.com`, you can map it to `meganb@contoso.com`, and that value passes on to the gateway.

You can map user names for Analysis Services in two different ways:

- Manual user remapping in Power BI
- Active Directory lookup mapping, which uses on-premises AD property lookup to remap Azure AD UPNs to on-premises AD users.

Manual mapping by using on-premises AD property lookup is possible, but is time consuming and difficult to maintain, especially when pattern matching isn't enough. For example, domain names or user account names might be different between Azure AD and on-premises AD. Therefore, manual mapping with the second approach isn't recommended.

The following sections describe the two mapping approaches.

## Manual user remapping in Power BI

You can configure custom UPN rules in Power BI for Analysis Services data sources. Custom rules help if your Power BI service sign-in name doesn't match your local directory UPN. For example, if you sign in to Power BI with `meganb@contoso.com` but your

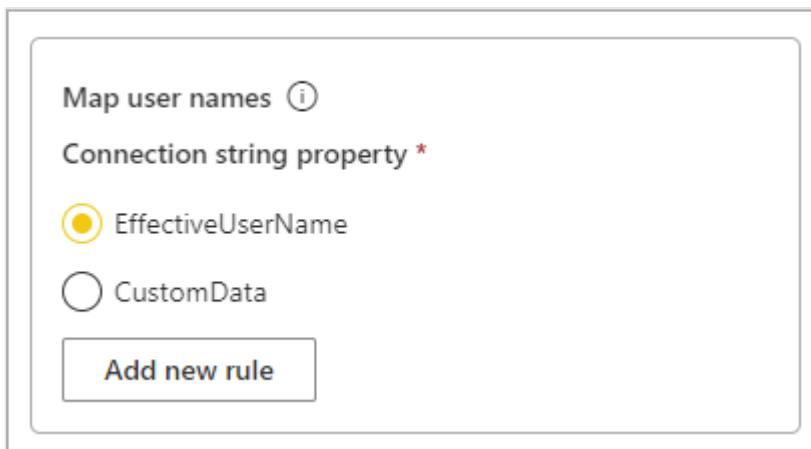
local directory UPN is `meganb@contoso.local`, you can configure a mapping rule to pass `meganb@contoso.local` to Analysis Services.

**ⓘ Important**

The mapping works for the specific data source that's being configured. It's not a global setting. If you have multiple Analysis Services data sources, you have to map the users for each data source.

To do manual UPN mapping, follow these steps:

1. Under the Power BI gear icon, select **Manage gateways and connections**.
2. Select the data source, and then select **Settings** from the top menu.
3. On the **Settings** screen, in the **Map user names** box, make sure **EffectiveUserName** is selected and then select **Add new rule**.



4. Under **Map user names**, for each user name to map, enter values for **Original name** and **New name**, and then select **Add new rule**. The **Replace** value is the sign-in address for Power BI, and the **With** value is the value to replace it with. The replacement passes to the `EffectiveUserName` property for the Analysis Services connection.

## Map user names ⓘ

Connection string property \*

EffectiveUserName

CustomData

Replace

With

Original name

New name



Required

Required

Add new rule

Enter a user name to see how the mapping rules will ch...

Original name

Original name

Test rule

For example:

Map user names ⓘ

Connection string property \*

EffectiveUserName

CustomData

Replace	With	
@contoso.com	@contoso...	
<input checked="" type="checkbox"/> meganb@locale.c...	meganb@cc	

**Add new rule** **Remove**

Enter a user name to see how the mapping rules will ch...

Original name

Original name

**Test rule**

### ⓘ Note

Be sure not to change users that you don't intend to change. For example, if you replace the **Original name** of `contoso.com` with a **New name** of `@contoso.local`, all user sign-ins that contain `@contoso.com` are replaced with `@contoso.local`. Also, if you replace an **Original name** of `meganb@contoso.com` with a **New name** of `meganb@contoso.local`, a sign-in of `v-meganb@contoso.com` is sent as `v-meganb@contoso.local`.

You can select an item in the list and reorder it by dragging and dropping, or delete an entry by selecting the garbage can icon.

## Use a wildcard

You can use a \* wildcard for your **Replace** (original name) string. You can only use the wildcard on its own and not with any other string part. Use a wildcard if you want to replace all users with a single value to pass to the data source. This approach is useful

when you want all users in an organization to use the same user in your local environment.

## Test the mapping rule

To validate the name replacement, enter a value for **Original name**, and select **Test rule**.

Enter a user name to see how the mapping rules will ch...

Original name

**Test rule**

After rule applied

### ① Note

The saved rules work immediately in the browser. It takes a few minutes before the Power BI service starts to use the saved rules.

## Active Directory lookup mapping

This section describes how to do an on-premises Active Directory property lookup to remap Azure AD UPNs to AD users. First, review how this remapping works.

Each query by a Power BI Azure AD user to an on-premises SSAS server passes along a UPN string such as `firstName.lastName@contoso.com`.

Lookup mapping in an on-premises data gateway with configurable custom user mapping follows these steps:

1. Find the Active Directory to search. You can use automatic or configurable.
2. Look up the attribute of the Active Directory user, such as **Email**, from the Power BI service. The attribute is based on an incoming UPN string like `firstName.lastName@contoso.com`.
3. If the Active Directory lookup fails, it attempts to pass along the UPN to SSAS as the `EffectiveUserName`.
4. If the Active Directory lookup succeeds, it retrieves the `UserPrincipalName` of that Active Directory user.

5. The mapping passes the `UserPrincipalName` email, such as `Alias@corp.on-prem.contoso`, to SSAS as the `EffectiveUserName`.

 **Note**

Any manual UPN user mappings defined in the Power BI data source gateway configuration are applied before sending the UPN string to the on-premises data gateway.

For the Active Directory lookup to work properly at runtime, you must change the on-premises data gateway service to run with a domain account instead of a local service account.

1. Make sure to [download and install the latest gateway](#).
2. In the [On-premises data gateway](#) app on your machine, go to **Service settings > Change service account**. Make sure you have the recovery key for the gateway, because you need to restore it on the same machine unless you want to create a new gateway. You must restart the gateway service for the change to take effect.
3. Go to the gateway's installation folder, `C:\Program Files\On-premises data gateway`, as an administrator to ensure that you have write permissions. Open the `Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config` file.
4. Edit the `ADUserNameLookupProperty` and the `ADUserNameReplacementProperty` values according to the AD attribute configurations for your AD users. The values in the following image are examples. These configurations are case sensitive, so make sure they match the values in AD.

```
<setting name="ADUserNameLookupProperty" serializeAs="String">
 <value>Email</value>
</setting>
<setting name="ADUserNameReplacementProperty" serializeAs="String">
 <value>UserPrincipalName</value>
</setting>
```

If the file provides no value for the `ADServerPath` configuration, the gateway uses the default global catalog. You can specify multiple values for the `ADServerPath`. The values must be separated by semicolons, as in the following example:

XML

```
<setting name="ADServerPath" serializeAs="String">
 <value> GC://serverpath1; GC://serverpath2;GC://serverpath3</value>
```

```
</setting>
```

The gateway parses the values for `ADServerPath` from left to right until it finds a match. If the gateway doesn't find a match, it uses the original UPN. Make sure the account that runs the gateway service, `PBIEgwService`, has query permissions to all AD servers that you specify in `ADServerPath`.

The gateway supports two types of `ADServerPath`:

- For WinNT: `<value="WinNT://usa.domain.corp.contoso.com,computer"/>`
- For global catalog (GC): `<value> GC://USA.domain.com </value>`

5. Restart the on-premises data gateway service for the configuration change to take effect.

## Authentication to a live Analysis Services data source

Each time a user interacts with Analysis Services, the effective user name is passed to the gateway and then to the on-premises Analysis Services server. The UPN, which is typically the email address you use to sign in to the cloud, is passed to Analysis Services as the effective user in the `EffectiveUserName` connection property.

When the dataset is in Import Mode, the gateway will send the `EffectiveUserName` of the UPN of the dataset owner. This means that the UPN of the dataset owner will be passed to Analysis Services as the effective user in the `EffectiveUserName` connection property.

This email address should match a defined UPN within the local Active Directory domain. The UPN is a property of an AD account. A Windows account must be present in an Analysis Services role to have access to the server. If no match is found in Active Directory, the sign-in won't be successful.

## Role-based and row-level security

Analysis Services can also provide filtering based on the Active Directory account. The filtering can use role-based security or row-level security. A user's ability to query and view model data depends on the roles that their Windows user account belongs to, and on dynamic row-level security if it's configured.

- **Role-based security.** Models provide security based on user roles. You can define roles for a particular model project during authoring in SQL Server Data Tools Business Intelligence tools. After a model is deployed, you can define roles by using SQL Server Management Studio. Roles contain members assigned by Windows user name or by Windows group.

Roles define the permissions users have to query or take actions on the model. Most users belong to a role with read permissions. Other roles give administrators permissions to process items, manage database functions, and manage other roles.

- **Row-level security.** Models can provide dynamic row-level security. Any defined row-level security is specific to Analysis Services. For role-based security, every user must have at least one role, but no tabular model requires dynamic row-level security.

At a high level, dynamic security defines a user's read access to data in particular rows in particular tables. Similar to roles, dynamic row-level security relies on a user's Windows user name.

Implementing role and dynamic row-level security in models is beyond the scope of this article. For more information, see [Roles in tabular models](#) and [Security roles \(Analysis Services - Multidimensional data\)](#). For the most in-depth understanding of tabular model security, download the [Securing the tabular BI semantic model](#) whitepaper.

## Azure AD authentication

Microsoft cloud services use [Azure AD](#) to authenticate users. Azure AD is the tenant that contains user names and security groups. Typically, the email address a user signs in with is the same as the UPN of the account.

## Roles in the local Active Directory instance

For Analysis Services to determine if a user belongs to a role with permissions to read data, the server needs to convert the effective user name passed from Azure AD to the gateway and on to the Analysis Services server. The Analysis Services server passes the effective user name to a Windows Active Directory domain controller (DC). The Active Directory DC then validates that the effective user name is a valid UPN on a local account. The DC returns the user's Windows user name back to the Analysis Services server.

You can't use `EffectiveUserName` on a non-domain joined Analysis Services server. The Analysis Services server must be joined to a domain to avoid sign-in errors.

# Identify your UPN

You might not know what your UPN is, and you might not be a domain administrator. You can use the following command from your workstation to find out the UPN for your account:

```
Windows Command Prompt
```

```
whoami /upn
```

The result looks similar to an email address, but is the UPN that's on your domain account. If you use an Analysis Services data source for live connections, and this UPN doesn't match the email address you use to sign in to Power BI, you might need to [map your user name](#).

## Synchronize an on-premises AD with Azure AD

If you plan to use Analysis Services live connections, your local AD accounts must match Azure AD. The UPN must match between the accounts.

Cloud services only use accounts within Azure AD. If you add an account in your local AD instance that doesn't exist in Azure AD, you can't use the account. There are several ways you can match your local AD accounts with Azure AD:

- Add accounts manually to Azure AD.  
Create an account on the Azure portal, or within the Microsoft 365 admin center, with an account name that matches the UPN of the local AD account.
- Use [Azure AD Connect sync](#) to synchronize local accounts to your Azure AD tenant.

Azure AD Connect ensures that the UPN matches between Azure AD and your local AD instance. The Azure AD Connect tool provides options for directory synchronization and setting up authentication. Options include password hash sync, pass-through authentication, and federation. If you're not an admin or a local domain administrator, contact your IT admin to help with configuration.

### Note

Synchronizing accounts with Azure AD Connect sync creates new accounts within your Azure AD tenant.

# Use the data source

After you add the SSAS data source, it's available to use with either live connections or through scheduled refresh.

## ⓘ Note

The server and database name must match between Power BI Desktop and the data source within the on-premises data gateway.

The link between your dataset and the data source within the gateway is based on your server name and database name. These names must match. For example, if you supply an IP address for the server name within Power BI Desktop, you must use the IP address for the data source within the gateway configuration. If you use *SERVER\INSTANCE* in Power BI Desktop, you also must use *SERVER\INSTANCE* within the data source configured for the gateway. This requirement holds for both live connections and scheduled refresh.

## Use the data source with live connections

You can use a live connection against tabular or multidimensional instances. You select a live connection in Power BI Desktop when you first connect to the data. Make sure that the server and database name matches between Power BI Desktop and the configured data source for the gateway. Also, to be able to publish live connection datasets, your users must appear under **Users** in the data source list.

After you publish reports, either from Power BI Desktop or by getting data in the Power BI service, your data connection should start to work. It might take several minutes after you create the data source in the gateway before you can use the connection.

## Use the data source with scheduled refresh

If you're listed in the **Users** tab of the data source configured within the gateway, and the server and database name match, you see the gateway as an option to use with scheduled refresh.

#### ⚠ Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Connections and Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click **Apply**.

Use an On-premises or VNet data gateway



On

Gateway	Department	Contact information	Status	Actions
<input type="radio"/>  Personal Gateway			 Running on LAPTOP-R5	
<input checked="" type="radio"/>  powerbi-docs		admin@contoso.com	 Running on LAPTOP-R5	 

**Apply**

**Discard**

## Limitations of Analysis Services live connections

- Cell level formatting and translation features aren't supported.
- Actions and named sets aren't exposed to Power BI. You can still connect to multidimensional cubes that contain actions or named sets to create visuals and reports.

## SKU requirements

Server version	Required SKU
2012 SP1 CU4 or later	Business Intelligence and Enterprise SKU
2014	Business Intelligence and Enterprise SKU
2016	Standard SKU or higher

## Next steps

- [Troubleshoot the on-premises data gateway](#)
- [Troubleshoot gateways - Power BI](#)

More questions? Try the [Power BI Community](#).

# Manage your data source - SAP HANA

Article • 02/22/2023

## Note

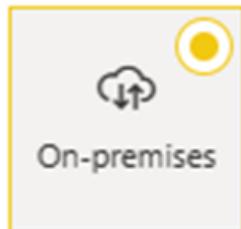
We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

After you [install the on-premises data gateway](#), you need to [add data sources](#) that can be used with the gateway. This article looks at how to work with gateways and SAP HANA data sources that are used either for scheduled refresh or for DirectQuery.

## Add a data source

For more information about how to add a data source, see [Add a data source](#). Under **Connection type**, select **SAP HANA**.

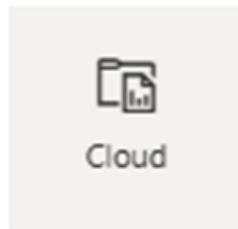
# New connection



On-premises



Virtual  
network



Cloud

**Gateway cluster name \***

GatewayApril



**Connection name \***

New data source

**Connection type \***

SAP HANA



After you select the SAP HANA data source type, fill in the **Server**, **Username**, and **Password** information for the data source.

**Note**

All queries to the data source run using these credentials. To learn more about how credentials are stored, see [Store encrypted credentials in the cloud](#).

## Server \*

ContosoSAP01

### Authentication ⓘ

#### Authentication method \*

Basic



#### Username \*

contosouser

#### Password \*

.....



Skip test connection

After you fill in everything, select **Create**. You can now use this data source for scheduled refresh or DirectQuery against an SAP HANA server that is on-premises. You see *Created New data source* if it succeeded.

## Settings



Created New data source.



## Advanced settings

Optionally, you can configure the privacy level for your data source. This setting controls how data can be combined. It's only used for scheduled refresh. The privacy-level setting doesn't apply to DirectQuery. To learn more about privacy levels for your data source, see [Set privacy levels \(Power Query\)](#).

## General

### Privacy level \*

Organizational



## Use the data source

After you create the data source, it's available to use with either DirectQuery connections or through scheduled refresh.

### ⓘ Note

The server and database names must match between Power BI Desktop and the data source within the on-premises data gateway.

The link between your dataset and the data source within the gateway is based on your server name and database name. These names must match. For example, if you supply an IP address for the server name within Power BI Desktop, you must use the IP address for the data source within the gateway configuration. If you use *SERVER\INSTANCE* in Power BI Desktop, you also must use it within the data source configured for the gateway.

This requirement is the case for both DirectQuery and scheduled refresh.

## Use the data source with DirectQuery connections

Make sure that the server and database names match between Power BI Desktop and the configured data source for the gateway. You also need to make sure your user is listed in the **Users** tab of the data source to publish DirectQuery datasets. The selection for DirectQuery occurs within Power BI Desktop when you first import data. For more information about how to use DirectQuery, see [Use DirectQuery in Power BI Desktop](#).

After you publish, either from Power BI Desktop or **Get Data**, your reports should start to work. It might take several minutes after you create the data source within the gateway for the connection to be usable.

# Use the data source with scheduled refresh

If you're listed in the **Users** tab of the data source configured within the gateway and the server name and database name match, you see the gateway as an option to use with scheduled refresh.

▲ **Gateway connection**

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#).

Use a data gateway

On

Gateway	Department	Contact information	Status	Actions
<input type="radio"/>	Personal Gateway		Running on ContosoSAP01	
<input checked="" type="radio"/>	Contoso Gateway	john@contoso.com	Running on ContosoSAP01	

**Apply** **Discard**

## Next steps

- Troubleshoot the on-premises data gateway
- Troubleshoot gateways - Power BI

More questions? Try asking the [Power BI Community](#).

# Manage a SQL Server data source

Article • 03/21/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

After you [install an on-premises data gateway](#), you can add data sources to use with the gateway. This article describes how to add a SQL Server data source to an on-premises data gateway to use for scheduled refresh or DirectQuery.

## Add a data source

Follow these instructions to add a SQL Server data source to your on-premises data gateway.

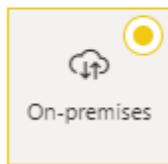
## ⓘ Note

When you use DirectQuery, the gateway supports only **SQL Server 2012 SP1** and later.

1. On the **New connection** screen, select **On-premises**. Enter the **Gateway cluster name** and new **Connection name**, and under **Connection type**, select **SQL Server**.

## New connection

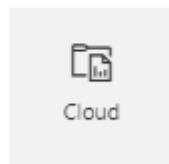
X



On-premises



Virtual  
network



Cloud

Gateway cluster name \*

powerbi-docs



Connection name \*

NewSQL

Connection type \*

SQL Server



2. Fill in the **Server** and **Database** information for the data source.

3. Under **Authentication Method**, choose either **Windows** or **Basic**. Choose **Basic** if you plan to use SQL authentication instead of Windows authentication. Then enter the credentials to use for this data source.

Server \*

Database \*

#### Authentication ⓘ

##### Authentication method \*

##### Username \*

##### Password \*

 Skip test connection

#### Single sign-on

- Use SSO via Kerberos for DirectQuery queries ⓘ
- Use SSO via Kerberos for DirectQuery and Import queries ⓘ
- Use SSO via Azure AD for DirectQuery queries ⓘ

All queries to the data source run using these credentials unless you configure and enable Kerberos single sign-on (SSO) for the data source. With SSO, datasets use the current Power BI user's SSO credentials to execute the queries.

For more information about storing and using credentials, see:

- [Store encrypted credentials in the cloud](#)
- [Use Kerberos for single sign-on \(SSO\) from Power BI to on-premises data sources.](#)

4. Configure the **Privacy level** for your data source. This setting controls how data can be combined for scheduled refresh only. The privacy level setting doesn't apply to DirectQuery. To learn more about privacy levels for your data source, see [Privacy levels \(Power Query\)](#).

5. Select **Create**.

Password \*

  
 Skip test connection

Single sign-on

Use SSO via Kerberos for DirectQuery queries ⓘ

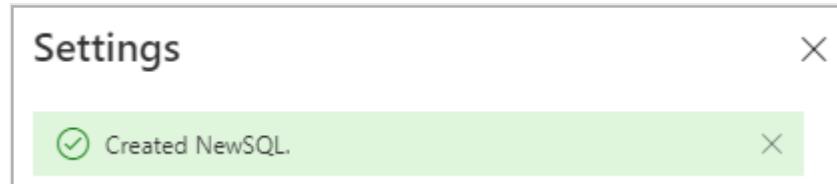
Use SSO via Kerberos for DirectQuery and Import queries ⓘ

Use SSO via Azure AD for DirectQuery queries ⓘ

General

Privacy level \*

You see a success message if the creation succeeds. You can now use this data source for scheduled refresh or DirectQuery against an on-premises SQL Server.



For more information about how to add a data source, see [Add a data source](#).

## Use the data source

After you create the data source, it's available to use with either DirectQuery connections or through scheduled refresh.

### Server and database names must match

The link between your dataset and the data source in the gateway is based on your server name and database name. These names must match exactly.

For example, if you supply an IP address for the server name in Power BI Desktop, you must use the IP address for the data source in the gateway configuration. If you use *SERVER\INSTANCE* in Power BI Desktop, you must use *SERVER\INSTANCE* in the data source you configure for the gateway. This requirement holds for both DirectQuery and scheduled refresh.

## Use the data source with DirectQuery connections

Make sure that the server and database names match between Power BI Desktop and the configured data source for the gateway. Also, to be able to publish DirectQuery datasets, your users must appear under **Users** in the data source list.

You select the DirectQuery connection method in Power BI Desktop when you first connect to data. For more information about how to use DirectQuery, see [Use DirectQuery in Power BI Desktop](#).

After you publish reports, either from Power BI Desktop or by getting data in Power BI service, your SQL Server on-premises data connection should work. It might take several minutes after you create the data source in the gateway to be able to use the connection.

## Use the data source with scheduled refresh

If you're listed in the **Users** column of the data source configured within the gateway, and the server name and database name match, you see the gateway as an option to use with scheduled refresh.

⚠ Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Connections and Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click apply.

Use an On-premises or VNet data gateway

On

Gateway	Department	Contact information	Status	Actions
<input type="radio"/>  Personal Gateway			 Running on LAPTOP-R5	
<input checked="" type="radio"/>  powerbi-docs		admin@contoso.com	 Running on LAPTOP-R5	 

**Apply** **Discard**

## Next steps

- [Connect to on-premises data in SQL Server](#)

- Troubleshoot the on-premises data gateway
- Troubleshoot gateways - Power BI
- Use Kerberos for single sign-on (SSO) from Power BI to on-premises data sources

More questions? Try asking the [Power BI Community](#).

# Manage your data source - Oracle

Article • 03/07/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

After you [install the on-premises data gateway](#), you can [add data sources](#) to use with the gateway. This article looks at how to work with the on-premises gateway and Oracle data sources either for scheduled refresh or for DirectQuery.

## Connect to an Oracle database

To connect to an Oracle database with the on-premises data gateway, [download and install the 64-bit Oracle Client](#) on the computer running the gateway. The Oracle client software you use depends on the Oracle server version, which always matches the 64-bit gateway.

Supported Oracle versions are:

- Oracle Server 9 and later.
- Oracle Data Access Components (ODAC) software 11.2 and later.

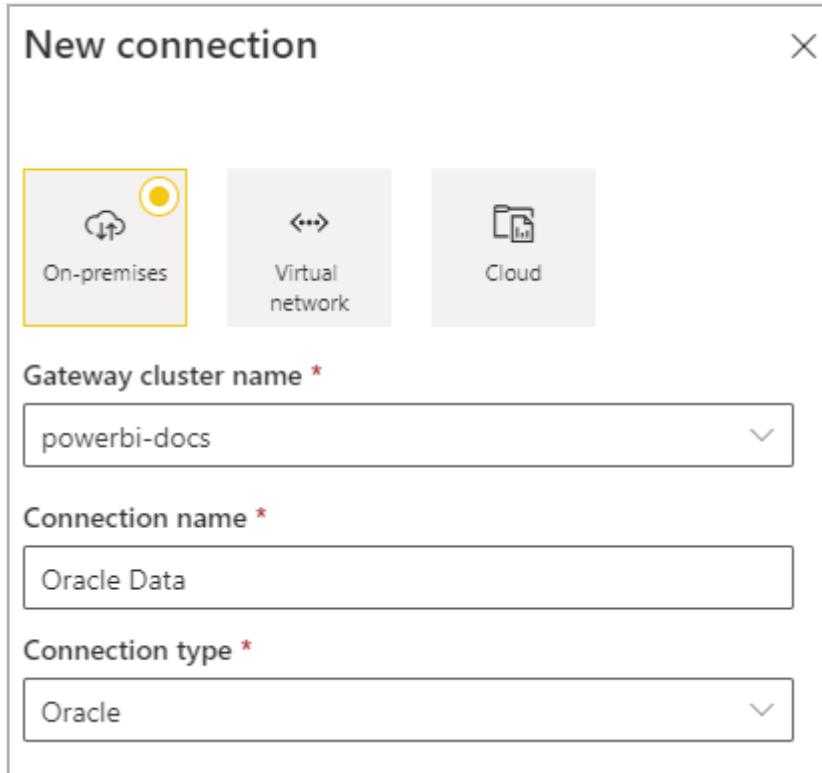
When you install the client:

- Choose an ODAC version that's compatible with your Oracle server. For instance, ODAC 12.x doesn't always support Oracle Server version 9.
- Choose the Windows installer.
- During setup, make sure to select the checkbox next to **Configure ODP.NET and/or Oracle Providers for ASP.NET at machine-wide level** in the setup wizard so that Power BI can connect to your Oracle database. Some versions of the setup wizard select this checkbox by default and others don't.

After you install the client and configure ODAC properly, use Power BI Desktop or another test client to verify correct installation and configuration on the gateway.

## Add a data source

1. On the **New connection** screen for your on-premises data gateway, select **Oracle** for **Connection type**.



2. In **Server**, enter the name for the data source.
3. Under **Authentication method**, choose either **Windows** or **Basic**. Choose **Basic** if you plan to use an account you created within Oracle instead of Windows authentication. Then enter the credentials to use for this data source.

**(!) Note**

All queries to the data source run with these credentials. To learn more about credential storage, see [Store encrypted credentials in the cloud](#).

4. Configure the **Privacy level** for your data source. This setting controls how data can combine for scheduled refresh. The privacy-level setting doesn't apply to DirectQuery. To learn more about privacy levels for your data source, see [Privacy levels \(Power Query\)](#).
5. Select **Create**.

**Server \***

**Authentication ⓘ**

**Authentication method \***

**Username \***

**Password \***

Skip test connection

**Single sign-on**

Use SSO via Kerberos for DirectQuery queries ⓘ

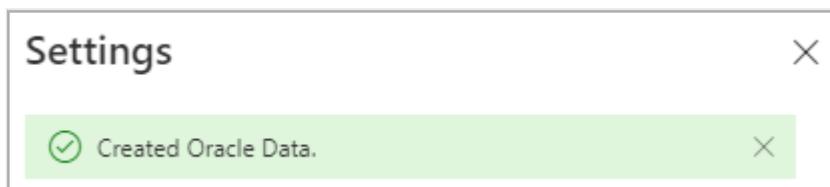
Use SSO via Kerberos for DirectQuery and Import queries ⓘ

**General**

**Privacy level \***

**Create** **Close**

If the creation succeeds, you see **Created <Data source name>**. You can now use this data source for scheduled refresh or DirectQuery with the on-premises Oracle server.



## Use the data source

After you create the data source, it's available to use with either DirectQuery or scheduled refresh.

### ⓘ Important

The server and database names must match between Power BI Desktop and the data source within the on-premises data gateway.

The link between your dataset and the data source within the gateway is based on your server name and database name. These names must match exactly. For example, if you supply an IP address for the server name within Power BI Desktop, you must use the IP address for the data source within the gateway configuration. This name also has to match an alias that the *tnsnames.ora* file defines. This requirement is the case for both DirectQuery and scheduled refresh.

## Use the data source with DirectQuery connections

Make sure that the server and database names match between Power BI Desktop and the configured data source for the gateway. Also, to be able to publish DirectQuery datasets, your users must appear under **Users** in the data source listing.

After you publish reports, either from Power BI Desktop or by getting data in Power BI service, your on-premises data connection should work. It might take several minutes after you create the data source in the gateway to be able to use the connection.

## Use the data source with scheduled refresh

If you're in the **Users** list of a data source you configure within the gateway, and the server and database names match, you see the gateway as an option to use with scheduled refresh.

ⓘ Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Connections and Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click **Apply**.

Use an On-premises or VNet data gateway

On

Gateway	Department	Contact information	Status	Actions
<input type="radio"/>  Personal Gateway			 Running on Contoso	
<input checked="" type="radio"/>  powerbi-docs		admin@contoso.com	 Running on Contoso	 

**Apply** **Discard**

# Troubleshooting

You might get one of the following Oracle errors when the naming syntax is either incorrect or improperly configured:

- ORA-12154: TNS:could not resolve the connect identifier specified.
- ORA-12514: TNS:listener does not currently know of service requested in connect descriptor.
- ORA-12541: TNS:no listener.
- ORA-12170: TNS:connect timeout occurred.
- ORA-12504: TNS:listener was not given the SERVICE\_NAME in CONNECT\_DATA.

These errors might occur if the Oracle client isn't installed or isn't configured properly. If the client is installed, verify that the *tnsnames.ora* file is properly configured and you're using the proper *net\_service\_name*. Also make sure that the *net\_service\_name* is the same between the machine that uses Power BI Desktop and the machine that runs the gateway.

You might also encounter a compatibility issue between the Oracle server version and the ODAC version. Typically, these versions should match, because some combinations are incompatible. For instance, ODAC 12.x doesn't support Oracle Server version 9.

To diagnose connectivity issues between the data source server and the gateway machine, install a client like Power BI Desktop or Oracle ODBC Test on the gateway machine. You can use the client to check connectivity to the data source server.

For more gateway troubleshooting information, see [Troubleshoot the on-premises data gateway](#).

## Next steps

- [Troubleshoot gateways - Power BI](#)
- [Power BI Premium](#)

More questions? Try asking the [Power BI Community](#).

# Manage your data source - import and scheduled refresh

Article • 03/07/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

After you [install the on-premises data gateway](#), you need to [add data sources](#) that can be used with the gateway. This article looks at how to work with gateways and data sources that are used for scheduled refresh as opposed to DirectQuery or live connections.

## Add a data source

Select a data source type. All of the data source types listed can be used for scheduled refresh with the on-premises data gateway. Analysis Services, SQL Server, and SAP HANA can be used for scheduled refresh, DirectQuery, or live connections. For more information about how to add a data source, see [Add a data source](#).



## New data source

Gateway cluster name \*

GatewayApril



Data source name \*

New data source

Data source type \*

SQL Server



Then fill in the information for the data source, which includes the source information and credentials that are used to access the data source.

Note

All queries to the data source run by using these credentials. To learn more about how credentials are stored, see [Store encrypted credentials in the cloud](#).

**Server \***

ContosoSQL01

**Database \***

AdventureWorks

**Authentication** ⓘ

**Authentication method \***

Basic



**Username \***

sqluser

**Password \***

.....



Skip test connection

For a list of data source types that can be used with scheduled refresh, see [List of available data source types](#).

After you fill in everything, select **Create**. If the action succeeds, you see *Created New data source*. You can now use this data source for scheduled refresh with your on-premises data.

## Settings

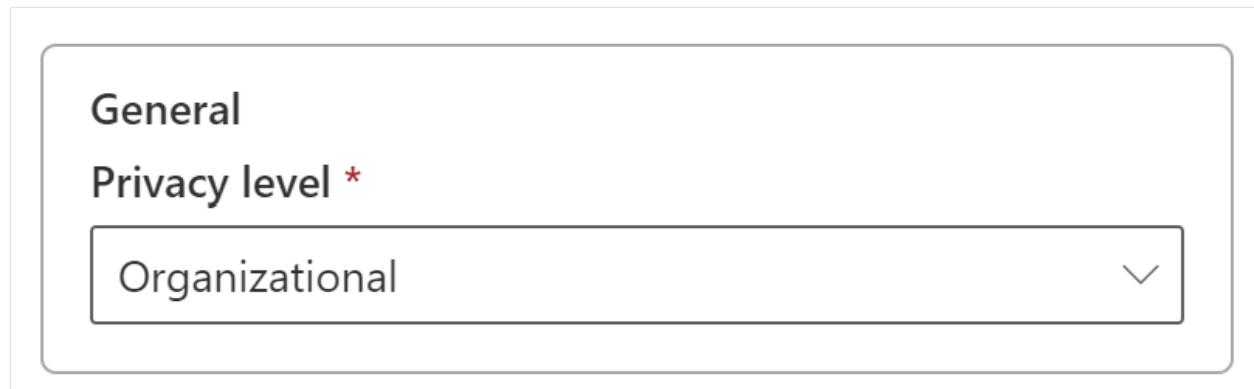


Created New data source.



## Advanced settings

Optionally, you can configure the privacy level for your data source. This setting controls how data can be combined. It's only used for scheduled refresh. To learn more about privacy levels for your data source, see [Privacy levels \(Power Query\)](#).



## Use the data source for scheduled refresh

After you create the data source, it's available to use with either DirectQuery connections or through scheduled refresh.

### ! Note

The server and database names must match between Power BI Desktop and the data source within the on-premises data gateway.

The link between your dataset and the data source within the gateway is based on your server name and database name. These names must match. For example, if you supply an IP address for the server name within Power BI Desktop, you must use the IP address for the data source within the gateway configuration. If you use *SERVER\INSTANCE* in Power BI Desktop, you also must use it within the data source configured for the gateway.

If you're listed in the **Users** tab of the data source configured within the gateway and the server name and database name match, you see the gateway as an option to use with scheduled refresh.

#### ► Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#).

Use a data gateway



Gateway	Department	Contact information	Status	Actions
<input type="radio"/> Personal Gateway			Running on ContosoORA01	
<input checked="" type="radio"/> Contoso Gateway		john@contoso.com	Running on ContosoORA01	

**Apply** **Discard**

#### ⓘ Important

Upon republish, the data set owner must associate the dataset to a gateway and corresponding data source again. The previous association is not maintained after republishing.

#### ⚠ Warning

If your dataset contains multiple data sources, each data source must be added within the gateway. If one or more data sources aren't added to the gateway, you don't see the gateway as available for scheduled refresh.

## Next steps

- [Troubleshooting the on-premises data gateway](#)
- [Troubleshoot gateways - Power BI](#)

More questions? Try the [Power BI Community](#).

# Merge or append on-premises and cloud data sources

Article • 02/22/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

You can use the on-premises data gateway to merge or append on-premises and cloud data sources in the same query. This solution is helpful when you want to combine data from multiple sources without having to use separate queries.

## ⓘ Note

This article applies only to datasets that have cloud and on-premises data sources merged or appended in a single query. For datasets that include separate queries, for instance, one that connects to an on-premises data source and the other to a cloud data source, the gateway doesn't execute the query for the cloud data source.

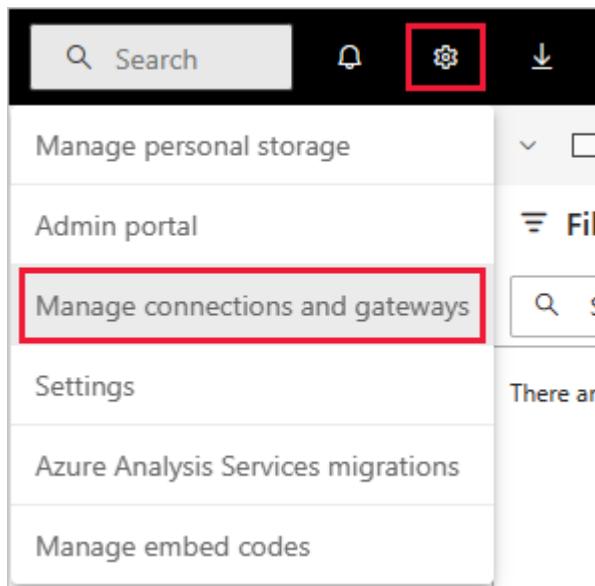
## Prerequisites

- A [gateway installed](#) on a local computer.
- A Power BI Desktop file with queries that combine on-premises and cloud data sources.

## ⓘ Note

To access any cloud data sources, you must ensure that the gateway has access to those data sources.

1. In the upper-right corner of the Power BI service, select the gear icon  then **Manage connections and gateways**.



2. Select the gateway you want to configure, and select **Settings** from the top ribbon

A screenshot of the 'On-premises data gateways' settings page. The top ribbon shows options like 'Details', 'Settings' (highlighted with a red box), 'Manage users', 'Remove', and 'Get help'. Below the ribbon, there is a section titled 'Data (preview)' with tabs for 'Data sources', 'On-premises data gateways' (selected and underlined), and 'Virtual network data gateways'. A note below the tabs says: 'The on-premises data gateway acts as a bridge, providing quick and secure data transfer between on-premises data and Power BI, Microsoft Flow, Logic Apps, and Power Automate.' A table lists existing gateways: 'diagnosticstesting' and 'GatewayApril'. The 'GatewayApril' row has a checked checkbox and three ellipsis buttons.

3. Under **Settings**, select **Allow user's cloud data sources to refresh through this gateway cluster**, then select **Save**.

4. To add any **on-premises data sources** used in your queries, select **Connections**, then select **New** to create a connection. You don't need to add the cloud data sources here.

5. Select your gateway for **Gateway cluster name**. Name the connection and specify the type of connection and other required information. Then select **Create**.

6. Upload to the Power BI service your Power BI Desktop file with the queries that combine on-premises and cloud data sources.

With the cloud credentials set, you can now refresh the dataset by using the **Refresh now** option. Or, you can schedule it to refresh periodically.

## Next steps

To learn more about data refresh for gateways, see [Use the data source for scheduled refresh](#).

# Overview of single sign-on for on-premises data gateways in Power BI

Article • 08/25/2023

You can get seamless single sign-on (SSO) connectivity, enabling Power BI reports and dashboards to update in real time by configuring your on-premises data gateway. You have the option of configuring your gateway with the following SSO options:

- Active Directory (AD) SSO, which includes:
  - [Kerberos](#) constrained delegation.
  - [Security Assertion Markup Language \(SAML\)](#).
- Azure Active Directory (Azure AD) SSO.

 **Note**

SSO is only supported by Power BI datasets and not by Power BI dataflows.

## Supported data sources for SSO

AD SSO is usually configured for on-premises data sources that are secured within your on-premises network. Azure AD SSO is configured for data sources that support Azure Active Directory authentication, typically cloud data sources, secured behind an Azure Virtual Network.

While the on-premises data gateway supports SSO by using [DirectQuery](#) or [Refresh](#) for the AD-based SSO options, only [DirectQuery](#) is supported for Azure AD SSO.

Power BI supports the following data sources:

- Amazon Redshift (Azure AD)
- Azure Databricks
- Azure Data Explorer (Azure AD)
- Azure SQL (Azure AD)
- Azure Synapse Analytics (Azure AD)
- Denodo (Kerberos)
- Hive LLAP (Kerberos)
- Impala (Kerberos)
- Oracle (Kerberos)
- SAP BW Application Server (Kerberos)

- SAP BW Message Server (Kerberos)
- SAP HANA (Kerberos and SAML)
- Snowflake (Azure AD)
- Spark (Kerberos)
- SQL Server (Kerberos)
- Teradata (Kerberos)
- Tibco Data Virtualization (Kerberos)

ⓘ Note

SQL Server Analysis Services also supports SSO, but does so using [Live connections](#), rather than using Kerberos or SAML. Power BI doesn't support SSO for M-extensions.

## Interact with reports that rely on SSO

When a user interacts with a DirectQuery report in the Power BI service, each cross-filter, slice, sort, and report editing operation can result in queries that execute live against the underlying data source. When you configure SSO for the data source, queries execute under the identity of the user that interacts with Power BI. That is, they run through the web experience or Power BI mobile apps. Therefore, each user sees precisely the data for which they have permissions in the underlying data source.

You can also configure a report that is set up for refresh in the Power BI service to use SSO. When you configure SSO for this data source, queries execute under the identity of the dataset owner within Power BI. Therefore, the refresh happens based on the dataset owner's permissions on the underlying data source. Refresh using SSO is currently enabled only for data sources using [Kerberos](#) constrained delegation.

## Next steps

Now that you understand the basics of SSO through the gateway, read detailed information about setting up SSO here:

- [Active Directory \(AD\) SSO](#)
- [Azure Active Directory \(Azure AD\) SSO](#)

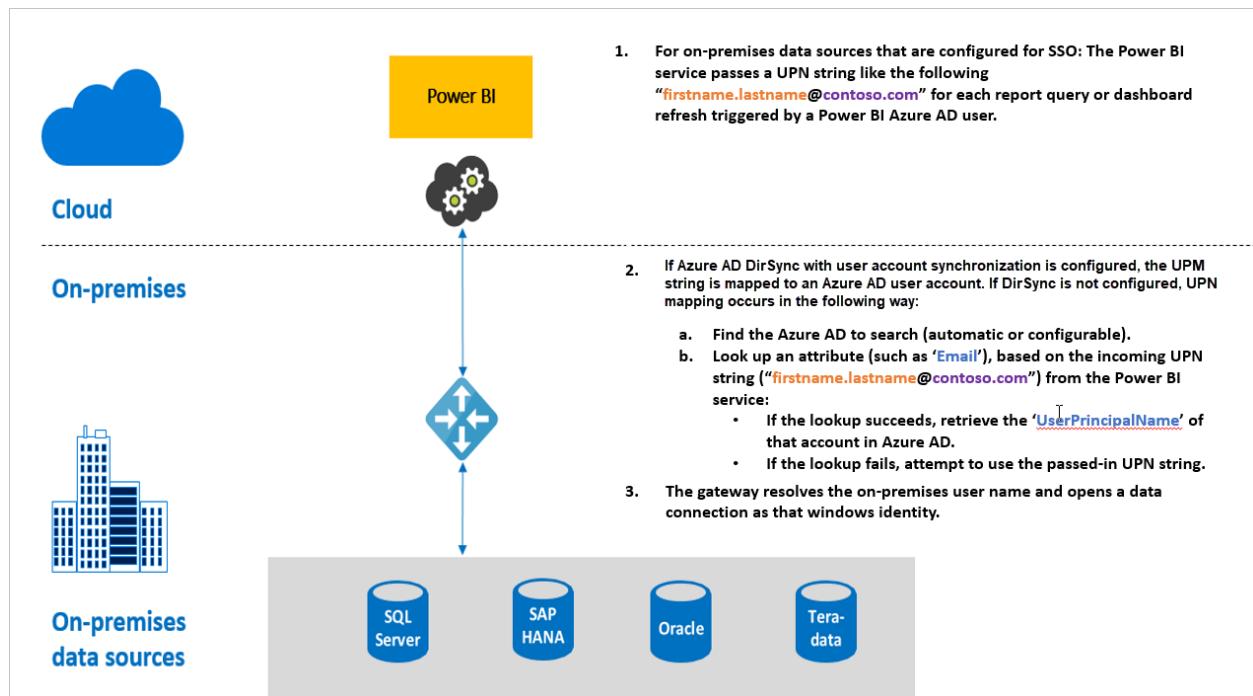
# Active Directory (AD) SSO

Article • 11/28/2022

The on-premises data gateway supports Active Directory (AD) SSO for connecting to your on-premises data sources that have Active Directory configured. AD SSO includes both [Kerberos](#) constrained delegation and [Security Assertion Markup Language \(SAML\)](#). For more information on SSO and the list of data sources supported for AD SSO, see [Overview of single sign-on \(SSO\) for on-premises data gateways in Power BI](#).

## Query steps when running Active Directory SSO

A query that runs with SSO consists of three steps, as shown in the following diagram.



Here are additional details about each step:

1. For each query, the Power BI service includes the *user principal name (UPN)*, which is the fully qualified username of the user currently signed in to the Power BI service, when it sends a query request to the configured gateway.
2. The gateway must map the Azure Active Directory UPN to a local Active Directory identity:
  - a. If Azure AD DirSync (also known as *Azure AD Connect*) is configured, then the mapping works automatically in the gateway.

- b. Otherwise, the gateway can look up and map the Azure AD UPN to a local AD user by performing a lookup against the local Active Directory domain.
- 3. The gateway service process impersonates the mapped local user, opens the connection to the underlying database, and then sends the query. You don't need to install the gateway on the same machine as the database.

## Next steps

Now that you understand the basics of enabling SSO through the gateway, read more detailed information about Kerberos and SAML:

- [Single sign-on \(SSO\) - Kerberos](#)
- [Single sign-on \(SSO\) - SAML](#)
- [Overview of single sign-on \(SSO\) for on-premises data gateways in Power BI](#)

# Configure Kerberos-based SSO from Power BI service to on-premises data sources

Article • 06/20/2023

Enabling SSO makes it easy for Power BI reports and dashboards to refresh data from on-premises sources while respecting user-level permissions configured on those sources. Use [Kerberos constrained delegation](#) to enable seamless SSO connectivity.

This article describes the steps you need to take to configure Kerberos-based SSO from Power BI service to on-premises data sources.

## Prerequisites

Several items must be configured for Kerberos constrained delegation to work properly, including \*Service Principal Names (SPN) and delegation settings on service accounts.

 Note

Using DNS aliasing with SSO is not supported.

## Configuration outline

The steps required for configuring gateway single sign-on are outlined below.

1. Complete all the steps in [Section 1: Basic configuration](#).
2. Depending on your Active Directory environment and the data sources used, you may need to complete some or all of the configuration described in [Section 2: Environment-specific configuration](#).

Possible scenarios that may require additional configuration are listed below:

Scenario	Go to
Your Active Directory environment is security hardened.	<a href="#">Add gateway service account to Windows Authorization and Access Group</a>

Scenario	Go to
The gateway service account and the user accounts that the gateway will impersonate are in separate domains or forests.	<a href="#">Add gateway service account to Windows Authorization and Access Group</a>
You don't have Azure AD Connect with user account synchronization configured and the UPN used in the Power BI for users does not match the UPN in your local Active Directory environment.	<a href="#">Set user-mapping configuration parameters on the gateway machine</a>
You plan to use an SAP HANA data source with SSO.	<a href="#">Complete data source-specific configuration steps</a>
You plan to use an SAP BW data source with SSO.	<a href="#">Complete data source-specific configuration steps</a>
You plan to use a Teradata data source with SSO.	<a href="#">Complete data source-specific configuration steps</a>

3. Validate your configuration as described in [Section 3: Validate configuration](#) to ensure that SSO is set up correctly.

## Section 1: Basic configuration

### Step 1: Install and configure the Microsoft on-premises data gateway

The on-premises data gateway supports an in-place upgrade, and *settings takeover* of existing gateways.

### Step 2: Obtain domain admin rights to configure SPNs (SetSPN) and Kerberos constrained delegation settings

To configure SPNs and Kerberos delegation settings, a domain administrator should avoid granting rights to someone that doesn't have domain admin rights. In the following section, we cover the recommended configuration steps in more detail.

### Step 3: Configure the Gateway service account

Option A below is the required configuration unless you have both Azure AD Connect configured and user accounts are synchronized. In that case, option B is recommended.

## Option A: Run the gateway Windows service as a domain account with SPN

In a standard installation, the gateway runs as the machine-local service account, **NT Service\PBIEgwService**.

Name	Description	Status	Startup Type	Log On As
On-premises data gateway service	The on-premises data ...	Running	Automatic	NT SERVICE\PBIEgwService

To enable Kerberos constrained delegation, the gateway must run as a domain account, unless your Azure Active Directory (Azure AD) instance is already synchronized with your local Active Directory instance (by using Azure AD DirSync/Connect). To switch to a domain account, see [change the gateway service account](#).

### Configure an SPN for the gateway service account

First, determine whether an SPN was already created for the domain account used as the gateway service account:

1. As a domain administrator, launch the **Active Directory Users and Computers** Microsoft Management Console (MMC) snap-in.
2. In the left pane, right-click the domain name, select **Find**, and then enter the account name of the gateway service account.
3. In the search result, right-click the gateway service account and select **Properties**.
4. If the **Delegation** tab is visible on the **Properties** dialog, then an SPN was already created and you can skip to [Configure Kerberos constrained delegation](#).
5. If there isn't a **Delegation** tab on the **Properties** dialog box, you can manually create an SPN on the account to enable it. Use the [setspn tool](#) that comes with Windows (you need domain admin rights to create the SPN).

For example, suppose the gateway service account is **Contoso\GatewaySvc** and the gateway service is running on the machine named **MyGatewayMachine**. To set the SPN for the gateway service account, run the following command:

```
setspn -S gateway/MyGatewayMachine Contoso\GatewaySvc
```

You can also set the SPN by using the Active Directory Users and Computers MMC snap-in.

## Option B: Configure computer for Azure AD Connect

If Azure AD Connect is configured and user accounts are synchronized, the gateway service doesn't need to perform local Azure AD lookups at runtime. Instead, you can simply use the local service SID for the gateway service to complete all required configuration in Azure AD. The Kerberos constrained delegation configuration steps outlined in this article are the same as the configuration steps required in the Azure AD context. They are applied to the gateway's computer object (as identified by the local service SID) in Azure AD instead of the domain account. The local service SID for NT SERVICE/PBIEgwService is as follows:

```
S-1-5-80-1835761534-3291552707-3889884660-1303793167-3990676079
```

To create the SPN for this SID against the Power BI Gateway computer, you would need to run the following command from an administrative command prompt (replace <COMPUTERNAME> with the name of the Power BI Gateway computer):

```
SetSPN -s HTTP/S-1-5-80-1835761534-3291552707-3889884660-1303793167-3990676079
<COMPUTERNAME>
```

## Step 4: Configure Kerberos constrained delegation

You can configure delegation settings for either standard Kerberos constrained delegation or resource-based Kerberos constrained delegation. For more information on the differences between the two approaches to delegation, see [Kerberos constrained delegation overview](#).

The following service accounts are required:

- Gateway service account: Service user representing the gateway in Active Directory, with an SPN configured in Step 3.
- Data Source service account: Service user representing the data source in Active Directory, with an SPN mapped to the data source.

### Note

The gateway and data source service accounts must be separate. The same service account cannot be used to represent both the gateway and data source.

Depending on which approach you want to use, proceed to one of the following sections. Don't complete both sections:

- [Option A: Standard Kerberos constrained delegation](#). This is the default recommendation for most environments.
- [Option B: Resource-based Kerberos constrained delegation](#). This is required if your data source belongs to a different domain than your gateway.

## Option A: Standard Kerberos constrained delegation

We'll now set the delegation settings for the gateway service account. There are multiple tools you can use to perform these steps. Here, we'll use the **Active Directory Users and Computers** MMC snap-in to administer and publish information in the directory. It's available on domain controllers by default; on other machines, you can enable it through Windows feature configuration.

We need to configure Kerberos constrained delegation with protocol transition. With constrained delegation, you must be explicit about which services you allow the gateway to present delegated credentials to. For example, only SQL Server or your SAP HANA server accepts delegation calls from the gateway service account.

This section assumes you have already configured SPNs for your underlying data sources (such as SQL Server, SAP HANA, SAP BW, Teradata, or Spark). To learn how to configure those data source server SPNs, refer to the technical documentation for the respective database server and see the section *What SPN does your app require?* in the [My Kerberos Checklist](#) blog post.

In the following steps, we assume an on-premises environment with two machines in the same domain: a gateway machine and a database server running SQL Server that has already been configured for Kerberos-based SSO. The steps can be adopted for one of the other supported data sources, so long as the data source has already been configured for Kerberos-based single sign-on. For this example, we'll use the following settings:

- Active Directory Domain (Netbios): **Contoso**
- Gateway machine name: **MyGatewayMachine**
- Gateway service account: **Contoso\GatewaySvc**
- SQL Server data source machine name: **TestSQLServer**
- SQL Server data source service account: **Contoso\SQLService**

Here's how to configure the delegation settings:

1. With domain administrator rights, open the **Active Directory Users and Computers** MMC snap-in.
2. Right-click the gateway service account (**Contoso\GatewaySvc**), and select **Properties**.
3. Select the **Delegation** tab.
4. Select **Trust this computer for delegation to specified services only > Use any authentication protocol**.
5. Under **Services to which this account can present delegated credentials**, select **Add**.
6. In the new dialog box, select **Users or Computers**.
7. Enter the service account for the data source, and then select **OK**.

For example, a SQL Server data source can have a service account like *Contoso\SQLService*. An appropriate SPN for the data source should have already been set on this account.

8. Select the SPN that you created for the database server.

In our example, the SPN begins with *MSSQLSvc*. If you added both the FQDN and the NetBIOS SPN for your database service, select both. You might see only one.

9. Select **OK**.

You should now see the SPN in the list of services to which the gateway service account can present delegated credentials.

## GatewaySvc Properties

?

X

Organization	Published Certificates	Member Of	Password Replication
Dial-in	Object	Security	Environment
Remote control	Remote Desktop Services Profile	COM+	Attribute Editor
General	Address	Account	Profile
		Telephones	Delegation

Delegation is a security-sensitive operation, which allows services to act on behalf of another user.

- Do not trust this user for delegation
- Trust this user for delegation to any service (Kerberos only)
- Trust this user for delegation to specified services only
  - Use Kerberos only
  - Use any authentication protocol

Services to which this account can present delegated credentials:

Service Type	User or Computer	Port	Service N...
MSSQLSvc	TestSQLServer		

Expanded

Add...

Remove

OK

Cancel

Apply

Help

10. To continue the setup process, proceed to [Grant the gateway service account local policy rights on the gateway machine](#).

## Option B: Resource-based Kerberos constrained delegation

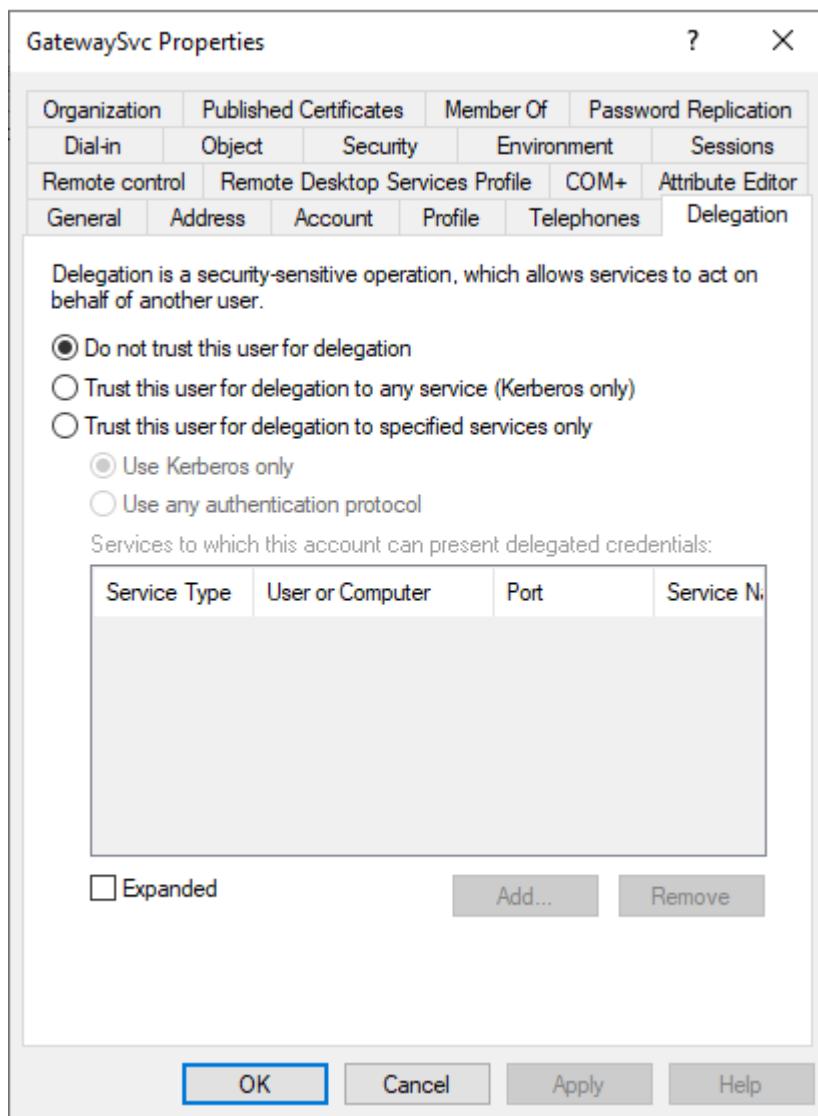
You use [resource-based Kerberos constrained delegation](#) to enable single sign-on connectivity for Windows Server 2012 and later versions. This type of delegation permits front-end and back-end services to be in different domains. For it to work, the back-end service domain needs to trust the front-end service domain.

In the following steps, we assume an on-premises environment with two machines in different domains: a gateway machine and a database server running SQL Server that has already been configured for Kerberos-based SSO. These steps can be adopted for one of the other supported data sources, so long as the data source has already been configured for Kerberos-based single sign-on. For this example, we'll use the following settings:

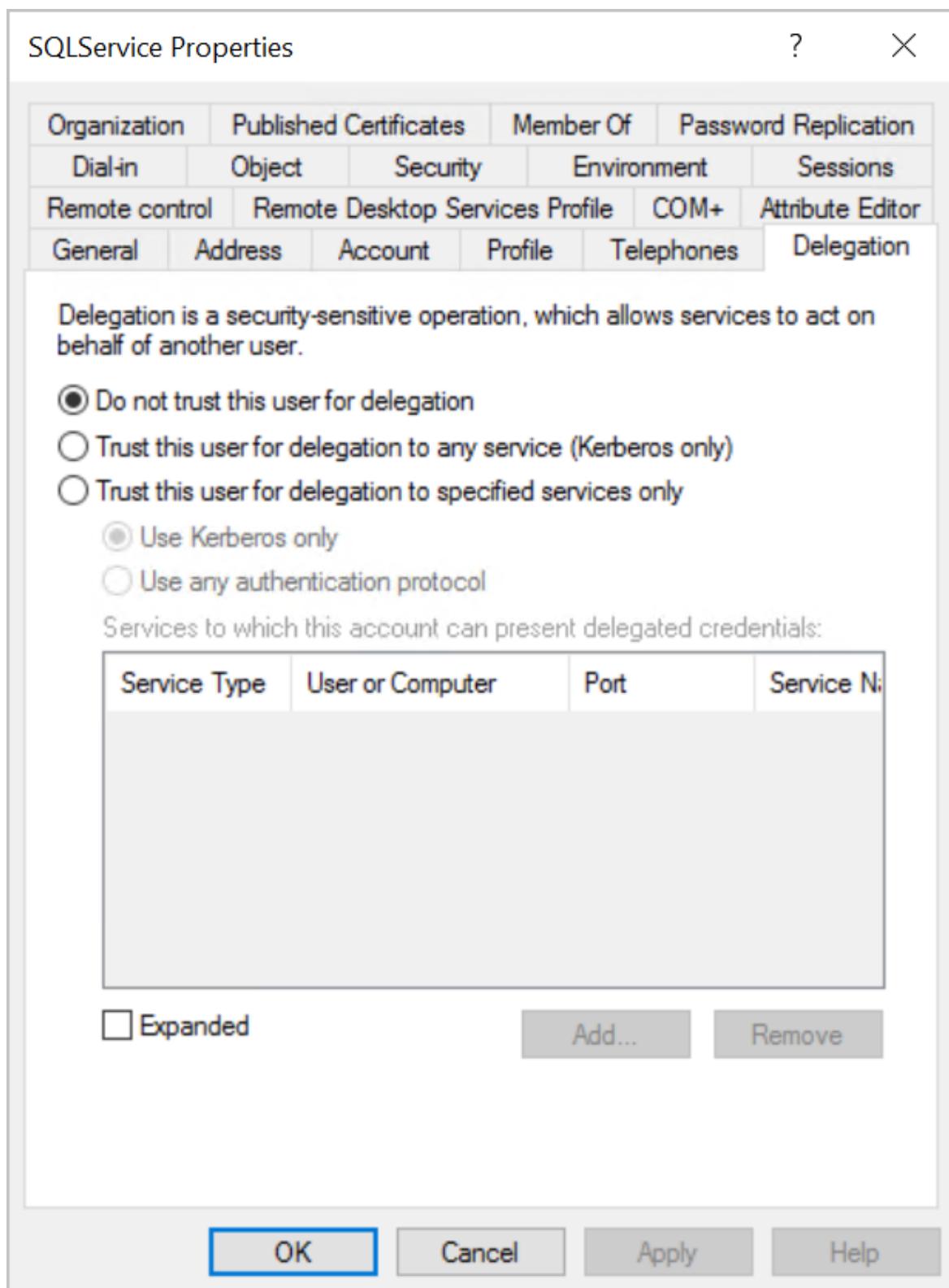
- Active Directory frontend Domain (Netbios): **ContosoFrontEnd**
- Active Directory backend Domain (Netbios): **ContosoBackEnd**
- Gateway machine name: **MyGatewayMachine**
- Gateway service account: **ContosoFrontEnd\GatewaySvc**
- SQL Server data source machine name: **TestSQLServer**
- SQL Server data source service account: **ContosoBackEnd\SQLService**

Complete the following configuration steps:

1. Use the **Active Directory Users and Computers** MMC snap-in on the domain controller for the **ContosoFrontEnd** domain and verify no delegation settings are applied for the gateway service account.



2. Use Active Directory Users and Computers on the domain controller for the **ContosoBackEnd** domain and verify no delegation settings are applied for the back-end service account.



3. In the **Attribute Editor** tab of the account properties, verify that the **msDS-AllowedToActOnBehalfOfOtherIdentity** attribute isn't set.

## SQLService Properties

?

X

Organization	Published Certificates	Member Of	Password Replication		
Dial-in	Object		Environment	Sessions	
General	Address	Account	Profile	Telephones	Delegation
Remote control	Remote Desktop Services Profile	COM+	Attribute Editor		

## Attributes:

Attribute	Value
mail	<not set>
manager	<not set>
maxStorage	<not set>
mhsORAddress	<not set>
middleName	<not set>
mobile	<not set>
msCOM-UserPartitionSetLink	<not set>
msDRM-IdentityCertificate	<not set>
msDS-AllowedToActOnBehalfOfOtherIdentity	<not set>
msDS-AllowedToDelegateTo	<not set>
msDS-AssignedAuthNPolicy	<not set>
msDS-AssignedAuthNPolicySilo	<not set>
msDS-AuthenticatedAtDC	<not set>
msDS-Cached-Membership	<not set>

View

Filter

OK

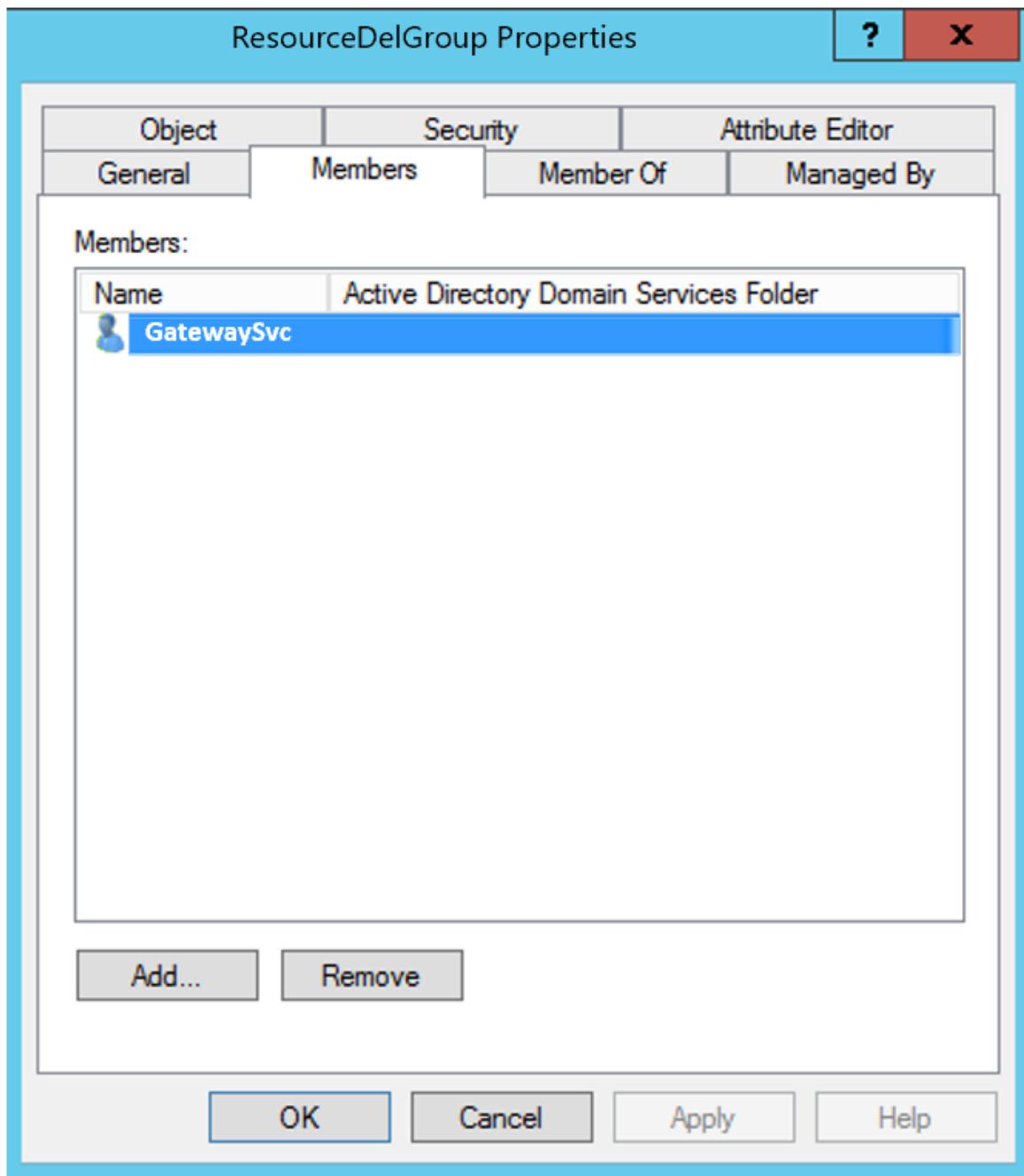
Cancel

Apply

Help

4. In Active Directory Users and Computers, create a group on the domain controller for the ContosoBackEnd domain. Add the **GatewaySvc** gateway service account to the **ResourceDelGroup** group.

To add users from a trusted domain, this group must have a scope of Domain local.



5. Open a command prompt and run the following commands in the domain controller for the **ContosoBackEnd** domain to update the **msDS-AllowedToActOnBehalfOfOtherIdentity** attribute of the back-end service account:

PowerShell

```
$c = Get-ADGroup ResourceDelGroup
Set-ADUser SQLService -PrincipalsAllowedToDelegateToAccount $c
```

6. In **Active Directory Users and Computers**, verify that the update is reflected in the **Attribute Editor** tab in the properties for the back-end service account.

## Step 5: Enable AES Encryption on Service Accounts

Apply the following settings to the gateway service account and **every** data source service account that the gateway can delegate to:

 **Note**

If there are existing enctype values defined on the service accounts(s), consult with your Active Directory Administrator, because following the below steps will overwrite the existing enctype values and may break clients.

1. With domain administrator rights, open the **Active Directory Users and Computers MMC** snap-in.
2. Right-click the gateway/data source service account and select **Properties**.
3. Select the **Account** tab.
4. Under **Account Options**, enable at least one (or both) of the following options.

Note that the same options need to be enabled for all service accounts.

- This account supports Kerberos AES 128-bit encryption
- This account supports Kerberos AES 256-bit encryption

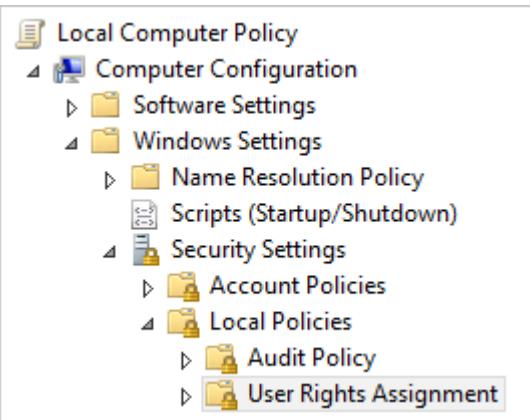
 **Note**

If you are unsure which encryption scheme to use, consult with your Active Directory Administrator.

## Step 6: Grant the gateway service account local policy rights on the gateway machine

Finally, on the machine running the gateway service (**MyGatewayMachine** in our example), grant the gateway service account the local policies **Impersonate a client after authentication** and **Act as part of the operating system** (**SeTcbPrivilege**). Perform this configuration with the Local Group Policy Editor (**gpedit.msc**).

1. On the gateway machine, run **gpedit.msc**.
2. Go to **Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.



3. Under **User Rights Assignment**, from the list of policies, select **Impersonate a client after authentication**.

Policy	Security Setting
Impersonate a client after authentication	LOCAL SERVICE.NET

4. Right-click the policy, open **Properties**, and then view the list of accounts.

The list must include the gateway service account (**Contoso\GatewaySvc** or **ContosoFrontEnd\GatewaySvc** depending on the type of constrained delegation).

5. Under **User Rights Assignment**, select **Act as part of the operating system (SeTcbPrivilege)** from the list of policies. Ensure that the gateway service account is included in the list of accounts.
6. Restart the On-premises data gateway service process.

## Step 7: Windows account can access gateway machine

SSO uses Windows Authentication, so make sure the Windows account can access the gateway machine. If not sure, add **NT-AUTHORITY\Authenticated Users (S-1-5-11)** to the local machine "Users" group.

## Section 2: Environment-specific configuration

### Add gateway service account to Windows Authorization and Access Group

Complete this section if **any** of the following situations apply:

- Your Active Directory environment is security hardened.

- When the gateway service account and the user accounts that the gateway will impersonate are in separate domains or forests.

You can also add the gateway service account to Windows Authorization and Access Group in situations where the domain / forest has not been hardened, but it isn't required.

For more information, see [Windows Authorization and Access Group](#).

To complete this configuration step, for each domain that contains Active Directory users you want the gateway service account to be able to impersonate:

1. Sign in to a computer in the domain, and launch the Active Directory Users and Computers MMC snap-in.
2. Locate the group **Windows Authorization and Access Group**, which is typically found in the **Builtin** container.
3. Double click on the group, and click on the **Members** tab.
4. Click **Add**, and change the domain location to the domain that the gateway service account resides in.
5. Type in the gateway service account name and click **Check Names** to verify that the gateway service account is accessible.
6. Click **OK**.
7. Click **Apply**.
8. Restart the gateway service.

## Set user-mapping configuration parameters on the gateway machine

Complete this section if:

- You don't have Azure AD Connect with user account synchronization configured AND
- The UPN used in Power BI for users does not match the UPN in your local Active Directory environment.

Each Active Directory user mapped in this way needs to have SSO permissions for your data source.

1. Open the main gateway configuration file,

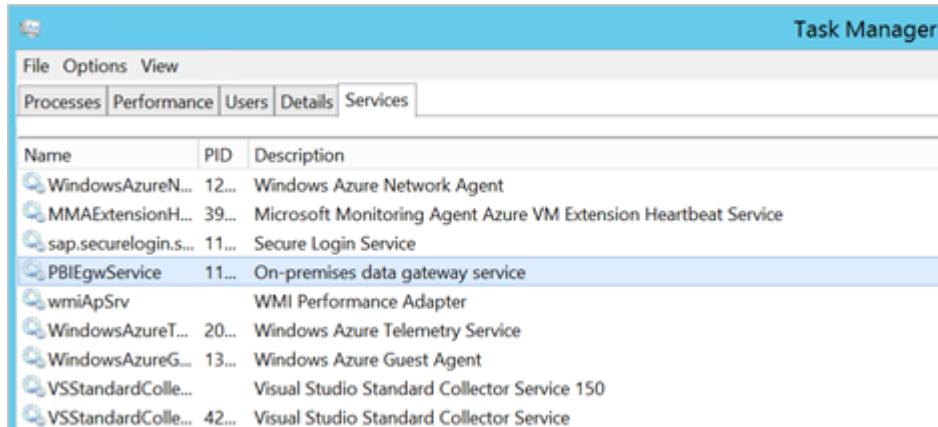
`Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll`. By default, this file is stored at `C:\Program Files\On-premises data gateway`.

2. Set **ADUserNameLookupProperty** to an unused Active Directory attribute. We'll use `msDS-cloudExtensionAttribute1` in the steps that follow. This attribute is available only in Windows Server 2012 and later.
3. Set **ADUserNameReplacementProperty** to `SAMAccountName` and then save the configuration file.

 **Note**

In multi-domain scenarios, you may need to set the **ADUserNameReplacementProperty** to `userPrincipalName` to preserve the domain information of the user.

4. From the **Services** tab of Task Manager, right-click the gateway service and select **Restart**.



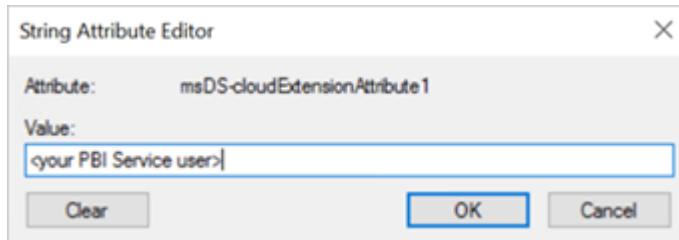
5. For each Power BI service user you want to enable Kerberos SSO for, set the `msDS-cloudExtensionAttribute1` property of a local Active Directory user (with SSO permission to your data source) to the full username (UPN) of the Power BI service user. For example, if you sign in to Power BI service as test@contoso.com and you want to map this user to a local Active Directory user with SSO permissions, say, test@LOCALDOMAIN.COM, set this user's `msDS-cloudExtensionAttribute1` attribute to test@contoso.com.

You can set the `msDS-cloudExtensionAttribute1` property with the Active Directory Users and Computers MMC snap-in:

- a. As a domain administrator, launch **Active Directory Users and Computers**.
- b. Right-click the domain name, select **Find**, and then enter the account name of the local Active Directory user to map.
- c. Select the **Attribute Editor** tab.

Locate the `msDS-cloudExtensionAttribute1` property, and double-click it. Set the value to the full username (UPN) of the user you use to sign in to the Power BI service.

d. Select OK.



e. Select **Apply**. Verify that the correct value has been set in the **Value** column.

## Complete data source-specific configuration steps

For SAP HANA, SAP BW, and Teradata data sources, additional configuration is required to use with gateway SSO:

- Use Kerberos for single sign-on (SSO) to SAP HANA.
- Use Kerberos single sign-on for SSO to SAP BW using CommonCryptoLib (`sapcrypto.dll`).
- Use Kerberos for single sign-on (SSO) to Teradata.

### ⓘ Note

Although other SNC libraries might also work for BW SSO, they aren't officially supported by Microsoft.

## Section 3: Validate configuration

### Step 1: Configure data sources in Power BI

After you complete all the configuration steps, use the **Manage Gateway** page in Power BI to configure the data source to use for SSO. If you have multiple gateways, ensure that you select the gateway you've configured for Kerberos SSO. Then, under **Settings** for the data source, ensure **Use SSO via Kerberos for DirectQuery queries** or **Use SSO via Kerberos for DirectQuery And Import queries** is checked for DirectQuery based Reports and **Use SSO via Kerberos for DirectQuery And Import queries** is checked for Import based Reports.

## Single sign-on

- Use SSO via Kerberos for DirectQuery queries (i)
- Use SSO via Kerberos for DirectQuery and Import queries (i)
- Use SSO via SAML for DirectQuery queries

The settings **Use SSO via Kerberos for DirectQuery queries** and **Use SSO via Kerberos for DirectQuery And Import queries** give a different behavior for DirectQuery based reports and Import based reports.

### Use SSO via Kerberos for DirectQuery queries:

- For DirectQuery based report, SSO credentials of the user are used.
- For Import based report, SSO credentials are not used, but the credentials entered in data source page are used.

### Use SSO via Kerberos for DirectQuery And Import queries:

- For DirectQuery based report, SSO credentials of the user are used.
- For Import based report, the SSO credentials of the dataset owner are used, regardless of the user triggering the Import.

## Step 2: Test single sign-on

Go to [Test single sign-on \(SSO\) configuration](#) to quickly validate that your configuration is set correctly and troubleshoot common problems.

## Step 3: Run a Power BI report

When you publish, select the gateway you've configured for SSO if you have multiple gateways.

## Next steps

For more information about the on-premises data gateway and DirectQuery, see the following resources:

- What is an on-premises data gateway?
- DirectQuery in Power BI
- Data sources supported by DirectQuery
- DirectQuery and SAP BW
- DirectQuery and SAP HANA

# Use Kerberos for SSO to SAP HANA

Article • 03/17/2023

## ⓘ Important

Because [SAP no longer supports OpenSSL](#), Microsoft has also discontinued its support. Your existing connections continue to work but you can no longer create new connections. Use SAP Cryptographic Library (CommonCryptoLib), or `sapcrypto`, instead.

This article describes how to configure your SAP HANA data source to enable single sign-on (SSO) from the Power BI service.

## ⓘ Note

Before you attempt to refresh a SAP HANA-based report that uses Kerberos SSO, complete the steps in both this article and [Configure Kerberos SSO](#).

## Enable SSO for SAP HANA

To enable SSO for SAP HANA, do the following steps:

1. Ensure the SAP HANA server is running the required minimum version, which depends on your SAP HANA server platform level:
  - [HANA 2 SPS 01 Rev 012.03](#)
  - [HANA 2 SPS 02 Rev 22](#)
  - [HANA 1 SP 12 Rev 122.13](#)
2. On the gateway computer, install the latest SAP HANA ODBC driver. The minimum version is HANA ODBC version 2.00.020.00 from August 2017.
3. Ensure that the SAP HANA server has been configured for Kerberos-based SSO.  
For more information about setting up SSO for SAP HANA by using Kerberos, see [Single sign-on using Kerberos](#). Also see the links from that page, particularly SAP Note 1837331 – HOWTO HANA DBSSO Kerberos/Active Directory.

We also recommend following these extra steps, which can yield a small performance improvement:

1. In the gateway installation directory, look for and open this configuration file:  
*Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config*.
2. Look for the `FullDomainResolutionEnabled` property, and change its value to `True`.

XML

```
<setting name=" FullDomainResolutionEnabled " serializeAs="String">
 <value>True</value>
</setting>
```

3. Run a Power BI report.

## Troubleshoot

This section provides instructions for troubleshooting using Kerberos for single sign-on (SSO) to SAP HANA in the Power BI service. By using these troubleshooting steps, you can self-diagnose and correct many issues you might be facing.

To follow the steps in this section, you need to [collect gateway logs](#).

### TLS/SSL error (certificate)

This issue has multiple symptoms.

- When you try to add a new data source, you might see an error like the following message:

Output

```
Unable to connect: We encountered an error while trying to connect to.
Details: "We could not register this data source for any gateway
instances within this cluster.
Please find more details below about specific errors for each gateway
instance."
```

- When you try to create or refresh a report, you might see the following error message:

## Cannot load model

X

Couldn't load the model schema associated with this report. Make sure you have a connection to the server, and try again.

Please try again later or contact support. If you contact support, please provide these details.

**Underlying error code** -2147467259

**Underlying error message** [SAP AG][LIBODBCHDB DLL][HDBODBC] Communication link failure;-10709 Connection failed (RTE:[300012] Cannot create SSL engine: The certificate chain was issued by an authority that is not trusted. (hana2-s4-sso2.westus2.cloudapp.azure.com:30044))

**Correlation ID** 0762cf29-50e1-86a4-acbb-a72329cabd2a

**DM\_ErrorDetailNameCode\_UnderlyingHRESULT** -2147467259

**Microsoft.Data.Mashup.CredentialError.DataSourceKind** SapHana

**Microsoft.Data.Mashup.CredentialError.DataSourceOriginKind** SapHana

**Microsoft.Data.Mashup.CredentialError.DataSourceOriginPath** hana2-s4-sso2.westus2.cloudapp.azure.com:30044

**Microsoft.Data.Mashup.CredentialError.DataSourcePath** hana2-s4-sso2.westus2.cloudapp.azure.com:30044

**Microsoft.Data.Mashup.CredentialError.Reason** EncryptedConnectionFailed

**Microsoft.Data.Mashup.MashupSecurityException.DataSources** [{"kind": "SapHana", "path": "hana2-s4-sso2.westus2.cloudapp.azure.com:30044"}]

**Microsoft.Data.Mashup.MashupSecurityException.Reason** EncryptedConnectionFailed

**Activity ID** ceaeb724-1085-4ac3-9d47-5423cce98235

**Request ID** 66401121-824d-6ccf-6c59-6e9384a264d9

**Time** Wed Sep 30 2020 14:52:26 GMT-0700 (Pacific Daylight Time)

**Service version** 13.0.14487.41

**Client version** 2010.1.03085-train

**Cluster URI** https://df-msit-scus-redirect.analysis.windows.net/

[Get help](#) [Copy](#)

Close 

- When you investigate the *Mashup[date]\*.log*, you see the following error message:

### Output

A connection was successfully established with the server, but then an error occurred during the login process and the certificate chain was issued by an authority that is not trusted.

## Resolution

To resolve this TLS/SSL error, go to the data source connection and then, in the **Validate Server Certificate** section, disable the setting, as shown in the following image:

### SSL

Validate server certificate

After you've disabled this setting, the error message no longer appears.

## Impersonation

Log entries for impersonation contain entries similar to:

#### Output

```
About to impersonate user DOMAIN\User (IsAuthenticated: True,
ImpersonationLevel: Impersonation).
```

The important element in this log entry is the information that's displayed after the `ImpersonationLevel:` entry. Any value different from `Impersonation` reveals that impersonation isn't occurring properly.

## Resolution

You can set up `ImpersonationLevel` properly by following the instructions in [Grant the gateway service account local policy rights on the gateway](#).

After you've changed the configuration file, restart the gateway service for the change to take effect.

## Validation

Refresh or create the report, and then collect the gateway logs. Open the most recent `GatewayInfo` file, and check the following string: `About to impersonate user DOMAIN\User (IsAuthenticated: True, ImpersonationLevel: Impersonation)`. Make sure that the `ImpersonationLevel` setting returns `Impersonation`.

## Delegation

Delegation issues usually appear in the Power BI service as generic errors. To make sure that the issue isn't a delegation issue, collect Wireshark traces and use *Kerberos* as a filter. To learn more about Wireshark, and for information about Kerberos errors, see [Kerberos errors in network captures](#).

The following symptoms and troubleshooting steps can help remedy some common issues.

## SPN issues

If you see the following error: `The import [table] matches no exports. Did you miss a module reference?`: while investigating the `Mashup[date]*.log`, then you're experiencing service principal name (SPN) issues.

When you investigate further by using Wireshark traces, you reveal the error `KRB4KDC_ERR_S_PRINCIPAL_UNKNOW`, which means that the SPN wasn't found or doesn't exist. The following image shows an example:

No.	Time	Source	Destination	Protocol	Length	Info
604	8.855005	10.0.5.6	10.0.5.5	KRB5	206	KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
625	8.922052	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
626	8.923318	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
645	9.039621	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
646	9.040728	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
661	9.060334	10.0.5.5	10.0.5.6	KRB5	2817	TGS-REQ
663	9.061677	10.0.5.6	10.0.5.5	KRB5	206	KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
672	9.069707	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
673	9.070851	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
688	9.087324	10.0.5.5	10.0.5.6	KRB5	2817	TGS-REQ
690	9.088602	10.0.5.6	10.0.5.5	KRB5	206	KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN

▼ Kerberos  
 > Record Mark: 148 bytes  
 > krb-error  
 > pvno: 5  
 msg-type: krb-error (30)  
 stime: 2020-09-29 22:59:06 (UTC)  
 susec: 680360  
 error-code: eRR-S-PRINCIPAL-UNKNOWN (7)  
 realm: HANABQ.COM  
 > sname  
 name-type: KRB5-NT-SRV-INST (2)  
 > sname-string: 2 items  
 SNameString: hdb  
 SNameString: hana2-s4-sso2.westus2.cloudapp.azure.com  
 e-data: 3015a103020103a20e040c720200c0000000003000000

## Resolution

To resolve SPN issues such as this issue, you must add an SPN to a service account. For more information, see the SAP documentation in [Configure Kerberos for SAP HANA database hosts ↗](#).

In addition, follow the resolution instructions described in the next section.

## No credentials issues

There might not be clear symptoms associated with this issue. When you investigate the `Mashup[date]*.log`, you see the following error:

### Output

```
29T20:21:34.6679184Z", "Action": "RemoteDocumentEvaluator/RemoteEvaluation/Han
dleException", "HostProcessId": "1396", "identity": "DirectQueryPool", "Exception
": "Exception:\r\nExceptionType:
Microsoft.Mashup.Engine1.Runtime.ValueException, Microsoft.MashupEngine,
Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35\r\nMessage:
```

When you investigate the same file further, the following (unhelpful) error appears:

### Output

No credentials are available in the security package

Capturing Wireshark traces reveals the following error: KRB5KDC\_ERR\_BADOPTION.

113 6.717377	10.0.5.6	10.0.5.5	KRB5	256 KRB Error: KRB5KDC_ERR_PREAMUTH_REQUIRED
128 6.737432	10.0.5.5	10.0.5.6	KRB5	2817 TGS-REQ
130 6.739193	10.0.5.6	10.0.5.5	KRB5	206 KRB Error: KRB5KDC_ERR_BADOPTION
139 6.750165	10.0.5.5	10.0.5.6	KRB5	284 AS-REQ
140 6.751245	10.0.5.6	10.0.5.5	KRB5	256 KRB Error: KRB5KDC_ERR_PREAMUTH_REQUIRED

▼ Kerberos  
  > Record Mark: 148 bytes  
  ▼ krb-error  
    pvno: 5  
    msg-type: krb-error (30)  
    stime: 2020-09-29 20:27:41 (UTC)  
    susec: 980647  
    error-code: eRR-BADOPTION (13)  
    realm: HANABO.COM  
    ▼ sname  
      name-type: KRB5-NT-SRV-INST (2)  
      ▼ sname-string: 2 items  
        SNameString: hdb  
        SNameString: hana2-s4-sso2.westus2.cloudapp.azure.com  
    > e-data: 3015a103020103a20e040c250200c00000000003000000 PA-PW-SALT



Usually, these errors mean that the SPN `hdb/hana2-s4-sso2.westus2.cloudapp.azure.com` file could be found but isn't in the **Services to which this account can present delegated credentials** list on the **Delegation** pane in the Gateway service account.

## Resolution

To resolve the *No credentials* issue, follow the steps described in [Configure Kerberos constrained delegation](#). When completed properly, the delegation tab at the gateway service account reflects the HansaWorld Database (HDB) file and fully qualified domain name (FQDN) in the list of **Services to which this account can present delegated credentials**.

## Validation

Following the preceding steps should resolve the issue. If you still experience Kerberos issues, you might have a misconfiguration in the Power BI gateway or in the HANA server itself.

## Credentials errors

If you experience credentials errors, errors in the logs or traces expose errors that describe `Credentials are invalid` or similar errors. These errors might manifest differently on the data source side of the connection, such as SAP HANA. The following image shows an example error:

## Cannot load model

X

Couldn't load the model schema associated with this report. Make sure you have a connection to the server, and try again.

Please try again later or contact support. If you contact support, please provide these details.

**Underlying error code** -2147467259

**Underlying error message** The credentials provided cannot be used for the SapHana source. (Source at hana2-s4-sso2.westus2.cloudapp.azure.com:30044.)

**Correlation ID** 177c476c-96a8-c526-7d7e-d642bee2af02

**DM\_ErrorDetailNameCode\_UnderlyingHResult** -2147467259

**Microsoft.Data.Mashup.CredentialError.DataSourceKind** SapHana

**Microsoft.Data.Mashup.CredentialError.DataSourcePath** hana2-s4-sso2.westus2.cloudapp.azure.com:30044

**Microsoft.Data.Mashup.CredentialError.Reason** CredentialInvalid

**Microsoft.Data.Mashup.MashupSecurityException.DataSources** [{"kind": "SapHana", "path": "hana2-s4-sso2.westus2.cloudapp.azure.com:30044"}]

**Microsoft.Data.Mashup.MashupSecurityException.Reason** CredentialInvalid

**Activity ID** ceaeb724-1085-4ac3-9d47-5423cce98235

**Request ID** 4d8440d6-7f88-64e2-9dd6-3e9aaefad8f

**Time** Tue Sep 29 2020 12:04:35 GMT-0700 (Pacific Daylight Time)

**Service version** 13.0.14437.52

**Client version** 2009.4.03061-train

**Cluster URI** https://df-msit-scus-redirect.analysis.windows.net/

[Get help](#) [Copy](#)

Close 

## Symptom 1

In HANA authentication traces, you might see entries similar to the following message:

Output

```
[Authentication|manager.cpp:166] Kerberos: Using Service Principal
Name johnny@contoso.com@CONTOSO.COM with name type:
GSS_KRB5_NT_PRINCIPAL_NAME
[Authentication|methodgssinitiator.cpp:367] Got principal name:
johnny@contoso.com@CONTOSO.COM
```

## Resolution

Follow the instructions described in [Set user-mapping configuration parameters on the gateway machine](#), even if you've already configured the [Azure AD Connect service](#).

## Validation

After you've completed the validation, you can successfully load the report in the Power BI service.

## Symptom 2

In HANA authentication traces, you might see entries similar to the following entry:

#### Output

```
Authentication ManagerAcceptor.cpp(00233) : Extending list of expected
external names by johnny@CONTOSO.COM (method: GSS) Authentication
AuthenticationInfo.cpp(00168) : ENTER getAuthenticationInfo
(externalName=johnny@CONTOSO.COM) Authentication
AuthenticationInfo.cpp(00237) :
Found no user with expected external name!
```

## Resolution

Check the Kerberos external ID under **HANA User** to determine whether the IDs match properly.

## Validation

After you've resolved the issue, you can create or refresh reports in the Power BI service.

## Next steps

For more information about the on-premises data gateway and DirectQuery, see the following resources:

- [What is an on-premises data gateway?](#)
- [DirectQuery in Power BI](#)
- [Data sources supported by DirectQuery](#)
- [DirectQuery and SAP Business Warehouse \(BW\)](#)
- [DirectQuery and SAP HANA](#)

# Use Kerberos single sign-on for SSO to SAP BW using CommonCryptoLib (sapcrypto.dll)

Article • 03/29/2022

This article describes how to configure your SAP BW data source to enable SSO from the Power BI service by using CommonCryptoLib (sapcrypto.dll).

## ⓘ Note

Before you attempt to refresh a SAP BW-based report that uses Kerberos SSO, complete both the steps in this article and the steps in [Configure Kerberos SSO](#). Using CommonCryptoLib as your SNC library enables SSO connections to both SAP BW Application Servers and SAP BW Message Servers.

## ⓘ Note

Configuring both libraries(sapcrypto and gx64krb5) on the same gateway server is an unsupported scenario. It's not recommended to configure both libraries on the same gateway server as it'll lead to a mix of libraries. If you want to use both libraries, please fully separate the gateway server. For example, configure gx64krb5 for server A then sapcrypto for server B. Please remember that any failure on server A which uses gx64krb5 is not supported, as gx64krb5 is no longer supported by SAP and Microsoft.

## Configure SAP BW to enable SSO using CommonCryptoLib

## ⓘ Note

The on-premises data gateway is 64-bit software and therefore requires the 64-bit version of CommonCryptoLib (sapcrypto.dll) to perform BW SSO. If you plan to test the SSO connection to your SAP BW server in SAP GUI prior to attempting an SSO connection through the gateway (recommended), you'll also need the 32-bit version of CommonCryptoLib, as SAP GUI is 32-bit software.

1. Ensure that your BW server is correctly configured for Kerberos SSO using CommonCryptoLib. If it is, you can use SSO to access your BW server (either directly or through an SAP BW Message Server) with an SAP tool like SAP GUI that has been configured to use CommonCryptoLib.

For more information on setup steps, see [SAP Single Sign-On: Authenticate with Kerberos/SPNEGO](#). Your BW server should use CommonCryptoLib as its SNC Library and have an SNC name that starts with *CN=*, such as *CN=BW1*. For more information on SNC name requirements (specifically, the *snc/identity/as* parameter), see [SNC Parameters for Kerberos Configuration](#).

2. If you haven't already done so, install the x64-version of the [SAP .NET Connector](#) on the computer the gateway has been installed on.

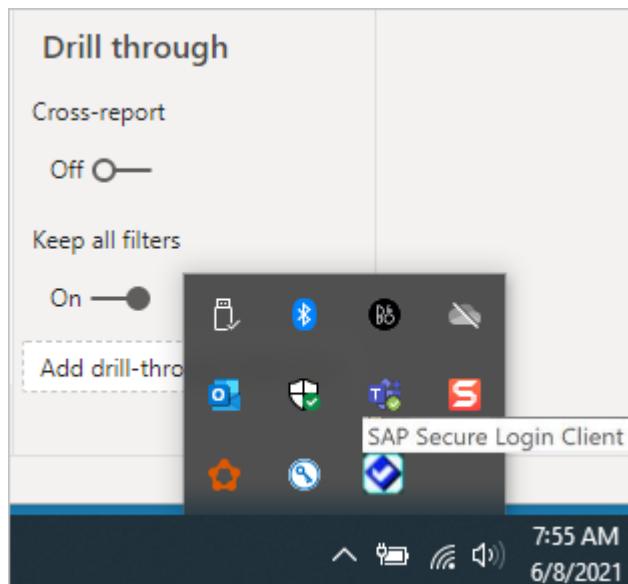
You can check whether the component has been installed by attempting to connect to your BW server in Power BI Desktop from the gateway computer. If you can't connect by using the 2.0 implementation, the .NET Connector isn't installed or hasn't been installed to the GAC.

3. Ensure that SAP Secure Login Client (SLC) isn't running on the computer the gateway is installed on.

SLC caches Kerberos tickets in a way that can interfere with the gateway's ability to use Kerberos for SSO.

4. If SLC is installed, uninstall it or make sure you exit SAP Secure Login Client. Right-click the icon in the system tray and select **Log Out** and **Exit** before you attempt an SSO connection by using the gateway.

SLC isn't supported for use on Windows Server machines. For more information, see [SAP Note 2780475](#) (s-user required).



5. If you uninstall SLC or select **Log Out** and **Exit**, open a cmd window and enter `klist purge` to clear any cached Kerberos tickets before you attempt an SSO connection through the gateway.
6. Download 64-bit CommonCryptoLib (sapcrypto.dll) version *8.5.25 or greater* from the SAP Launchpad, and copy it to a folder on your gateway machine. In the same directory where you copied sapcrypto.dll, create a file named sapcrypto.ini, with the following content:

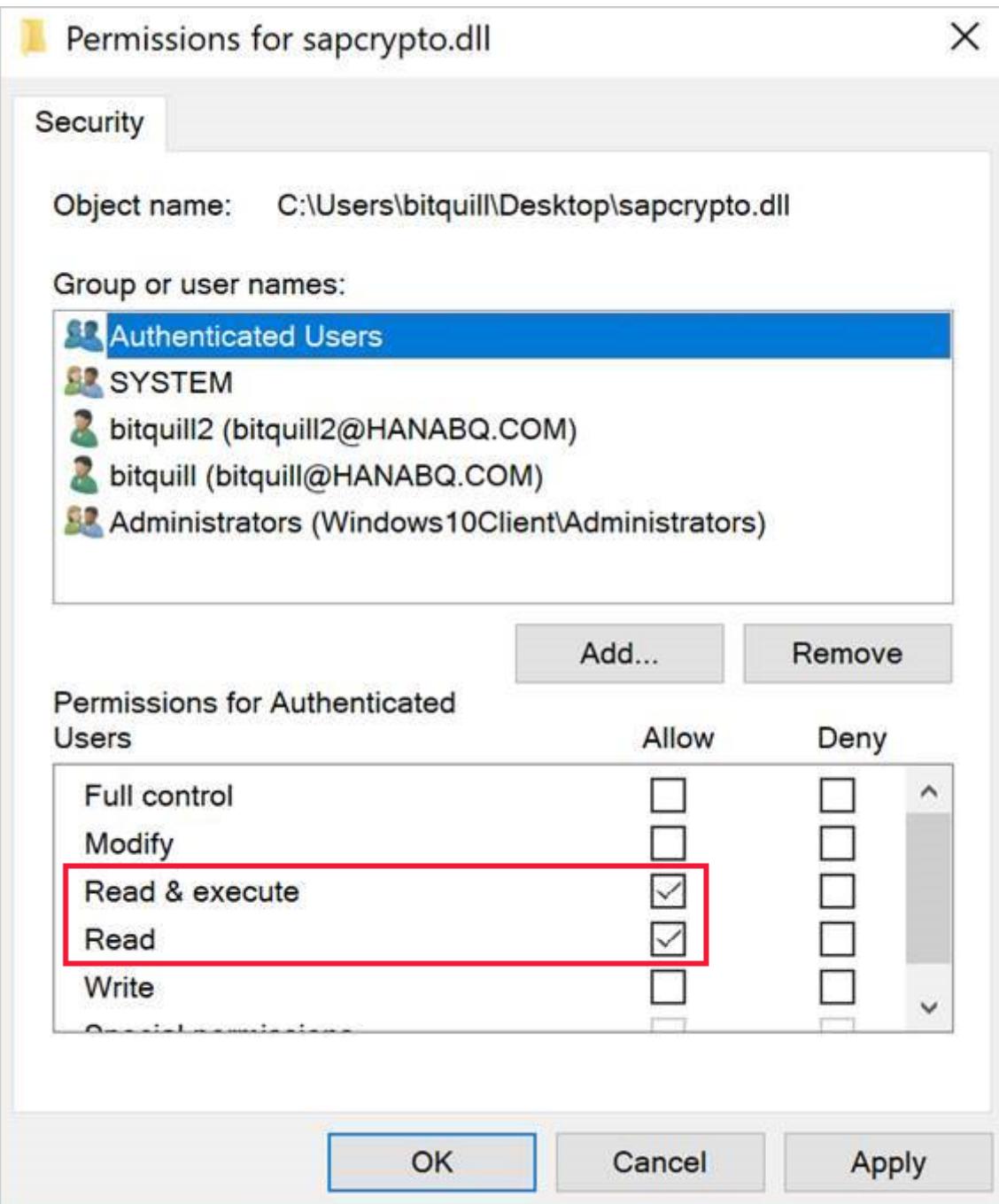
```
ccl/snc/enable_kerberos_in_client_role = 1
```

The .ini file contains configuration information required by CommonCryptoLib to enable SSO in the gateway scenario.

 **Note**

These files must be stored in the same location; in other words, `/path/to/sapcrypto/` should contain both sapcrypto.ini and sapcrypto.dll.

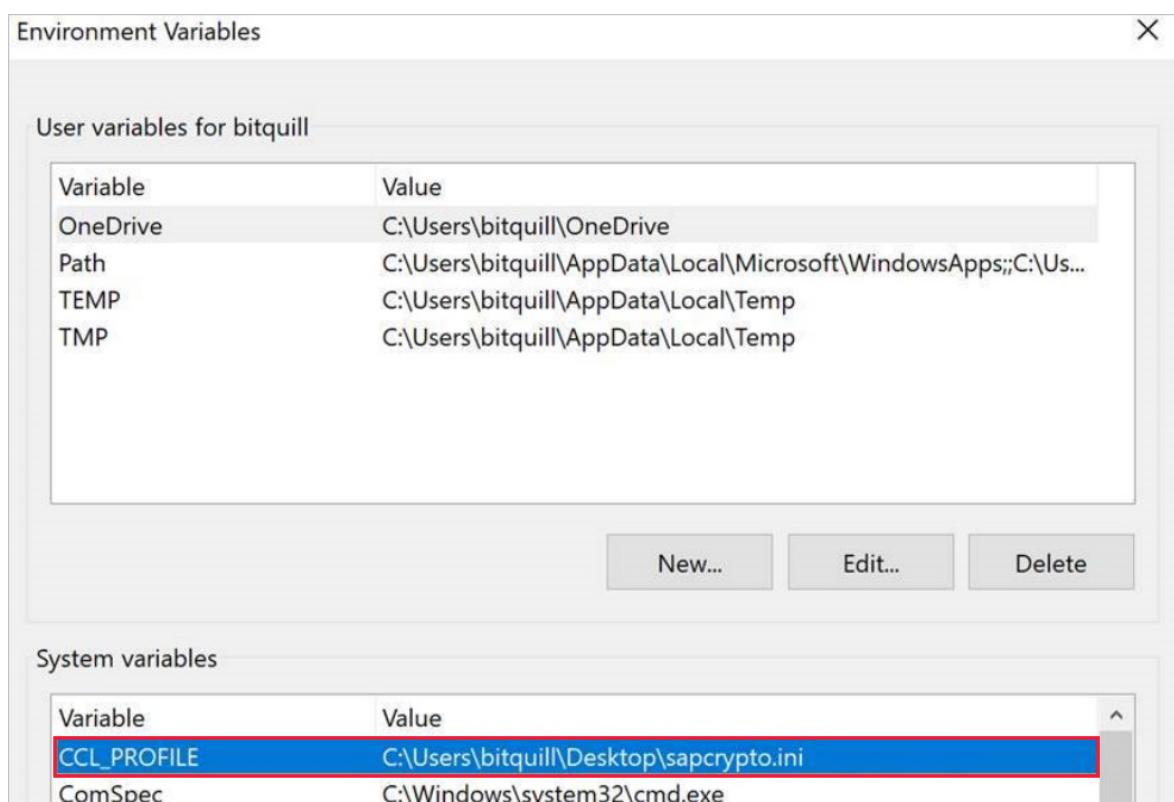
Both the gateway service user and the Active Directory (AD) user that the service user impersonates need read and execute permissions for both files. We recommend granting permissions on both the .ini and .dll files to the Authenticated Users group. For testing purposes, you can also explicitly grant these permissions to both the gateway service user and the Active Directory user you use for testing. In the following screenshot we've granted the Authenticated Users group **Read & execute** permissions for sapcrypto.dll:



7. If you don't already have an SAP BW data source associated with the gateway you want the SSO connection to flow through, add one on the **Manage gateways** page in the Power BI service. If you already have such a data source, edit it:
  - Choose **SAP Business Warehouse** as the **Data Source Type** if you want to create an SSO connection to a BW Application Server.
  - Select **Sap Business Warehouse Message Server** if you want to create an SSO connection to a BW Message Server.
8. For **SNC Library**, select either the **SNC\_LIB** or **SNC\_LIB\_64** environment variable, or **Custom**.
  - If you select **SNC\_LIB**, you must set the value of the **SNC\_LIB\_64** environment variable on the gateway machine to the absolute path of the 64-bit copy of

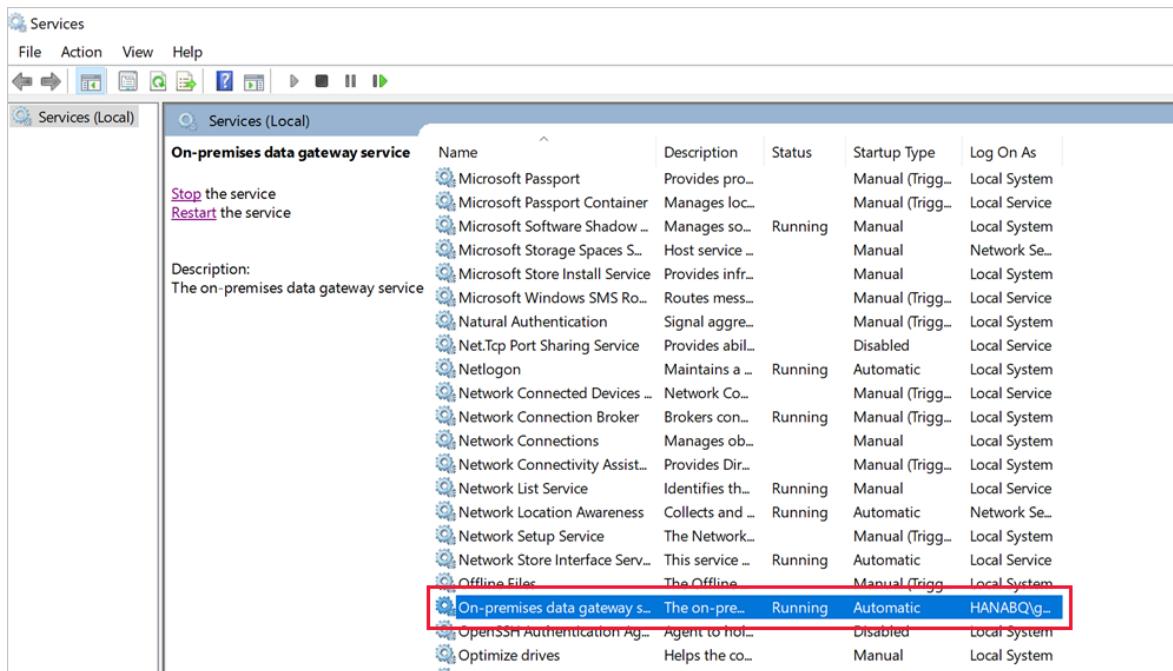
sapcrypto.dll on the gateway machine. For example,  
C:\Users\Test\Desktop\sapcrypto.dll.

- If you choose **Custom**, paste the absolute path to *sapcrypto.dll* into the Custom SNC Library Path field that appears on the **Manage gateways** page.
9. For **SNC Partner Name**, enter the SNC Name of the BW server. Under **Advanced settings**, ensure that **Use SSO via Kerberos for DirectQuery queries** is checked. Fill in the other fields as if you were establishing a Windows Authentication connection from PBI Desktop.
10. Create a **CCL\_PROFILE** system environment variable and set its value to the path to sapcrypto.ini.



The sapcrypto .dll and .ini files must exist in the same location. In the above example, sapcrypto.ini and sapcrypto.dll are both located on the desktop.

11. Restart the gateway service.



## 12. Run a Power BI report

# Troubleshooting

If you're unable to refresh the report in the Power BI service, you can use gateway tracing, CPIC tracing, and CommonCryptoLib tracing to diagnose the issue. Because CPIC tracing and CommonCryptoLib are SAP products, Microsoft can't provide support for them.

## Gateway logs

1. Reproduce the issue.
2. Open the [gateway app](#), and select **Export logs** from the **Diagnostics** tab.



## On-premises data gateway

Status	Additional logging	<input checked="" type="checkbox"/>
Service Settings	You can enable additional logging to output queries and their timings to help understand what is performing slow. It is not recommended to leave this setting enabled long term.	
Diagnostics		
Network	<a href="#">Learn more</a>	
Connectors	<p>Gateway logs</p> <p>Export all of the gateway's configuration and service logs to a single .zip file.</p> <p><a href="#">Export logs</a></p>	

## CPIC tracing

1. To enable CPIC tracing, set two environment variables: **CPIC\_TRACE** and **CPIC\_TRACE\_DIR**.

The first variable sets the trace level and the second variable sets the trace file directory. The directory must be a location that members of the Authenticated Users group can write to.

2. Set **CPIC\_TRACE** to 3 and **CPIC\_TRACE\_DIR** to whichever directory you want the trace files written to. For example:

CPIC_TRACE	3
CPIC_TRACE_DIR	C:\traces\cpic

3. Reproduce the issue and ensure that **CPIC\_TRACE\_DIR** contains trace files.

CPIC tracing can diagnose higher level issues such as a failure to load the sapcrypto.dll library. For example, here is a snippet from a CPIC trace file where a .dll load error occurred:

### Output

```
[Thr 7228] *** ERROR => D1LoadLib()==DLENOACCESS -
LoadLibrary("C:\Users\test\Desktop\sapcrypto.dll")
Error 5 = "Access is denied." [d1nt.c 255]
```

If you encounter such a failure but you've set the Read & Execute permissions on sapcrypto.dll and sapcrypto.ini as described [in the section above](#), try setting the same Read & Execute permissions on the folder that contains the files.

If you're still unable to load the .dll, try turning on [auditing for the file](#). Examining the resulting audit logs in the Windows Event Viewer might help you determine why the file is failing to load. Look for a failure entry initiated by the impersonated Active Directory user. For example, for the impersonated user MYDOMAIN\mytestuser a failure in the audit log would look something like this:

#### Output

A handle to an object was requested.

**Subject:**

Security ID: MYDOMAIN\mytestuser  
Account Name: mytestuser  
Account Domain: MYDOMAIN  
Logon ID: 0xCF23A8

**Object:**

Object Server: Security  
Object Type: File  
Object Name: <path information>\sapcrypto.dll  
Handle ID: 0x0  
Resource Attributes: -

**Process Information:**

Process ID: 0x2b4c  
Process Name: C:\Program Files\On-premises data gateway\Microsoft.Mashup.Container.NetFX45.exe

**Access Request Information:**

Transaction ID: {00000000-0000-0000-0000-000000000000}  
Accesses: ReadAttributes

Access Reasons: ReadAttributes: Not granted

Access Mask: 0x80  
Privileges Used for Access Check: -  
Restricted SID Count: 0

## CommonCryptoLib tracing

1. Turn on CommonCryptoLib tracing by adding these lines to the sapcrypto.ini file you created earlier:

```
ccl/trace/level=5
ccl/trace/directory=<drive>:\logs\sectrace
```

2. Change the `ccl/trace/directory` option to a location to which members of the Authenticated Users group can write.
3. Alternatively, create a new .ini file to change this behavior. In the same directory as sapcrypto.ini and sapcrypto.dll, create a file named sectrace.ini, with the following content. Replace the `DIRECTORY` option with a location on your machine that members of the Authenticated Users group can write to:

```
LEVEL = 5
DIRECTORY = <drive>:\logs\sectrace
```

4. Reproduce the issue and verify that the location pointed to by `DIRECTORY` contains trace files.
5. When you're finished, turn off CPIC and CCL tracing.

For more information on CommonCryptoLib tracing, see [SAP Note 2491573](#) (SAP s-user required).

## Impersonation

This section describes troubleshooting symptoms and resolution steps for impersonation issues.

**Symptom:** When looking at the `GatewayInfo[date].log` you find an entry similar to the following: **About to impersonate user DOMAIN\User (IsAuthenticated: True, ImpersonationLevel: Impersonation)**. If the value for **ImpersonationLevel** is different from **Impersonation**, impersonation is not happening properly.

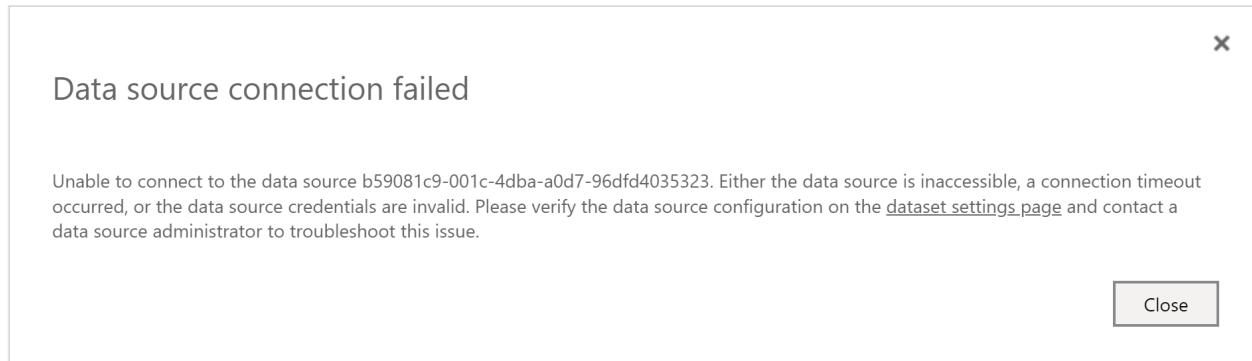
**Resolution:** Follow the steps found in [grant the gateway service account local policy rights on the gateway machine](#) article. Restart the gateway service after changing the configuration.

**Validation:** Refresh or create the report and collect the `GatewayInfo[date].log`. Open the latest GatewayInfo log file and check again the following string: **About to impersonate user DOMAIN\User (IsAuthenticated: True, ImpersonationLevel: Impersonation)** to ensure that the value for **ImpersonationLevel** matches **Impersonation**.

# Delegation

Delegation issues usually appear in the Power BI service as generic errors. To determine whether delegation is the issue, it's useful to collect the Wireshark traces and use *Kerberos* as a filter. For Kerberos errors reference, consult the [blog post](#). The rest of this section describes troubleshooting symptoms and resolution steps for delegation issues.

**Symptom:** In the Power BI service you may encounter an unexpected error, similar to the following screenshot. the *GatewayInfo[date].log* you'll see *[DM.GatewayCore]* ingesting an exception during Ado query execution attempt for *clientPipelinelId* and the import *[OD\_NW\_CHANN]* matches no exports.



In the *Mashup[date].log* you see the generic error **GSS-API(maj): No credentials were supplied.**

Looking into the CPIC traces (sec-Microsoft.Mashup\*.trc) you will see something similar to the following:

```
[Thr 4896] *** ERROR => SncPEstablishContext() failed for target='p:CN=BW5' [sncxxall.c 3638]
[Thr 4896] *** ERROR => SncPEstablishContext()==SNCERR_GSSAPI [sncxxall.c 3604]
[Thr 4896] GSS-API(maj): No credentials were supplied
[Thr 4896] Unable to establish the security context
[Thr 4896] target="p:CN=BW5"
[Thr 4896] <<- SncProcessOutput()==SNCERR_GSSAPI
[Thr 4896]
[Thr 4896] LOCATION CPIC (TCP/IP) on local host HNCL2 with Unicode
[Thr 4896] ERROR GSS-API(maj): No credentials were supplied
[Thr 4896] Unable to establish the security context
[Thr 4896] target="p:CN=BW5"
[Thr 4896] TIME Thu Oct 15 20:49:31 2020
[Thr 4896] RELEASE 721
[Thr 4896] COMPONENT SNC (Secure Network Communication)
[Thr 4896] VERSION 6
[Thr 4896] RC -4
[Thr 4896] MODULE sncxxall.c
```

```
[Thr 4896] LINE 3604
[Thr 4896] DETAIL SncPEstablishContext
[Thr 4896] SYSTEM CALL gss_init_sec_context
[Thr 4896] COUNTER 3
[Thr 4896]
[Thr 4896] *** ERROR => STISEND:STISncOut failed 20 [r3cpic.c 9834]
[Thr 4896] STISearchConv: found conv without search
```

The error becomes clearer in the sectraces from the Gateway machine *sec-Microsoft.Mashup.Con-[]*.trc:

```
[2020.10.15 20:31:38.396000][4][Microsoft.Mashup.Con][Kerberos][3616]
AcquireCredentialsHandleA called successfully.
[2020.10.15 20:31:38.396000][2][Microsoft.Mashup.Con][Kerberos][3616]
InitializeSecurityContextA returned -2146893053 (0x80090303). Preparation
for kerberos failed!
[2020.10.15 20:31:38.396000][2][Microsoft.Mashup.Con][Kerberos][3616]
Getting kerberos ticket for 'SAP/BW5' failed (user name is
affonso_v@HANABQ.COM)
[2020.10.15 20:31:38.396000][2][Microsoft.Mashup.Con][Kerberos][3616]
Error for requested algorithm 18: 0/C000018B The security database on the
server does not have a computer account for this workstation trust
relationship.
[2020.10.15 20:31:38.396000][2][Microsoft.Mashup.Con][Kerberos][3616]
Error for requested algorithm 17: 0/C000018B The security database on the
server does not have a computer account for this workstation trust
relationship.
[2020.10.15 20:31:38.396000][2][Microsoft.Mashup.Con][Kerberos][3616]
Error for requested algorithm 23: 0/C000018B The security database on the
server does not have a computer account for this workstation trust
relationship.
[2020.10.15 20:31:38.396000][2][Microsoft.Mashup.Con][Kerberos][3616]
Error for requested algorithm 3: 0/C000018B The security database on the
server does not have a computer account for this workstation trust
relationship.
```

You can also see the issue if you look at Wireshark traces.

No.	Time	Source	Destination	Protocol	Length	Info
123	11.196755	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
124	11.197813	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
131	11.200120	10.0.5.5	10.0.5.6	KRB5	286	AS-REQ
132	11.201087	10.0.5.6	10.0.5.5	KRB5	258	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
139	11.203914	10.0.5.5	10.0.5.6	KRB5	366	AS-REQ
141	11.204958	10.0.5.6	10.0.5.5	KRB5	178	AS-REP
149	11.206143	10.0.5.5	10.0.5.6	KRB5	1660	TGS-REQ
152	11.208280	10.0.5.6	10.0.5.5	KRB5	198	TGS-REP
175	11.333402	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
176	11.334472	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
197	11.380187	10.0.5.5	10.0.5.6	KRB5	2780	TGS-REQ
199	11.386278	10.0.5.6	10.0.5.5	KRB5		167 KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN

▼ Kerberos

- > Record Mark: 109 bytes
- ▼ krb-error
  - pvno: 5
  - msg-type: krb-error (30)
  - stime: 2020-10-15 20:31:37 (UTC)
  - susec: 313461
  - error-code: eR-S-PRINCIPAL-UNKNOWN (7)
  - realm: HANABQ.COM
- ▼ sname
  - name-type: KRB5-NT-SRV-INST (2)
  - ▼ sname-string: 2 items
    - SNameString: SAP
    - SNameString: BW5

e-data: 3015a103020103a20e040c130400c00000000003000000

## ⓘ Note

The other errors **KRB5KDC\_ERR\_PREAUTH\_REQUIRED** can be safely ignored.

**Resolution:** You must add an SPN SAP/BW5 to a service account. Detailed information and steps are available in the [SAP documentation ↗](#).

You may run into a similar, but not identical error that manifests in Wireshark traces as the following error **KRB5KDC\_ERR\_BADOPTION**:

No.	Time	Source	Destination	Protocol	Length	Info
212	6.2223974	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
213	6.225221	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
220	6.227375	10.0.5.5	10.0.5.6	KRB5	1660	TGS-REQ
223	6.229514	10.0.5.6	10.0.5.5	KRB5	198	TGS-REP
248	6.721624	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
249	6.722961	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
484	9.811768	10.0.5.5	10.0.5.6	KRB5	2780	TGS-REQ
486	9.813352	10.0.5.6	10.0.5.5	KRB5		167 KRB Error: KRB5KDC_ERR_BADOPTION
421	10.247972	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
422	10.249059	10.0.5.6	10.0.5.5	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
460	10.582092	10.0.5.5	10.0.5.6	KRB5	284	AS-REQ
461	10.583386	10.0.5.5	10.0.5.6	KRB5	256	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED

▼ Kerberos

- > Record Mark: 109 bytes
- ▼ krb-error
  - pvno: 5
  - msg-type: krb-error (30)
  - stime: 2020-10-15 21:30:09 (UTC)
  - susec: 631780
  - error-code: eR-BADOPTION (13)
  - realm: HANABQ.COM
- ▼ sname
  - name-type: KRB5-NT-SRV-INST (2)
  - ▼ sname-string: 2 items
    - SNameString: SAP
    - SNameString: BW5

e-data: 3015a103020103a20e040c130400c00000000003000000 PA-PW-SALT

This error indicates the **SPN SAP/BW5** could be found, but it's not in the *Services to which this account can present delegated credentials* at the Delegation tab from the Gateway service account. To fix this issue, follow the steps to [configure the gateway service account for standard kerberos constrained delegation](#).

**Validation:** Proper configuration will prevent generic or unexpected errors to be presented by the gateway. If you still see errors, check the configuration of the gateway itself, or the configuration of the BW server.

## Credentials errors

This section describes troubleshooting symptoms and resolution steps for credentials error issues. You may also see generic errors from the Power BI service, as described in the earlier section on [delegation](#).

There are different resolutions, based on the symptoms you see in the data source (SAP BW), so we'll review both.

**Symptom 1:** In the *sectraces sec-disp+work[].trc* from the BW Server, you see traces similar to the following:

```
[2020.05.26 14:21:28.668325][4][disp+work][SAPCRYPTOLIB][435584] {
 gss_display_name [2020.05.26 14:21:28.668338][4][disp+work][GSS][435584]
 gss_display_name output buffer (41 bytes) [2020.05.26 14:21:28.668338][4]
 [disp+work][GSS][435584] CN=DAVID@XS.CONTOSO.COM@CONTOSO.COM
```

**Resolution:** Complete the configuration steps to [set user mapping configuration parameters on the gateway machine if necessary](#). You'll need to complete those steps even if you already have the Azure AD Connect configured.

**Validation:** You'll be able to successfully load the report in the Power BI service. If not successful, see the steps in symptom 2.

**Symptom 2:** In the *sectraces sec-disp+work[].trc* from the BW Server, you see traces similar to the following:

```
[2020.10.19 23:10:15.469000][4][disp+work.EXE][SAPCRYPTOLIB][4460] {
 gss_display_name
[2020.10.19 23:10:15.469000][4][disp+work.EXE][GSS][4460]
 gss_display_name output buffer (23 bytes)
[2020.10.19 23:10:15.469000][4][disp+work.EXE][GSS][4460]
CN=DAVID@CONTOSO.COM
```

**Resolution:** Check whether the Kerberos external ID for the User match what the sectraces are showing.

1. Open SAP Logon.
2. Use the SU01 transaction.
3. Edit the user.
4. Navigate to the SNC tab, verify that the SNC name matches what is shown in your logs.

**Validation:** When properly completed, you'll be able to create and refresh reports in the Power BI service.

## Next steps

For more information about the on-premises data gateway and DirectQuery, see the following resources:

- [What is an on-premises data gateway?](#)
- [DirectQuery in Power BI](#)
- [Data sources supported by DirectQuery](#)
- [DirectQuery and SAP BW](#)
- [DirectQuery and SAP HANA](#)

# Use Kerberos for single sign-on (SSO) to SAP BW using gx64krb5

Article • 03/29/2022

This article describes how to configure your SAP BW data source to enable SSO from the Power BI service by using gx64krb5.

## ⓘ Important

Microsoft will allow you to create connections using SNC libraries (like gx64krb5) but will not provide support for these configurations. Additionally SAP no longer supports the gx64krb5 for on-premises data gateways in Power BI and the steps required to configure it for the gateway are significantly more complex compared to CommonCryptoLib. As a result, Microsoft recommends using CommonCryptoLib instead. For more information, see [SAP Note 352295](#). Note that gx64krb5 doesn't allow for SSO connections from the data gateway to SAP BW Message Servers; only connections to SAP BW Application Servers are possible. This restriction doesn't exist if you use **CommonCryptoLib** as your SNC library. For information about how to configure SSO by using CommonCryptoLib, see [Configure SAP BW for SSO using CommonCryptoLib](#). Use CommonCryptoLib or gx64krb5 as your SNC library, but not both. Do not complete the configuration steps for both libraries.

## ⓘ Note

Configuring both libraries(sapcrypto and gx64krb5) on the same gateway server is an unsupported scenario. It's not recommended to configure both libraries on the same gateway server as it'll lead to a mix of libraries. If you want to use both libraries, please fully separate the gateway server. For example, configure gx64krb5 for server A then sapcrypto for server B. Please remember that any failure on server A which uses gx64krb5 is not supported, as gx64krb5 is no longer supported by SAP and Microsoft.

This guide is comprehensive; if you've already completed some of the described steps, you can skip them. For example, you might have already configured your SAP BW server for SSO using gx64krb5.

# Set up gx64krb5 on the gateway machine and the SAP BW server

The gx64krb5 library must be used by both the client and server to complete an SSO connection through the gateway. That is, both the client and server must be using the same SNC library.

1. Download gx64krb5.dll from [SAP Note 2115486](#) (SAP s-user required). Ensure you have at least version 1.0.11.x. Also, download gsskrb5.dll (the 32-bit version of the library) if you want to test the SSO connection in SAP GUI before you attempt the SSO connection through the gateway (recommended). The 32-bit version is required to test with SAP GUI because SAP GUI is 32-bit only.
2. Put gx64krb5.dll in a location on your gateway machine that's accessible by your gateway service user. If you want to test the SSO connection with SAP GUI, also put a copy of gsskrb5.dll on your machine and set the **SNC\_LIB** environment variable to point to it. Both the gateway service user and the Active Directory (AD) users that the service user will impersonate need read and execute permissions for the copy of gx64krb5.dll. We recommend granting permissions on the .dll to the Authenticated Users group. For testing purposes, you can also explicitly grant these permissions to both the gateway service user and the Active Directory user you use to test.
3. If your BW server hasn't already been configured for SSO using gx64krb5.dll, put another copy of the .dll on your SAP BW server machine in a location accessible by the SAP BW server.

For more information on configuring gx64krb5.dll for use with an SAP BW server, see [SAP documentation](#) (SAP s-user required).
4. On the client and server machines, set the **SNC\_LIB** and **SNC\_LIB\_64** environment variables:
  - If you use gsskrb5.dll, set the **SNC\_LIB** variable to its absolute path.
  - If you use gx64krb5.dll, set the **SNC\_LIB\_64** variable to its absolute path.

## Configure an SAP BW service user and enable SNC communication on the BW server

Complete this section if you haven't already configured your SAP BW server for SNC communication (for example, SSO) by using gx64krb5.

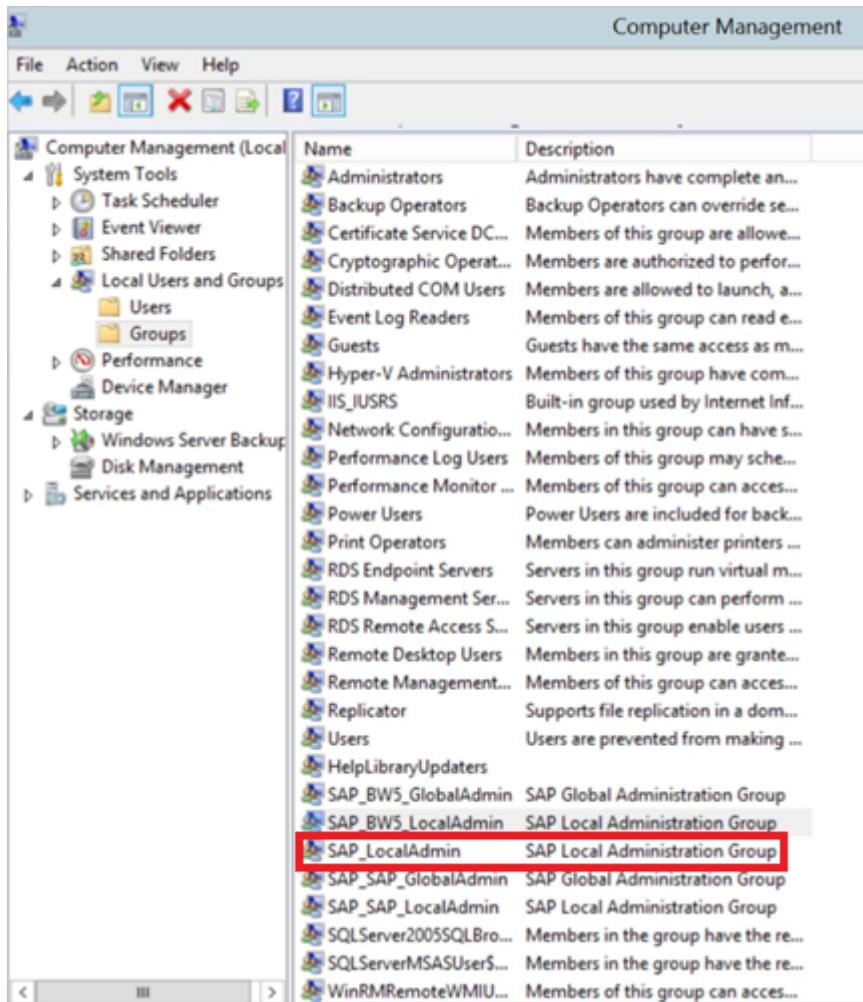
## Note

This section assumes that you've already created a service user for BW and bound a suitable SPN to it (that is, a name that begins with *SAP/*).

1. Give the service user access to your SAP BW Application Server:

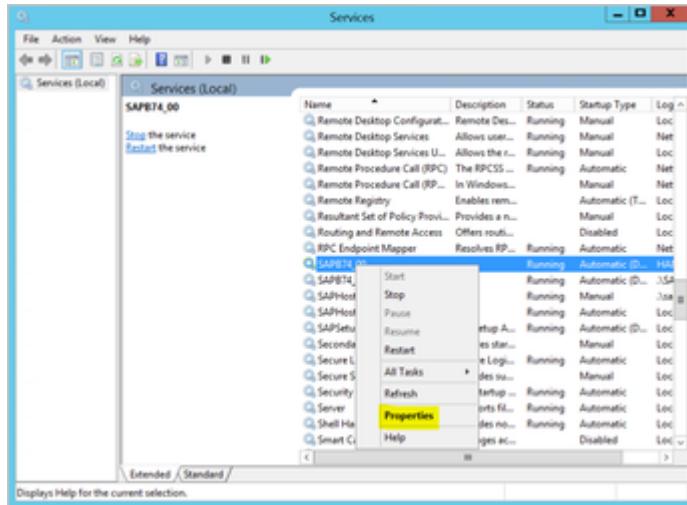
- a. On the SAP BW server machine, add the service user to the Local Admin group.

Open the **Computer Management** program and identify the Local Admin group for your server.



- b. Double-click the Local Admin group, and select **Add** to add your service user to the group.
    - c. Select **Check Names** to ensure you've entered the name correctly, and then select **OK**.
  2. Set the SAP BW server's service user as the user that starts the SAP BW server service on the SAP BW server machine:
    - a. Open Run, and then enter **Services.msc**.

- b. Find the service corresponding to your SAP BW Application Server instance, right-click it, and then select **Properties**.



- c. Switch to the **Log on** tab, and change the user to your SAP BW service user.
- d. Enter the user's password, and then select **OK**.
3. In SAP Logon, sign in to your server and set the following profile parameters by using the RZ10 transaction:
- Set the **snc/identity/as** profile parameter to *p:<SAP BW service user you created>*. For example, *p:BWServiceUser@MYDOMAIN.COM*. Note that *p:* precedes the service user's UPN, as opposed to *p:CN=*, which precedes the UPN when you use CommonCryptoLib as the SNC library.
  - Set the **snc/gssapi\_lib** profile parameter to *<path to gx64krb5.dll on the BW server>*. Place the library in a location that the SAP BW Application Server can access.
  - Set the following additional profile parameters, changing the values as required to fit your needs. The last five options enable clients to connect to the SAP BW server by using SAP Logon without having SNC configured.

Setting	Value
snc/data_protection/max	3
snc/data_protection/min	1
snc/data_protection/use	9
snc/accept_insecure_cpic	1
snc/accept_insecure_gui	1

Setting	Value
snc/accept_insecure_r3int_rfc	1
snc/accept_insecure_rfc	1
snc/permit_insecure_start	1

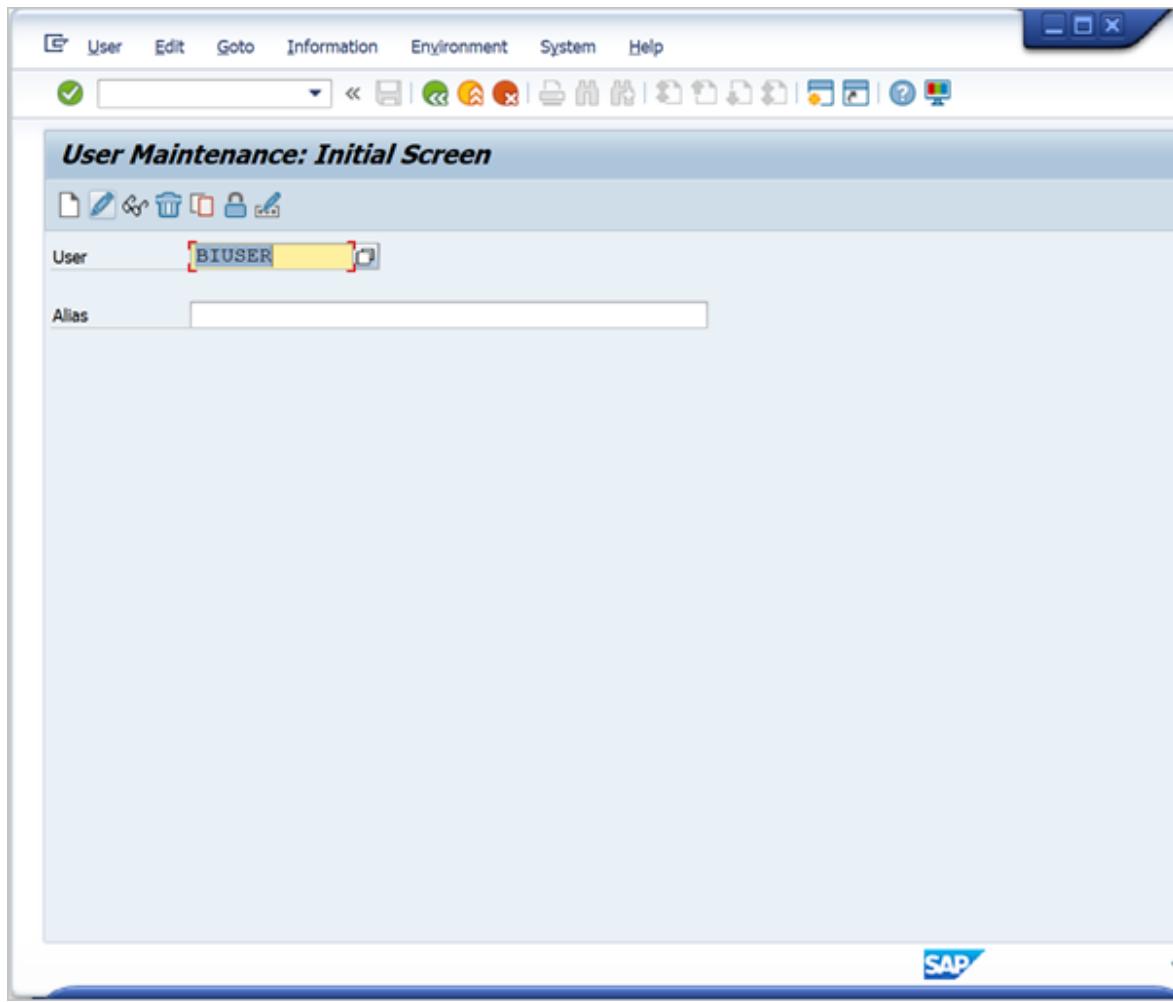
- d. Set the **snc/enable** property to 1.
- 4. After you set these profile parameters, open the SAP Management Console on the server machine and restart the SAP BW instance.

If the server won't start, confirm that you've set the profile parameters correctly. For more information on profile parameter settings, see the [SAP documentation](#). You can also consult the [Troubleshooting](#) section in this article.

## Map an SAP BW user to an Active Directory user

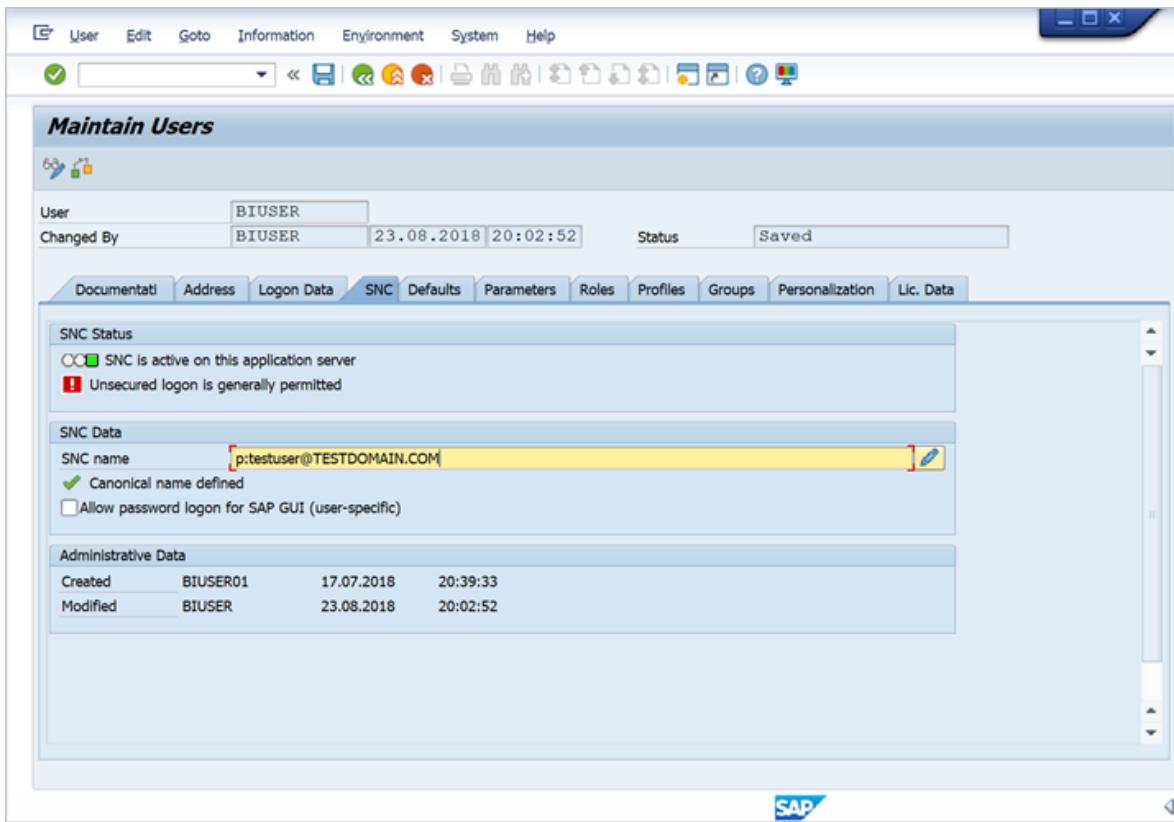
If you haven't done so already, map an Active Directory user to an SAP BW Application Server user and test the SSO connection in SAP Logon.

1. Sign in to your SAP BW server with SAP Logon. Run transaction SU01.
2. For **User**, enter the SAP BW user for which you want to enable SSO connection. Select the **Edit** icon (pen icon) near the top left of the SAP Logon window.



3. Select the **SNC** tab. In the SNC name input box, enter *p:<your Active Directory user>@<your domain>*. For the SNC name, *p:* must precede the Active Directory user's UPN. Note that the UPN is case-sensitive.

The Active Directory user you specify should belong to the person or organization for whom you want to enable SSO access to the SAP BW Application Server. For example, if you want to enable SSO access for the user testuser@TESTDOMAIN.COM, enter *p:testuser@TESTDOMAIN.COM*.

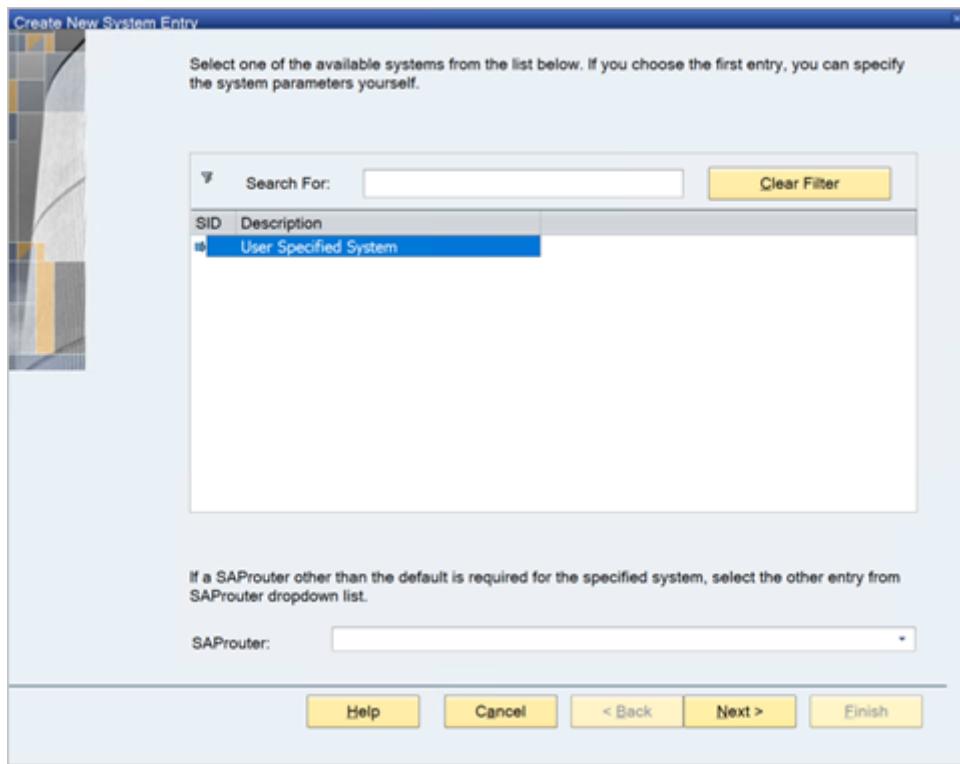


4. Select the **Save** icon (floppy disk image) near the top left of the screen.

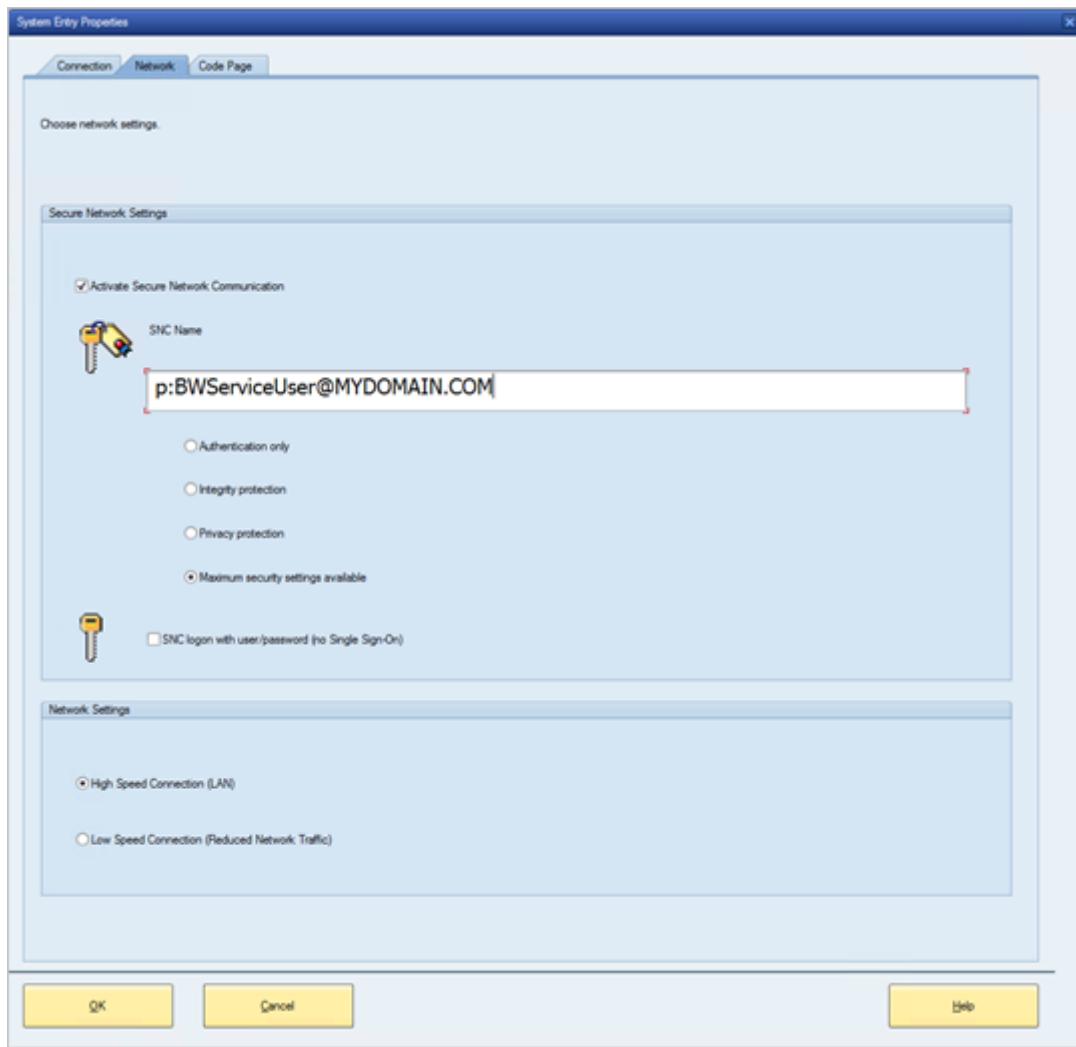
## Test sign in via SSO

Verify that you can sign in to the server by using SAP Logon through SSO as the Active Directory user for whom you've enabled SSO access:

1. As the Active Directory user for which you've just enabled SSO access, sign in to a machine in your domain on which SAP Logon is installed. Launch SAP Logon, and create a new connection.
2. Copy the gsskrb5.dll file you downloaded earlier to a location on the machine you signed-in to. Set the **SNC\_LIB** environment variable to the absolute path of this location.
3. Launch SAP Logon, and create a new connection.
4. In the **Create New System Entry** screen, select **User Specified System**, then select **Next**.



5. Fill in the appropriate details on the next screen, including the application server, instance number, and system ID. Then, select **Finish**.
6. Right-click the new connection, select **Properties**, and then select the **Network** tab.
7. In the **SNC Name** box, enter *p:< SAP BW service user's UPN >*. For example, *p:BWSERVICEUSER@MYDOMAIN.COM*. Select **OK**.



- Double-click the connection you just created to attempt an SSO connection to your SAP BW server.

If this connection succeeds, continue to the next section. Otherwise, review the earlier steps in this document to make sure they've been completed correctly, or review the [Troubleshooting](#) section. If you can't connect to the SAP BW server via SSO in this context, you won't be able to connect to the SAP BW server by using SSO in the gateway context.

## Add registry entries to the gateway machine

Add required registry entries to the registry of the machine that the gateway is installed on, and to machines intended to connect from Power BI Desktop. To add these registry entries, run the following commands:

- `REG ADD HKLM\SOFTWARE\Wow6432Node\SAP\gsskrb5 /v ForceIniCredOK /t REG_DWORD /d 1 /f`
- `REG ADD HKLM\SOFTWARE\SAP\gsskrb5 /v ForceIniCredOK /t REG_DWORD /d 1 /f`

# Add a new SAP BW Application Server data source to the Power BI service, or edit an existing one

1. In the data source configuration window, enter the SAP BW Application Server's **Hostname**, **System Number**, and **client ID**, as you would to sign in to your SAP BW server from Power BI Desktop.
  2. In the **SNC Partner Name** field, enter *p:<SPN you mapped to your SAP BW service user>*. For example, if the SPN is SAP/BWServiceUser@MYDOMAIN.COM, enter *p:SAP/BWServiceUser@MYDOMAIN.COM* in the **SNC Partner Name** field.
  3. For the SNC Library, select the **Custom** option and provide the absolute path for GX64KRB5.DLL or GSSKRB5.DLL on the gateway machine.
  4. Select **Use SSO via Kerberos for DirectQuery queries**, and then select **Apply**. If the test connection is not successful, verify that the previous setup and configuration steps were completed correctly.
5. [Run a Power BI report](#)

## Troubleshooting

### Troubleshoot gx64krb5 configuration

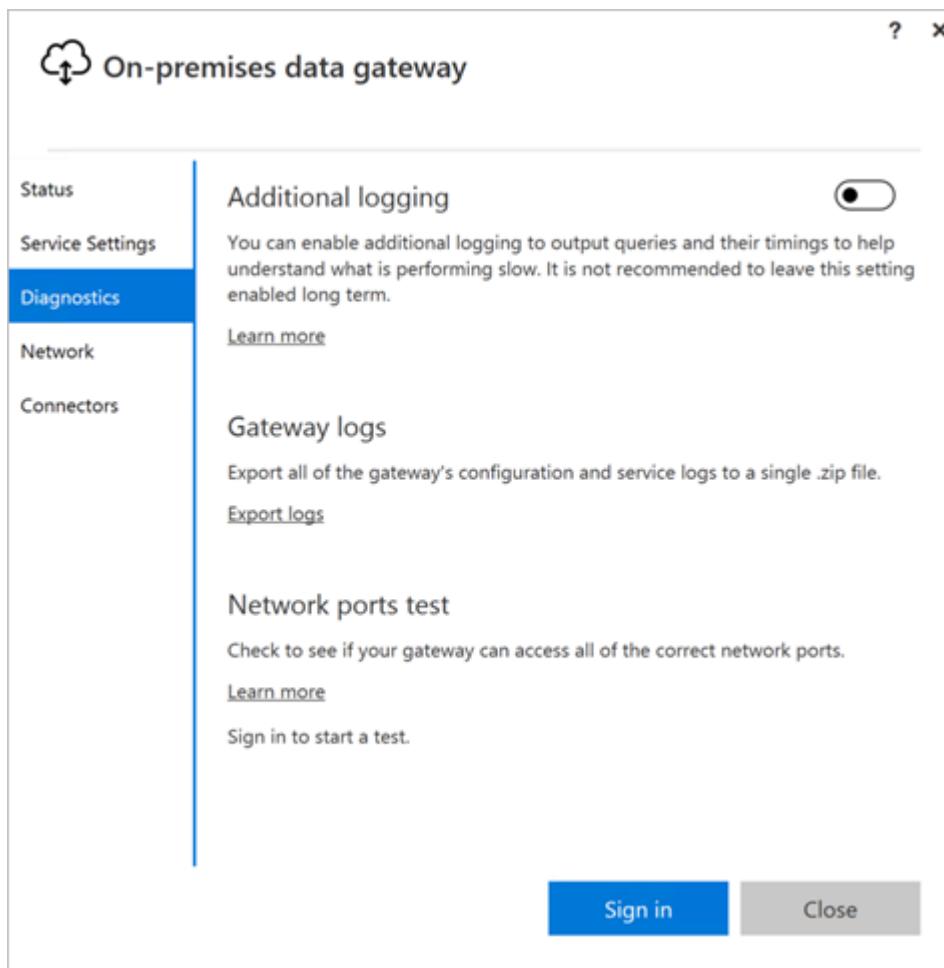
If you encounter any of the following problems, follow these steps to troubleshoot the gx64krb5 installation and SSO connections:

- You encounter errors when you complete the gx64krb5 setup steps. For example, the SAP BW server won't start after you've changed the profile parameters. View the server logs (...work\dev\_w0 on the server machine) to troubleshoot these errors.
- You can't start the SAP BW service due to a sign-on failure. You might have provided the wrong password when you set the SAP BW *start-as* user. Verify the password by signing in as the SAP BW service user on a machine in your Active Directory environment.
- You get errors about underlying data source credentials (for example, SQL Server), which prevent the server from starting, verify that you've granted the service user access to the SAP BW database.

- You get the following message: *(GSS-API) specified target is unknown or unreachable*. This error usually means you have the wrong SNC name specified. Make sure to use *p:* only, not *p:CN=*, to precede the service user's UPN in the client application.
- You get the following message: *(GSS-API) An invalid name was supplied*. Make sure *p:* is the value of the server's SNC identity profile parameter.
- You get the following message: *(SNC error) the specified module could not be found*. This error is often caused by placing *gx64krb5.dll* in a location that requires elevated privileges (./administrator rights) to access.

## Troubleshoot gateway connectivity issues

1. Check the gateway logs. Open the gateway configuration application, and select **Diagnostics**, then **Export logs**. The most recent errors are at the end of any log files you examine.



2. Turn on SAP BW tracing, and review the generated log files. There are several different types of SAP BW tracing available (for example, CPIC tracing):

- a. To enable CPIC tracing, set two environment variables: **CPIC\_TRACE** and **CPIC\_TRACE\_DIR**.

The first variable sets the trace level and the second variable sets the trace file directory. The directory must be a location that members of the Authenticated Users group can write to.

- b. Set **CPIC\_TRACE** to 3 and **CPIC\_TRACE\_DIR** to whichever directory you want the trace files written to. For example:

<b>CPIC_TRACE</b>	3
<b>CPIC_TRACE_DIR</b>	C:\traces\cpic

- c. Reproduce the issue and ensure that **CPIC\_TRACE\_DIR** contains trace files.

- d. Examine the contents of the trace files to determine the blocking issue. For example, you may find that gx64krb5.dll was not loaded properly, or that an Active Directory user different than the one you were expecting initiated the SSO connection attempt.

## Next steps

For more information about the on-premises data gateway and DirectQuery, see the following resources:

- [What is an on-premises data gateway?](#)
- [DirectQuery in Power BI](#)
- [Data sources supported by DirectQuery](#)
- [DirectQuery and SAP BW](#)
- [DirectQuery and SAP HANA](#)

# Use Kerberos for SSO to Teradata

Article • 02/23/2023

This article describes a specific added requirement to successfully enable single sign-on (SSO) to Teradata from the Power BI service.

- If Teradata identifies user accounts by using *sAMAccountNames*, you must set `FullDomainResolutionEnabled` on the gateway to `True`.
- If Teradata identifies user accounts by using *User Principal Names (UPNs)*, keep `FullDomainResolutionEnabled` on the gateway set to `False`.

## Enable SSO for Teradata

To change the `FullDomainResolutionEnabled` configuration on the gateway to enable SSO for Teradata:

1. In the on-premises gateway directory at `%ProgramFiles%\On-premises data gateway`, open the configuration file `Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config`.
2. In the file, find the `FullDomainResolutionEnabled` property and change its value to `True`.

XML

```
<setting name=" FullDomainResolutionEnabled " serializeAs="String">
 <value>True</value>
</setting>
```

## Next steps

For more information about the on-premises data gateway and DirectQuery, see the following resources:

- [What is an on-premises data gateway?](#)
- [DirectQuery in Power BI](#)
- [Data sources supported by DirectQuery](#)
- [DirectQuery and SAP Business Warehouse \(BW\)](#)
- [DirectQuery and SAP HANA](#)

# Use Security Assertion Markup Language for SSO from Power BI to on-premises data sources

Article • 05/17/2022

By enabling single sign-on (SSO), you can make it easy for Power BI reports and dashboards to refresh data from on-premises sources while you respect user-level permissions that are configured on those sources. To enable seamless SSO connectivity, you use [Security Assertion Markup Language \(SAML\)](#).

## ⓘ Note

You can connect to only one data source using Single Sign-On SAML with an on-premises data gateway. To connect to an additional data source using Single Sign-On SAML, you must use a different on-premises data gateway.

## Supported data sources for SAML

Microsoft currently supports SAP HANA with SAML. For more information about setting up and configuring single sign-on for SAP HANA by using SAML, see [SAML SSO for BI Platform to HANA](#).

We support additional data sources with [Kerberos](#) (including SAP HANA).

For SAP HANA, we recommend that you enable encryption before you establish a SAML SSO connection. To enable encryption, configure the HANA server to accept encrypted connections, and then configure the gateway to use encryption to communicate with your HANA server. Because the HANA ODBC driver doesn't encrypt SAML assertions by default, the signed SAML assertion is sent from the gateway to the HANA server *in the clear* and is vulnerable to interception and reuse by third parties.

## ⓘ Important

Because [SAP no longer supports OpenSSL](#), Microsoft has also discontinued its support. Your existing connections continue to work but you can no longer create new connections. Use SAP Cryptographic Library (CommonCryptoLib), or `sapcrypto`, instead.

# Configure the gateway and data source

To use SAML, you must establish a trust relationship between the HANA servers for which you want to enable SSO and the gateway. In this scenario, the gateway serves as the SAML identity provider (IdP). You can establish this relationship in various ways. SAP recommends that you use CommonCryptoLib to complete the setup steps. For more information, see the official SAP documentation.

## Create the certificates

You can establish a trust relationship between a HANA server and the gateway IdP by signing the gateway IdP's X509 certificate with a root certificate authority (CA) that's trusted by the HANA server.

To create the certificates, do the following:

1. On the device that's running SAP HANA, create an empty folder to store your certificates, and then go to that folder.
2. Create the root certificates by running the following command:

```
openssl req -new -x509 -newkey rsa:2048 -days 3650 -sha256 -keyout
CA_Key.pem -out CA_Cert.pem -extensions v3_ca'''
```

Be sure to copy and save the passphrase to use this certificate to sign other certificates. You should see the *CA\_Cert.pem* and *CA\_Key.pem* files being created.

3. Create the IdP certificates by running the following command:

```
openssl req -newkey rsa:2048 -days 365 -sha256 -keyout IdP_Key.pem -out
IdP_Req.pem -nodes
```

You should see the *IdP\_Key.pem* and *IdP\_Req.pem* files being created.

4. Sign the IdP certificates with the root certificates:

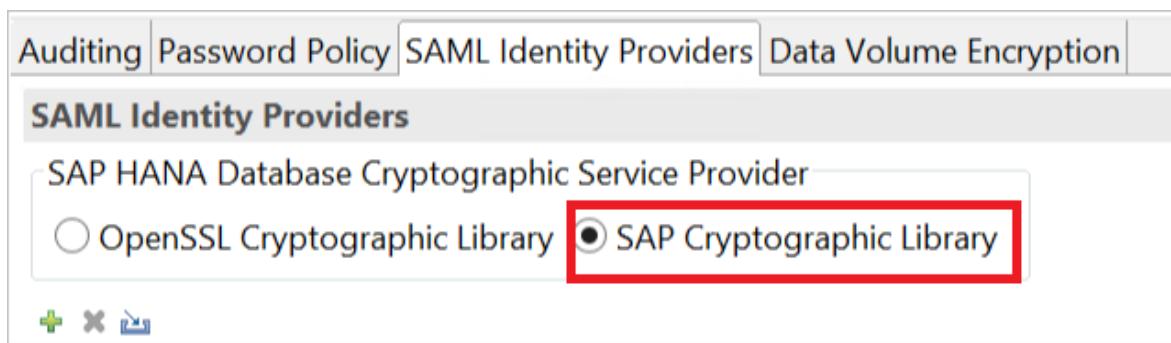
```
openssl x509 -req -days 365 -in IdP_Req.pem -sha256 -extensions
usr_cert -CA CA_Cert.pem -CAkey CA_Key.pem -CAcreateserial -out
IdP_Cert.pem
```

You should see the *CA\_Cert.srl* and *IdP\_Cert.pem* files being created. At this time, you're concerned only with the *IdP\_Cert.pem* file.

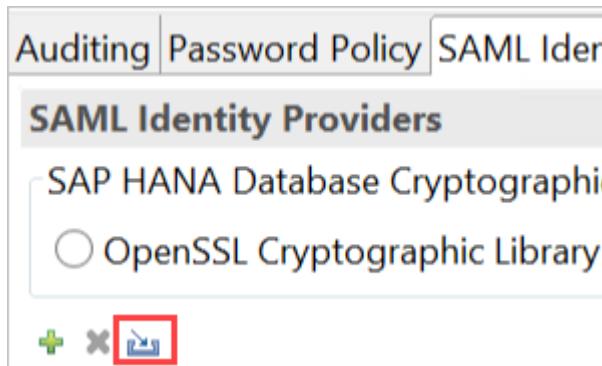
## Create mapping for the SAML identity provider certificate

To create mapping for the SAML Identity Provider certificate, do the following:

1. In SAP HANA Studio, right-click your SAP HANA server name, and then select **Security > Open Security Console > SAML Identity Provider**.
2. Select the **SAP Cryptographic Library** option. Do *not* use the OpenSSL Cryptographic Library option, which is deprecated by SAP.



3. To import the signed certificate *IdP\_Cert.pem*, select the blue **Import** button, as shown in the following image:



4. Remember to assign a name for your identity provider.

## Import and create the signed certificates in HANA

To import and create the signed certificates in HANA, do the following:

1. In SAP HANA Studio, run the following query:

```
CREATE CERTIFICATE FROM '<idp_cert_pem_certificate_content>'
```

Here's an example:

```
CREATE CERTIFICATE FROM
'-----BEGIN CERTIFICATE-----
MIIDyDCCArCgA...veryLongString...0WkC5deeawTyMje6
-----END CERTIFICATE-----
'
```

2. If there's no personal security environment (PSE) with purpose SAML, create one by running the following query in SAP HANA Studio:

```
CREATE PSE SAMLCOLLECTION;
set pse SAMLCOLLECTION purpose SAML;
```

3. Add the newly created signed certificate to the PSE by running the following command:

```
alter pse SAMLCOLLECTION add CERTIFICATE <certificate_id>;
```

For example:

```
alter pse SAMLCOLLECTION add CERTIFICATE 1978320;
```

You can check the list of created certificates by running the following query:

```
select * from PUBLIC"."CERTIFICATES"
```

The certificate is now properly installed. To confirm the installation, you can run the following query:

```
select * from "PUBLIC"."PSE_CERTIFICATES"
```

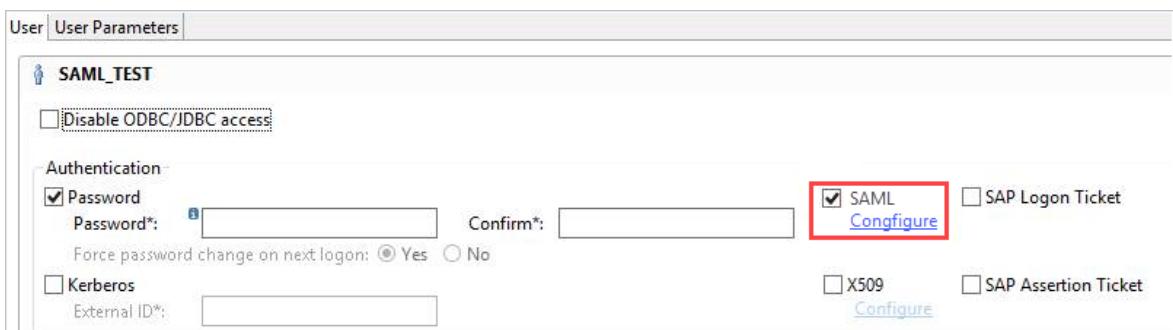
## Map the user

To map the user, do the following:

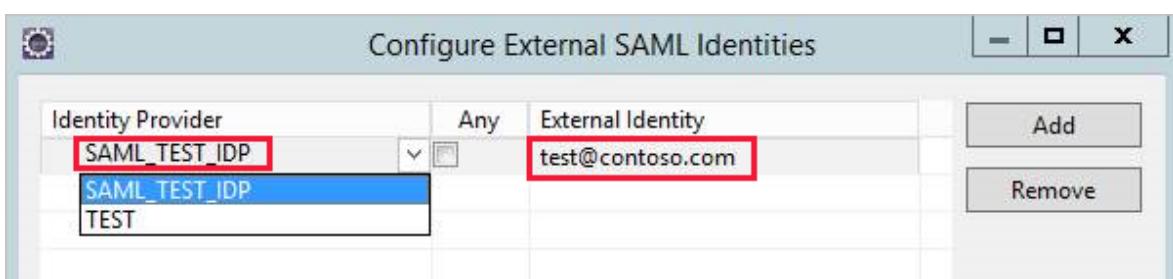
1. In SAP HANA Studio, select the **Security** folder.



2. Expand **Users**, and then select the user that you want to map your Power BI user to.
3. Select the **SAML** checkbox, and then select **Configure**, as shown in the following image:



4. Select the identity provider that you created in the [Create mapping for the SAML identity provider certificate](#) section. For **External Identity**, enter the Power BI user's UPN (ordinarily, the email address the user uses to sign in to Power BI), and then select **Add**.



If you've configured your gateway to use the *ADUserNameReplacementProperty* configuration option, enter the value that will replace the Power BI user's original UPN. For example, if you set *ADUserNameReplacementProperty* to *SAMAccountName*, enter the user's *SAMAccountName*.

# Configure the gateway

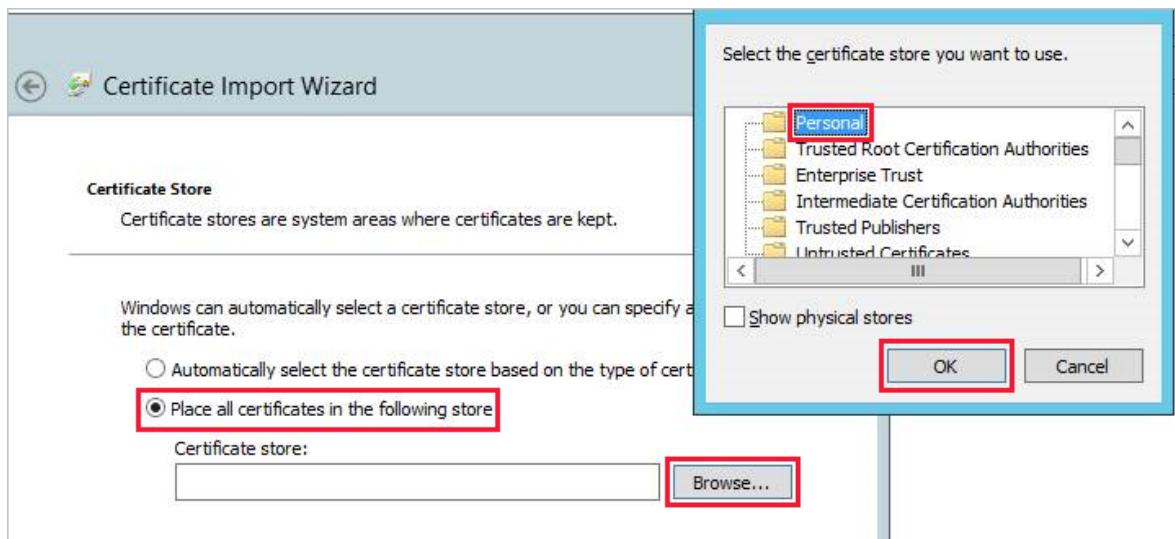
Now that you've configured the gateway certificate and identity, convert the certificate to a PFX file format, and then configure the gateway to use the certificate by doing the following:

1. Convert the certificate to PFX format by running the following command. This command names the resulting file *samlcert.pfx* and sets *root* as its password, as shown here:

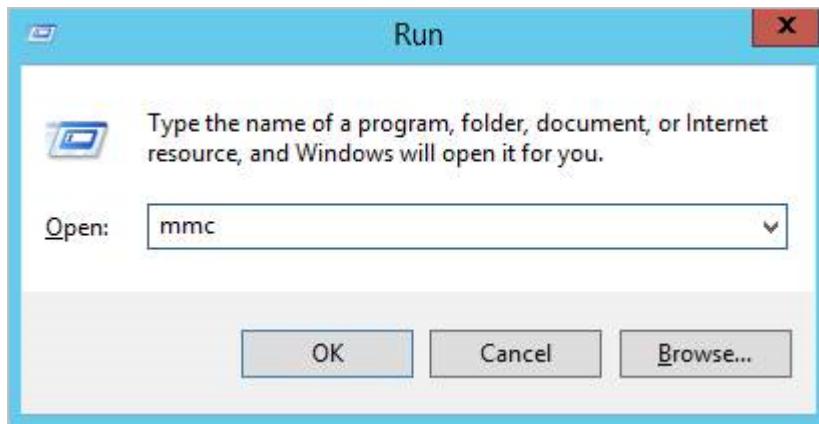
```
openssl pkcs12 -export -out samltest.pfx -in IdP_Cert.pem -inkey
IdP_Key.pem -passin pass:root -passout pass:root
```

2. Copy the PFX file to the gateway machine:

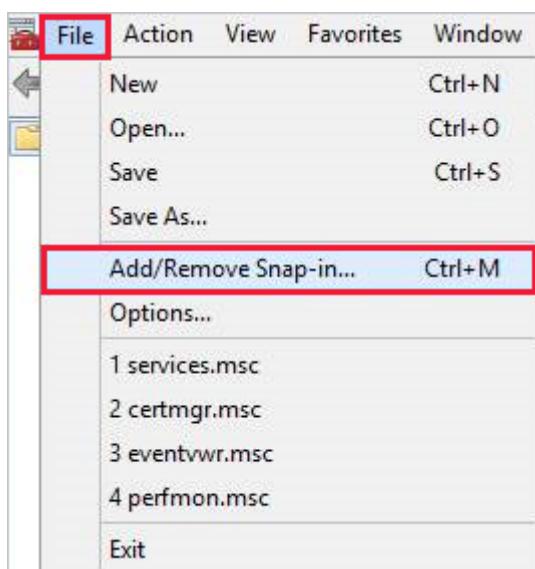
- a. Double-click *samltest.pfx*, and then select **Local Machine > Next**.
- b. Enter the password, and then select **Next**.
- c. Select **Place all certificates in the following store**, and then select **Browse > Personal > OK**.



- d. Select **Next**, and then select **Finish**.
3. To grant the gateway service account access to the private key of the certificate, do the following:
    - a. On the gateway machine, run Microsoft Management Console (MMC).



b. In MMC, select File > Add/Remove Snap-in.

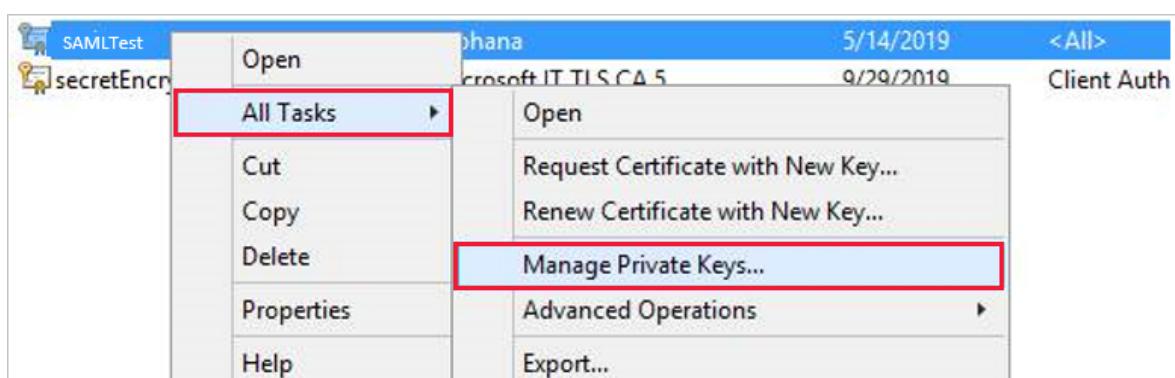


c. Select Certificates > Add, and then select Computer account > Next.

d. Select Local Computer > Finish > OK.

e. Expand Certificates > Personal > Certificates, and then look for the certificate.

f. Right-click the certificate, and then select All Tasks > Manage Private Keys.



g. Add the gateway service account to the list. By default, the account is NT SERVICE\PBIEgwService. You can find out which account is running the gateway

service by running `services.msc` and then looking for **On-premises data gateway service**.

Offline Files	The Offline ...	Disabled	Local System
On-premises data gateway service	The on-pre... Helps the c...	Running Automatic	NT SERVICE\PowerBI_EgwService
Optimize drives		Manual	Local System

Finally, add the certificate thumbprint to the gateway configuration:

1. To list the certificates on your machine, run the following PowerShell command:

```
PowerShell
```

```
Get-ChildItem -path cert:\LocalMachine\My
```

2. Copy the thumbprint for the certificate you created.
3. Go to the gateway directory, which is `C:\Program Files\On-premises data gateway` by default.
4. Open `PowerBI.DataMovement.Pipeline.GatewayCore.dll.config`, and then look for the `SapHanaSAMLCertThumbprint` section. Paste the thumbprint you copied in step 2.
5. Restart the gateway service.

## Run a Power BI report

Now you can use the **Manage Gateway** page in Power BI to configure the SAP HANA data source. Under **Advanced Settings**, enable SSO via SAML. By doing so, you can publish reports and datasets binding to that data source.

### Single sign-on

- Use SSO via Kerberos for DirectQuery queries (i)
- Use SSO via Kerberos for DirectQuery and Import queries (i)
- Use SSO via SAML for DirectQuery queries

## Note

SSO uses Windows Authentication so make sure the windows account can access the gateway machine. If not sure, make sure to add NT-AUTHORITY\Authenticated Users (S-1-5-11) to the local machine “Users” group.

# Troubleshoot using SAML for single sign-on to SAP HANA

This section provides extensive steps to troubleshoot using SAML for single sign-on to SAP HANA. Using these steps can help you self-diagnose and correct any issues you might face.

## Rejected credentials

After you configure SAML-based SSO, you might see the following error in the Power BI portal: "The credentials provided cannot be used for the SapHana source." This error indicates that the SAML credentials were rejected by SAP HANA.

Server-side authentication traces provide detailed information for troubleshooting credential issues on SAP HANA. To configure tracing for your SAP HANA server, do the following:

1. On the SAP HANA server, turn on the authentication trace by running the following query:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') set
('trace', 'authentication') = 'debug' with reconfigure
```

2. Reproduce the issue.
3. In SAP HANA Studio, open the administration console, and then select the **Diagnosis Files** tab.
4. Open the most recent index server trace, and then search for *SAMLAuthenticator.cpp*.

You should find a detailed error message that indicates the root cause, as shown in the following example:

```
[3957]{-1}[-1/-1] 2018-09-11 21:40:23.815797 d Authentication
SAMLAuthenticator.cpp(00091) : Element
'{urn:oasis:names:tc:SAML:2.0:assertion}Assertion', attribute 'ID':
'123123123123123' is not a valid value of the atomic type 'xs:ID'.
[3957]{-1}[-1/-1] 2018-09-11 21:40:23.815914 i Authentication
SAMLAuthenticator.cpp(00403) : No valid SAML Assertion or SAML Protocol
detected
```

5. After you've finished troubleshooting, turn off the authentication trace by running the following query:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') UNSET
('trace', 'authentication');
```

## Verify and troubleshoot gateway errors

To follow the procedures in this section, you need to [collect gateway logs](#).

### SSL error (certificate)

#### Error symptoms

This issue has multiple symptoms. When you try to add a new data source, you might see an error message like the following:

```
Unable to connect: We encountered an error while trying to connect to . Details:
"We could not register this data source for any gateway instances within this
cluster. Please find more details below about specific errors for each gateway
instance."
```

When you try to create or refresh a report, you might see an error message like the one in the following image:

## Cannot load model

X

Couldn't load the model schema associated with this report. Make sure you have a connection to the server, and try again.

Please try again later or contact support. If you contact support, please provide these details.

**Underlying error code** -2147467259

**Underlying error message** [SAP AG][LIBODBCHDB DLL][HDBODBC] Communication link failure;-10709 Connection failed (RTE:[300012] Cannot create SSL engine: The certificate chain was issued by an authority that is not trusted. (hana2-s4-sso2.westus2.cloudapp.azure.com:30044))

**Correlation ID** 0762cf29-50e1-86a4-acbb-a72329cabd2a

**DM\_ErrorDetailNameCode\_UnderlyingHRESULT** -2147467259

**Microsoft.Data.Mashup.CredentialError.DataSourceKind** SapHana

**Microsoft.Data.Mashup.CredentialError.DataSourceOriginKind** SapHana

**Microsoft.Data.Mashup.CredentialError.DataSourceOriginPath** hana2-s4-sso2.westus2.cloudapp.azure.com:30044

**Microsoft.Data.Mashup.CredentialError.DataSourcePath** hana2-s4-sso2.westus2.cloudapp.azure.com:30044

**Microsoft.Data.Mashup.CredentialError.Reason** EncryptedConnectionFailed

**Microsoft.Data.Mashup.MashupSecurityException.DataSources** [{"kind":"SapHana","path":"hana2-s4-sso2.westus2.cloudapp.azure.com:30044"}]

**Microsoft.Data.Mashup.MashupSecurityException.Reason** EncryptedConnectionFailed

**Activity ID** ceaeb724-1085-4ac3-9d47-5423cce98235

**Request ID** 66401121-824d-6ccf-6c59-6e9384a264d9

**Time** Wed Sep 30 2020 14:52:26 GMT-0700 (Pacific Daylight Time)

**Service version** 13.0.14487.41

**Client version** 2010.1.03085-train

**Cluster URI** https://df-msit-scus-redirect.analysis.windows.net/

[Get help](#) [Copy](#)

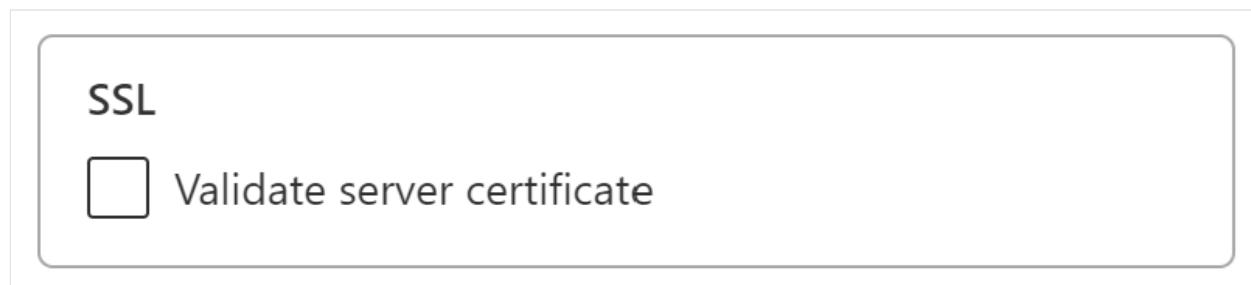
Close

When you investigate the Mashup[date]\*.log, you'll see the following error message:

```
A connection was successfully established with the server, but then an error
occurred during the login process and the certificate chain was issued by an
authority that is not trusted
```

## Resolution

To resolve this SSL error, go to the data source connection and then, in the **Validate Server Certificate** dropdown list, select **No**, as shown in the following image:



After you've selected this setting, the error message will no longer appear.

## Gateway SignXML error

The gateway SignXML error can be the result of incorrect *SapHanaSAMLCertThumbprint* settings, or it can be an issue with the HANA server. Entries in the gateway logs help identify where the issue resides, and how to resolve it.

## Error symptoms

Log entries for `SignXML: Found the cert...`: If your `GatewayInfo[date].log` file contains this error, the SignXML cert was found, and your troubleshooting efforts should focus on steps that are found in the "["Verify and troubleshoot the HANA server side"](#)" section.

Log entries for `Couldn't find saml cert`: If your `GatewayInfo[date].log` file contains this error, *SapHanaSAMLCertThumbprint* is set incorrectly. The following resolution section describes how to resolve the issue.

## Resolution

To properly set *SapHanaSAMLCertThumbprint*, follow the instructions in the "["Configure the gateway"](#)" section. The instructions begin with *Finally, add the certificate thumbprint to the gateway configuration.*

After you've changed the configuration file, you need to restart the gateway service for the change to take effect.

## Validation

When *SapHanaSAMLCertThumbprint* is properly set, your gateway logs will have entries that include `SignXML: Found the cert....`. At this point, you should be able to proceed to the "["Verify and troubleshoot the HANA server side"](#)" section.

If the gateway is unable to use the certificate to sign the SAML assertion, you might see an error in the logs that's similar to the following:

```
GatewayPipelineErrorCode=DM_GWPipeline_UnknownError GatewayVersion=
InnerType=CryptographicException InnerMessage=<pi>Signing key is not loaded.</pi>
InnerToString=<pi>System.Security.Cryptography.CryptographicException: Signing key
is not loaded.
```

To resolve this error, follow the instructions beginning with step 3 in the "["Configure the gateway"](#)" section.

After you've changed the configuration, restart the gateway service for the change to take effect.

## [Verify and troubleshoot the HANA server side](#)

Use the solutions in this section if the gateway can find the certificate and sign the SAML assertion but you're still experiencing errors. You'll need to collect HANA authentication traces, as described earlier in the "[Rejected credentials](#)" section.

## The SAML identity provider

The presence of the `Found SAML provider` string in the HANA authentication traces indicates that the SAML identity provider is configured properly. If the string is not present, the configuration is incorrect.

## Resolution

First, determine whether your organization is using OpenSSL or commoncrypto as the `sslcryptoprocessor`. To determine which provider is being used, do the following:

1. Open SAP HANA Studio.
2. Open the Administration Console for the tenant that you're using.
3. Select the **Configuration** tab, and use `sslcryptoprocessor` as a filter, as shown in the following image:

The screenshot shows the SAP HANA Studio Administration Console interface. The title bar says "PQO2@H23 (SYSTEM) PQO2". The top navigation bar includes "Overview", "Landscape", "Alerts", "Performance", "Volumes", "Configuration" (which is highlighted), "System Information", "Diagnosis Files", and "Trace Configuration". A "Last Update" timestamp is shown as "15-Sep-2020 4:07:02 PM". A "Filter" field contains "sslcryptoprocessor". The main table has columns "Name", "Default", "System", "Database", and "Host". Under "Name", there are entries for "global.ini" and "communication". Under "communication", there is an entry for "sslcryptoprocessor" with the value "commoncrypto".

Next, verify that the cryptographic library is set correctly by doing the following:

1. Go to Security Console in SAP HANA Studio by selecting the **SAML Identity Providers** tab, and do either of the following:
  - If the `sslcryptoprocessor` is OpenSSL, select **OpenSSL Cryptographic Library**.
  - If the `sslcryptoprocessor` is commonCrypto, select **SAP Cryptographic Library**.

In the following image, **SAP Cryptographic Library** is selected:

The screenshot shows the SAP HANA Studio Security Console interface. The title bar says "Security PQO1 (SYSTEM)". The top navigation bar includes "Auditing", "Password Policy", "SAML Identity Providers" (which is highlighted), and "Data Volume Encryption". Below the navigation bar, it says "SAML Identity Providers". There is a list item "SAP HANA Database Cryptographic Service Provider" with two radio button options: "OpenSSL Cryptographic Library" and "SAP Cryptographic Library". The "SAP Cryptographic Library" option is selected and highlighted with a red box. The main table below has columns "Identity Provider Name", "Issued To", "Issued By", and "Entity ID". One row is visible for "GATEWAYTEAM".

2. Deploy your changes by selecting the **Deploy** button at the upper right, as shown in the following image:



## Validation

When the traces are properly configured, they'll report `Found SAML provider` and will *not* report `SAML Provider not found`. You can proceed to the next section, "[Troubleshoot the SAML assertion signature](#)."

If the cryptographic provider is set but `SAML Provider not found` is still being reported, search for a string in the trace that begins with the following text:

```
Search SAML provider for certificate with subject =
```

In that string, ensure that the subject and issuer are exactly the same as displayed in the SAML identity provider tab in Security Console. A difference of even a single character can cause the problem. If you find a difference, you can fix the issue in the SAP Cryptographic Library so that the entries match exactly.

If changing the SAP Cryptographic Library doesn't fix the issue, you can manually edit the *Issued To* and *Issued By* fields simply by double-clicking them.

## Troubleshoot the SAML assertion signature

You might find HANA authentication traces that contain entries similar to the following:

```
[48163]{-1}[-1/-1] 2020-09-11 21:15:18.896165 i Authentication
SAMLAuthenticator.cpp(00398) : Unable to verify XML signature [48163]{-1}[-1/-1]
2020-09-11 21:15:18.896168 i Authentication MethodSAML.cpp(00103) : unsuccessful
login attempt with SAML ticket!
```

The presence of such entries means that the signature isn't trusted.

## Resolution

If you're using **OpenSSL** as your `sslcryptoprovider`, check to see whether the `trust.pem` and `key.pem` files are in the SSL directory. For more information, see the SAP blog

## Securing the communication between SAP HANA Studio and SAP HANA Server through SSL ↗.

If you're using **commoncrypto** as your `sslcryptopprovider`, check to see whether there's a collection with your certificate in the tenant.

### Validation

When the traces are properly configured, they'll report `Found valid XML signature`.

## Troubleshoot the UPN mapping

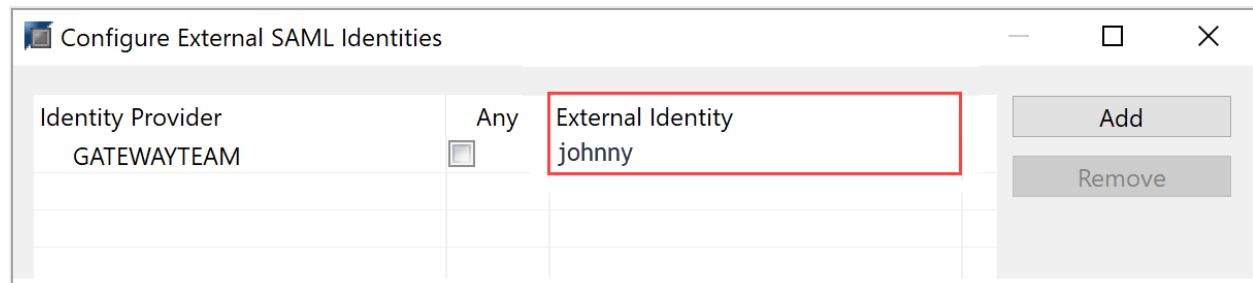
You might find HANA traces that contain entries similar to the following:

```
SAMLAuthenticator.cpp(00886) : Assertion Subject NameID: `johnny@contoso.com`
SAMLAuthenticator.cpp(00398) : Database user does not exist
```

The error indicates that nameld `johnny@contoso.com` is found in the SAML assertions, but it doesn't exist or isn't mapped correctly in HANA Server.

### Resolution

Go to the HANA database user and, under the selected SAML checkbox, select the **Configure** link. The following window appears:



As the error message describes, HANA was trying to find `johnny@contoso.com`, but the external identity is displayed only as `johnny`. These two values must match. To resolve the issue, under **External Identity**, change the value to `johnny@contoso.com`. Note that this value is case sensitive.

## Next steps

For more information about the on-premises data gateway and DirectQuery, see the following resources:

- [What is an on-premises data gateway?](#)
- [DirectQuery in Power BI](#)

- Data sources supported by DirectQuery
- DirectQuery and SAP Business Warehouse (BW)
- DirectQuery and SAP HANA

# Test single sign-on (SSO) configuration

Article • 08/22/2023

*Single sign-on (SSO)* enables each Power BI user to access the precise data they have permissions for in an underlying data source. Many Power BI data sources are enabled for SSO, using either [Kerberos](#) constrained delegation or Security Assertion Markup Language ([SAML](#)). For more information, see [Overview of single sign-on for on-premises data gateways in Power BI](#).

Setting up SSO is complex, so you can use the *test single sign-on (SSO) configuration* feature to test your configuration.

The single sign-on test:

- Lets the gateway connect to the data source by using a test User Principal Name (UPN) that you provide.
- Validates the SSO setup, which includes checking UPN mapping to a local Active Directory (AD) identity for impersonation and data source access.
- Helps identify problems if connection failures occur. For example, an error message indicates if a UPN maps to a local AD identity that doesn't have access to the data source.

The test single sign-on feature works for both Kerberos and SAML-based SSO for the data sources listed in [Supported data sources for SSO](#). For Kerberos constrained delegation, the test single sign-on feature can help test SSO for both DirectQuery and Import, or only DirectQuery data sources.

 **Important**

The test single sign-on feature requires the March 2021 gateway release or later.

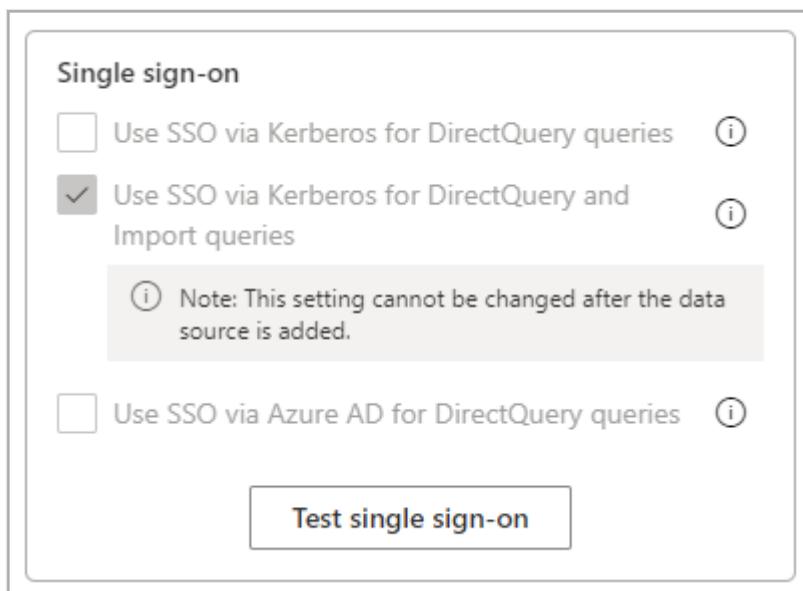
## Test SSO for the gateway

To test the SSO configuration:

1. From **Manage connections and gateways** in Power BI, select **Settings** for the data source.

Data (preview)					
Connections	On-premises data gateways	Virtual network data gateways			
Cloud and data gateway connections for artifacts. <a href="#">Learn more about supported connections.</a>					
Name ↑	Connection type	Users	Status	Gateway cluster name	
AdventureWorks ... new connection	SQL Server  Settings Manage users Remove	All Users All Users All Users	@(2) @(2) @(2)	powerbi-docs	
NewSQL				powerbi-docs	

2. In the **Settings** pane, under **Single sign-on**, select **Test single sign-on**.



3. Provide a User Principal Name to test.



If the gateway cluster is able to impersonate the user and successfully connect to the data source, the test succeeds, as shown in the following image:

### User Principal Name (UPN)

miquella@contoso.com

Start test

vmkone



Test Passed

### Additional Information

#### Activity Id

afc60af1-2b29-4374-b792-8eba7300de1d

#### Request Id

14668592-7d81-4ec1-9b44-06072990069f

#### Date and time

5/4/2022, 11:45:43 AM

## Troubleshooting

This section describes common errors you might see when testing single sign-on, and actions you can take to fix them.

### Impersonation error

If the gateway cluster can't impersonate the user and connect to the data source, the test fails with the error message: **Error: The on-premises data gateway's service account failed to impersonate the user.**

## User Principal Name (UPN)

miquella@contoso.com

**Start test**

**vmkone**



Test Failed

### Error message

The on-premises data gateway's service account failed to impersonate the user.

### Error code

DM\_GWPipeline\_Gateway\_ImpersonationError

### Local Active Directory UPN

miquella@contoso.com

See the gateway logs for more detailed error information.

## Additional Information

### Activity Id

afc60af1-2b29-4374-b792-8eba7300de1d

### Request Id

9f10d670-401d-41d9-974d-cfb6e0fe8c7b

### Date and time

5/4/2022, 11:53:01 AM

There can be the following possible causes and solutions:

- The user doesn't exist in Azure Active Directory (Azure AD). Check if the user is present in Azure AD.
- The user isn't mapped correctly to a local AD account. Check configurations and follow the steps in [Overview of single sign-on for on-premises data gateways in Power BI](#).
- The gateway doesn't have impersonation rights. Grant the gateway service account local policy rights on the gateway machine as described in [Grant the gateway service account local policy rights on the gateway machine](#).

## Invalid credentials error

The error **Error: Invalid connection credentials** appears when the gateway can't connect to the data source, because the provided UPN doesn't have access to the data source.

### User Principal Name (UPN)

miquella@contoso.com

**Start test**

**vmkone**



Test Failed

#### Error message

We could not access the data source. Please make sure you have permission to access the data source and that your credentials are correct.

#### Error code

DM\_GWPipeline\_Gateway\_DataSourceAccessError

#### Local Active Directory UPN

Alpha

See the gateway logs for more detailed error information.

### Additional Information

#### Activity Id

afc60af1-2b29-4374-b792-8eba7300de1d

#### Request Id

ec84b0f0-c045-48a3-b9a3-84a4b508f95d

#### Date and time

5/4/2022, 12:12:20 PM

Check whether the data source has been misconfigured to deny access to the user. You may need to work with your data source/database administrator to access the data source's configuration and settings.

## Next steps

- [Overview of single sign-on \(SSO\) for gateways in Power BI](#)
- [Single sign-on \(SSO\) - Kerberos](#)
- [Single sign-on \(SSO\) - SAML](#)

# Azure Active Directory SSO

Article • 08/25/2023

Azure Active Directory (Azure AD) SSO enables single sign-on to the data gateway to access cloud data sources that rely on Azure AD based authentication. When you configure Azure AD SSO on the on-premises data gateway for an applicable data source, queries run under the Azure AD identity of the user that interacts with the Power BI report.

While Azure Virtual Networks (VNets) offer network isolation and security for your resources on the Microsoft cloud, you now require a secure way to connect to these data sources. On-premises data gateways help you achieve that. Azure AD SSO as explained previously allows users to see only data that they have access to.

## ⓘ Note

VNet data gateways, which are available in public preview for Power BI Premium Datasets, eliminate the need to install an on-premises data gateway for connecting to your VNet data sources. To learn more about VNet gateways and their current limitations, see [What is a virtual network \(VNet\) data gateway](#).

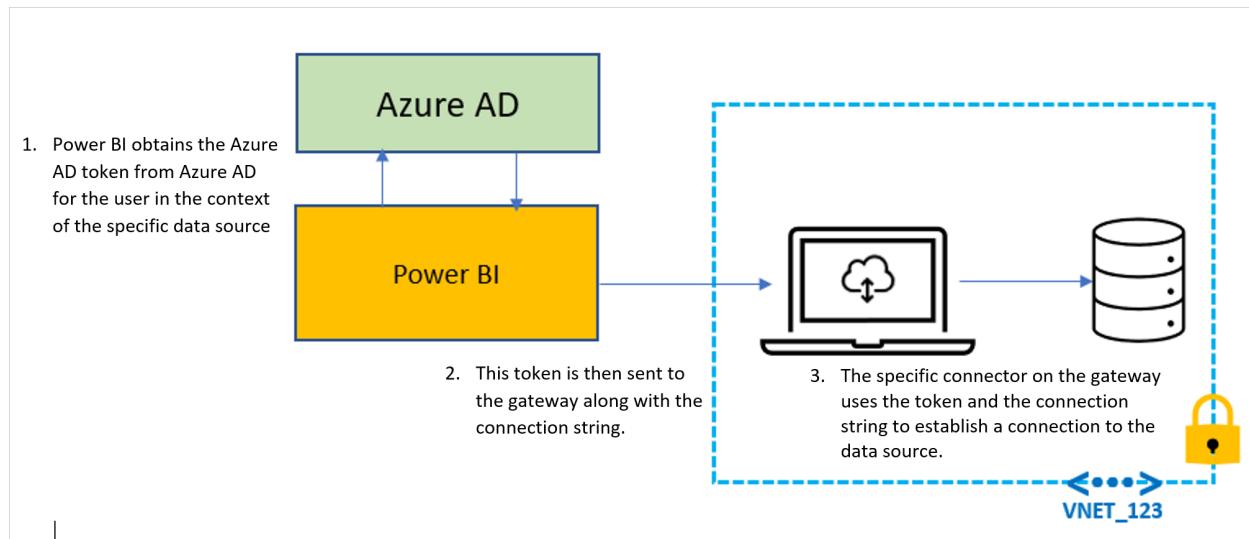
The following data sources aren't supported with Azure AD SSO using an on-premises data gateway behind an Azure VNet:

- Analysis Services
- ADLS Gen1
- ADLS Gen2
- Azure Blobs
- CDPA
- Exchange
- OData
- SharePoint
- SQL Server
- Web
- AzureDevOpsServer
- CDSTOData
- Cognite
- CommonDataService
- Databricks
- EQuIS

- Kusto (when using the newer “DataExplorer” function)
- VSTS
- Workplace Analytics

For more information on SSO, and a list of supported data sources for Azure AD SSO, see [Overview of single sign-on for on-premises data gateways in Power BI](#).

## Query steps when running Azure AD SSO



## Enable Azure AD SSO for Gateway

Since the Azure AD token of the user is passed via the gateway, it's possible for an admin of the gateway computer to obtain access to these tokens. To make sure a user with malicious intent isn't able to intercept these tokens, the following safeguard mechanisms are available:

- A tenant-level setting in the Power BI admin portal allows only Power BI service admins to enable this feature for a tenant. For more information, see [Azure AD single sign-on for gateways](#).
- As a Power BI service admin, you can also control who can install gateways in your tenant. For more information, see [Manage gateway installers](#).

The Azure AD SSO feature is disabled by default for on-premises data gateways. As a Power BI admin, you must enable the **Azure AD Single Sign-On (SSO) for Gateway** tenant setting in the Power BI Admin portal before data sources are enabled for Azure AD SSO on an on-premises data gateway.

△ Azure AD Single Sign-On (SSO) for Gateway

*Enabled for the entire organization*

Enable AAD SSO via the on-premises data gateway for applicable data sources. By enabling user access token information including name and email will be sent to these data sources for authentication via the on-premises data gateway. [Learn More](#)



 This setting applies to the entire organization

Apply

Cancel

## Next steps

- [Overview of single sign-on for on-premises data gateways in Power BI](#)

# Troubleshoot gateways - Power BI

Article • 08/31/2022

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

This article discusses some common issues when you use the on-premises data gateway with Power BI. If you encounter an issue that isn't listed here, you can use the Power BI [Community](#) site. Or, you can create a [support ticket](#).

## Configuration

### Error: Power BI service reported local gateway as unreachable. Restart the gateway and try again.

At the end of configuration, the Power BI service is called again to validate the gateway. The Power BI service doesn't report the gateway as live. Restarting the Windows service might allow the communication to be successful. To get more information, you can collect and review the logs as described in [Collect logs from the on-premises data gateway app](#).

## Data sources

## ⓘ Note

Not all data sources have dedicated articles detailing their connection settings or configuration. For many data sources and non-Microsoft connectors, connection options may vary between Power BI Desktop, and **Manage gateways > Data source settings** configurations in the Power BI service. In such cases, the default settings provided are the currently supported scenarios for Power BI.

## Error: Unable to Connect. Details: "Invalid connection credentials"

Within **Show details**, the error message that was received from the data source is displayed. For SQL Server, you see a message like the following:

Output

```
Login failed for user 'username'.
```

Verify that you have the correct username and password. Also, verify that those credentials can successfully connect to the data source. Make sure the account that's being used matches the authentication method.

## Error: Unable to Connect. Details: "Cannot connect to the database"

You were able to connect to the server but not to the database that was supplied. Verify the name of the database and that the user credential has the proper permission to access that database.

Within **Show details**, the error message that was received from the data source is displayed. For SQL Server, you see something like the following:

Output

```
Cannot open database "AdventureWorks" requested by the login. The login failed. Login failed for user 'username'.
```

## Error: Unable to Connect. Details: "Unknown error in data gateway"

This error might occur for different reasons. Be sure to validate that you can connect to the data source from the machine that hosts the gateway. This situation could be the result of the server not being accessible.

Within **Show details**, you can see an error code of `DM_GWPipeline_UnknownError`.

You can also look in **Event Logs > Applications and Services Logs > On-premises data gateway Service** for more information.

## Error: We encountered an error while trying to connect to <server>. Details: "We reached the data gateway, but the gateway can't access the on-premises data source."

You were unable to connect to the specified data source. Be sure to validate the information provided for that data source.

Within **Show details**, you can see an error code of **DM\_GWPipeline\_Gateway\_DataSourceAccessError**.

If the underlying error message is similar to the following, this means that the account you're using for the data source isn't a server admin for that Analysis Services instance. For more information, see [Grant server admin rights to an Analysis Services instance](#).

### Output

```
The 'CONTOSO\account' value of the 'EffectiveUserName' XML for Analysis property is not valid.
```

If the underlying error message is similar to the following, it could mean that the service account for Analysis Services might be missing the [token-groups-global-and-universal](#) (TGGAU) directory attribute.

### Output

```
The username or password is incorrect.
```

Domains with pre-Windows 2000 compatibility access have the TGGAU attribute enabled. Most newly created domains don't enable this attribute by default. For more information, see [Some applications and APIs require access to authorization information on account objects](#).

To confirm whether the attribute is enabled, follow these steps.

1. Connect to the Analysis Services machine within SQL Server Management Studio. Within the Advanced connection properties, include EffectiveUserName for the user in question and see if this addition reproduces the error.
2. You can use the dsacl Active Directory tool to validate whether the attribute is listed. This tool is found on a domain controller. You need to know what the distinguished domain name is for the account and pass that name to the tool.

### Console

```
dsacls "CN=John Doe,CN=UserAccounts,DC=contoso,DC=com"
```

You want to see something similar to the following in the results:

## Console

Allow BUILTIN\Windows Authorization Access Group  
SPECIAL ACCESS for  
tokenGroupsGlobalAndUniversal  
READ PROPERTY

To correct this issue, you must enable TGGAU on the account used for the Analysis Services Windows service.

Another possibility for "The username or password is incorrect."

This error could also be caused if the Analysis Services server is in a different domain than the users and there isn't a two-way trust established.

Work with your domain administrators to verify the trust relationship between domains.

**Unable to see the data gateway data sources in the Get Data experience for Analysis Services from the Power BI service**

Make sure that your account is listed in the **Users** tab of the data source within the gateway configuration. If you don't have access to the gateway, check with the administrator of the gateway and ask them to verify. Only accounts in the **Users** list can see the data source listed in the Analysis Services list.

Error: You don't have any gateway installed or configured for the data sources in this dataset.

Ensure that you've added one or more data sources to the gateway, as described in [Add a data source](#). If the gateway doesn't appear in the admin portal under **Manage gateways**, clear your browser cache or sign out of the service and then sign back in.

Error: Your data source can't be refreshed because the credentials are invalid. Please update your credentials and try again.

You were able to connect and refresh the dataset, with no runtime errors for the connection, yet in the Power BI service this error bar appears. When the user attempts to update the credentials with known-good credentials, an error appears stating that the credentials supplied were invalid.

This error can occur when the gateway attempts a test connection, even if the credentials supplied are acceptable and the refresh operation is successful. This occurs because when the gateway performs a connection test, it does not include any optional parameters during the connection attempt, and some data connectors (such as Snowflake, for example) require optional connection parameters in order to connect.

When your refresh is completing properly and you do not experience runtime errors, you can ignore these test connection errors for data sources that require optional parameters.

## Datasets

### **Error: There is not enough space for this row.**

This error occurs if you have a single row greater than 4 MB in size. Determine what the row is from your data source, and attempt to filter it out or reduce the size for that row.

### **Error: The server name provided doesn't match the server name on the SQL Server SSL certificate.**

This error can occur when the certificate common name is for the server's fully qualified domain name (FQDN), but you supplied only the NetBIOS name for the server. This situation causes a mismatch for the certificate. To resolve this issue, make the server name within the gateway data source and the PBIX file use the FQDN of the server.

### **Error: You don't see the on-premises data gateway present when you configure scheduled refresh.**

A few different scenarios could be responsible for this error:

- The server and database name don't match what was entered in Power BI Desktop and the data source configured for the gateway. These names must be the same. They aren't case sensitive.
- Your account isn't listed in the **Users** tab of the data source within the gateway configuration. You need to be added to that list by the administrator of the

gateway.

- Your Power BI Desktop file has multiple data sources within it, and not all of those data sources are configured with the gateway. You need to have each data source defined with the gateway for the gateway to show up within scheduled refresh.

## Error: The received uncompressed data on the gateway client has exceeded the limit.

The exact limitation is 10 GB of uncompressed data per table. If you're hitting this issue, there are good options to optimize and avoid it. In particular, reduce the use of highly constant, long string values and instead use a normalized key. Or, removing the column if it's not in use helps.

### Error:

## DM\_GWPipeline\_Gateway\_SpooledOperationMissing

A few different scenarios could be responsible for this error

- Gateway process may have restarted when the dataset refresh was in progress.
- The gateway machine is cloned where gateway is running. We should not clone gateway machine.

## Reports

## Error: Report could not access the data source because you do not have access to our data source via an on-premises data gateway.

This error is usually caused by one of the following:

- The data source information doesn't match what's in the underlying dataset. The server and database name need to match between the data source defined for the on-premises data gateway and what you supply within Power BI Desktop. If you use an IP address in Power BI Desktop, the data source for the on-premises data gateway needs to use an IP address as well.
- There's no data source available on any gateway within your organization. You can configure the data source on a new or existing on-premises data gateway.

## Error: Data source access error. Please contact the gateway administrator.

If this report makes use of a live Analysis Services connection, you could encounter an issue with a value being passed to EffectiveUserName that either isn't valid or doesn't have permissions on the Analysis Services machine. Typically, an authentication issue is due to the fact that the value being passed for EffectiveUserName doesn't match a local user principal name (UPN).

To confirm the effective username, follow these steps.

1. Find the effective username within the [gateway logs](#).
2. After you have the value being passed, validate that it's correct. If it's your user, you can use the following command from a command prompt to see the UPN. The UPN looks like an email address.

```
Console
```

```
whoami /upn
```

Optionally, you can see what Power BI gets from Azure Active Directory.

1. Browse to <https://developer.microsoft.com/graph/graph-explorer>.
2. Select **Sign in** in the upper-right corner.
3. Run the following query. You see a rather large JSON response.

```
HTTP
```

```
https://graph.windows.net/me?api-version=1.5
```

4. Look for **userPrincipalName**.

If your Azure Active Directory UPN doesn't match your local Active Directory UPN, you can use the [Map user names](#) feature to replace it with a valid value. Or, you can work with either your Power BI admin or local Active Directory admin to get your UPN changed.

## Kerberos

If the underlying database server and on-premises data gateway aren't appropriately configured for [Kerberos constrained delegation](#), enable [verbose logging](#) on the gateway. Then, investigate based on the errors or traces in the gateway's log files as a starting point for troubleshooting. To collect the gateway logs for viewing, see [Collect logs from the on-premises data gateway app](#).

## ImpersonationLevel

The ImpersonationLevel is related to the SPN setup or the local policy setting.

```
[DataMovement.PipeLine.GatewayDataAccess] About to impersonate user
DOMAIN\User (IsAuthenticated: True, ImpersonationLevel: Identification)
```

### Solution

Follow these steps to solve the issue.

1. Set up an SPN for the on-premises gateway.
2. Set up constrained delegation in your Active Directory.

## FailedToImpersonateUserException: Failed to create Windows identity for user userid

The FailedToImpersonateUserException happens if you're unable to impersonate on behalf of another user. This error could also happen if the account you're trying to impersonate is from another domain than the one the gateway service domain is on. This is a limitation.

### Solution

- Verify that the configuration is correct as per the steps in the previous "ImpersonationLevel" section.
- Ensure that the user ID it's trying to impersonate is a valid Active Directory account.

## General error: 1033 error while you parse the protocol

You get the 1033 error when your external ID that's configured in SAP HANA doesn't match the sign-in if the user is impersonated by using the UPN (alias@domain.com). In

the logs, you see "Original UPN 'alias@domain.com' replaced with a new UPN 'alias@domain.com'" at the top of the error logs, as seen here:

```
[DM.GatewayCore] SingleSignOn Required. Original UPN 'alias@domain.com'
replaced with new UPN 'alias@domain.com.'
```

## Solution

- SAP HANA requires the impersonated user to use the sAMAccountName attribute in Active Directory (user alias). If this attribute isn't correct, you see the 1033 error.

The screenshot shows the Active Directory Attribute Editor window. The tabs at the top are General, Address, Account, Profile, Telephone, Delegation, Remote control, Remote Desktop Services Profile, COM+, and Attribute Editor. The Attribute Editor tab is highlighted with a red box. Below it, the 'Attributes:' section displays a table with columns 'Attribute' and 'Value'. The table contains the following entries:

Attribute	Value
revision	<not set>
rid	<not set>
roomNumber	<not set>
sAMAccountName	SAP
sAMAccountType	805306368 = ( NORMAL_USER_ACCOUNT )
scriptPath	<not set>
secretary	<not set>

- In the logs, you see the sAMAccountName (alias) and not the UPN, which is the alias followed by the domain (alias@doimain.com).

```
<setting name="ADUserNameReplacementProperty" serializeAs="String">
 <value>sAMAccount</value>
</setting>
<setting name="ADServerPath" serializeAs="String">
 <value />
</setting>
<setting name="CustomASDataSource" serializeAs="String">
 <value />
</setting>
<setting name="ADUserNameLookupProperty" serializeAs="String">
 <value>AADEmail</value>
```

XML

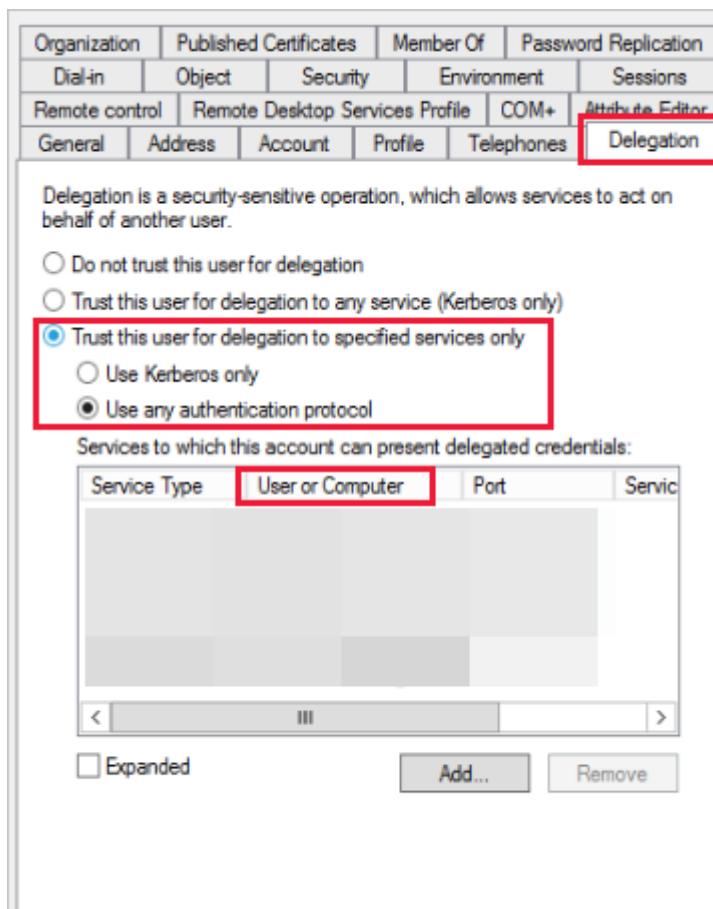
```
<setting name="ADUserNameReplacementProperty" serializeAs="String">
 <value>sAMAccount</value>
</setting>
<setting name="ADServerPath" serializeAs="String">
 <value />
</setting>
<setting name="CustomASDataSource" serializeAs="String">
 <value />
</setting>
<setting name="ADUserNameLookupProperty" serializeAs="String">
 <value>AADEmail</value>
```

# [SAP AG][LIBODBCHDB DLL][HDBODBC] Communication link failure:-10709 Connection failed (RTE:[-1] Kerberos error. Major: "Miscellaneous failure [851968]." Minor: "No credentials are available in the security package."

You get the "-10709 Connection failed" error message if your delegation isn't configured correctly in Active Directory.

## Solution

- Make sure that you have the SAP Hana server on the delegation tab in Active Directory for the gateway service account.



## Export logs for a support ticket

Gateway logs are required for troubleshooting and creating a support ticket. Use the following steps for extracting these logs.

1. Identify the gateway cluster.

If you're a dataset owner, first check the gateway cluster name associated with your dataset. In the following image, *IgniteGateway* is the gateway cluster.

The screenshot shows the 'Settings for AdventureWorksLocal - Multi Cred' page. At the top, it says 'This dataset has been configured by art@contoso.com.' and has a 'Refresh history' link. Below that is a section titled 'Gateway connection' with a note about using a data gateway. A toggle switch is set to 'On'. The main area displays a table of data sources:

Gateway	Department	Contact information	Status	Actions
IgniteGateway		art@contoso.com	Running on MINT587	

Below the table, a section titled 'Data sources included in this dataset:' lists two entries:

- SqlServer("server": "mint587\\sqlexpress", "database": "adventureworks2017") Maps to: SQLfordataflows (SSO)
- SqlServer("server": "mint587\\sqlexpress", "database": "powerbiest") Maps to: PowerBI Test

At the bottom are 'Apply' and 'Discard' buttons.

## 2. Check the gateway properties.

The gateway admin should then check the number of gateway members in the cluster and if load balancing is enabled.

If load balancing is enabled, then step 3 should be repeated for all gateway members. If it's not enabled, then exporting logs on the primary gateway is sufficient.

## 3. Retrieve and export the gateway logs.

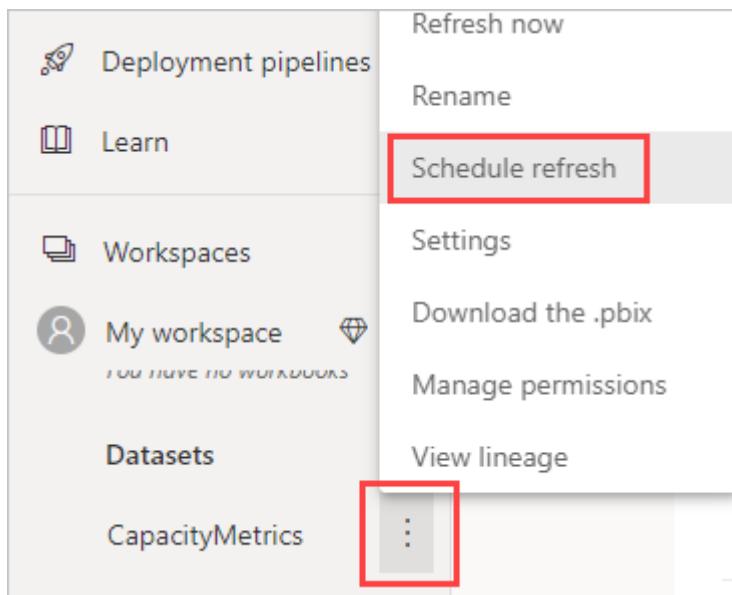
Next, the gateway admin, who is also the administrator of the gateway system, should do the following steps:

- Sign in to the gateway machine, and then launch the [on-premises data gateway app](#) to sign in to the gateway.
- Enable [additional logging](#).
- Optionally, you can [enable the performance monitoring features](#) and include performance logs to provide additional details for troubleshooting.
- Run the scenario for which you're trying to capture gateway logs.
- [Export the gateway logs](#).

# Refresh history

When you use the gateway for a scheduled refresh, Refresh history can help you see what errors occurred. It can also provide useful data if you need to create a support request. You can view scheduled and on-demand refreshes. The following steps show how you can get to the refresh history.

1. In the Power BI nav pane, in **Datasets**, select a dataset. Open the menu, and select **Schedule refresh**.



2. In **Settings for...**, select **Refresh history**.

A screenshot of the 'Settings for CapacityMetrics' page. At the top, there are tabs: General, Alerts, Subscriptions, Dashboards, Datasets (which is selected and highlighted in dark grey), Workbooks, and Reports. Below the tabs, it says 'CapacityMetrics'. There are two entries under 'Power BI Premium Capacity Metrics': 'Power BI Premium Capacity Metrics' and 'Power BI Premium Capacity Metrics'. To the right, there is a 'Settings for CapacityMetrics' section with a 'Refresh history' link, which is highlighted with a red box. Below that is a 'Gateway connection' link.

Refresh history					
Scheduled	OneDrive	Type	Start	End	Status
		On demand	7/5/2016, 5:30:12 PM	7/5/2016, 5:30:51 PM	Completed

For more information about troubleshooting refresh scenarios, see [Troubleshoot refresh scenarios](#).

# Fiddler trace

Fiddler [↗](#) is a free tool from Telerik that monitors HTTP traffic. You can see the back and forth with the Power BI service from the client machine. This traffic list might show errors and other related information.

8	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/dmm/gateways/discover
9	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/dmm/aggregateDataSource/147516?testConnection=true
11	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/content/packages/147029/refresh/
13	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/metadata/models/147516/?modelOptions=Default
14	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/refresh/subscribe
16	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/metadata/dashboard/95433/tiles
17	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/metadata/models/147516/?modelOptions=Default
18	-	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/refresh/subscribe

## Next steps

- Troubleshoot the on-premises data gateway
- Configure proxy settings for the on-premises data gateway
- Manage your data source - Analysis Services
- Manage your data source - SAP HANA
- Manage your data source - SQL Server
- Manage your data source - Import/scheduled refresh

More questions? Try the [Power BI Community ↗](#).

# Troubleshoot Power BI gateway (personal mode)

Article • 01/17/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

The following sections go through some common issues you might come across when you use the Power BI on-premises data gateway (personal mode).

## Update to the latest version

The current version of the gateway for personal use is the on-premises data gateway (personal mode). Update your installation to use that version.

Many issues can surface when the gateway version is out of date. It's a good general practice to make sure you're on the latest version. If you haven't updated the gateway for a month or longer, consider installing the latest version of the gateway. Then see if you can reproduce the issue.

## Installation

**Gateway (personal mode) operates on 64-bit versions:** If your computer is a 32-bit version, you can't install the gateway (personal mode). Your operating system has to be a 64-bit version. Install a 64-bit version of Windows or install the gateway (personal mode) on a 64-bit computer.

**Operation timed out:** This message is common if the computer, physical or virtual machine, on which you're installing the gateway (personal mode) has a single core processor. Close any applications, turn off any non-essential processes, and try installing again.

**Data management gateway or Analysis Services connector can't be installed on the same computer as gateway (personal mode):** If you already have an Analysis

If a services connector or a data management gateway is installed, you must first uninstall the connector or the gateway. Then, try installing the gateway (personal mode).

### Note

If you encounter an issue during installation, the setup logs can provide information to help you resolve the issue. For more information, see [Setup logs](#).

**Proxy configuration:** You might see issues with configuring the gateway (personal mode) if your environment needs the use of a proxy. To learn more about how to configure proxy information, see [Configure proxy settings for the on-premises data gateway](#).

## Schedule refresh

**Error:** The credential stored in the cloud is missing.

You might get this error in settings for a dataset if you have a scheduled refresh and then you uninstalled and reinstalled the gateway (personal mode). When you uninstall a gateway (personal mode), the data source credentials for a dataset that was configured for refresh are removed from the Power BI service.

**Solution:** In the Power BI service, go to the refresh settings for a dataset. In **Manage Data Sources**, for any data source with an error, select **Edit credentials**. Then sign in to the data source again.

**Error:** The credentials provided for the dataset are invalid. Please update the credentials through a refresh or in the Data Source Settings dialog to continue.

**Solution:** If you get a credentials message, it could mean:

- The usernames and passwords that you used to sign in to data sources aren't up to date. In the Power BI service, go to refresh settings for the dataset. In **Manage Data Sources**, select **Edit credentials** to update the credentials for the data source.
- Mashups between a cloud source and an on-premises source, in a single query, fail to refresh in the gateway (personal mode) if one of the sources is using OAuth for authentication. An example of this issue is a mashup between CRM Online and a local SQL Server instance. The mashup fails because CRM Online requires OAuth.

This error is a known issue, and it's being looked at. To work around the problem, have a separate query for the cloud source and the on-premises source. Then, use a merge or append query to combine them.

## Error: Unsupported data source.

**Solution:** If you get an unsupported data source message in **Schedule Refresh** settings, it could mean:

- The data source isn't currently supported for refresh in the Power BI service.
- The Excel workbook doesn't contain a data model, only worksheet data. The Power BI service currently only supports refresh if the uploaded Excel workbook contains a data model. When you import data by using Power Query in Excel, choose the **Load** option to load data to a data model. This option ensures that data is imported into a data model.

**Error:** [Unable to combine data] <query part>/<...>/<...> is accessing data sources that have privacy levels, which cannot be used together. Please rebuild this data combination.

**Solution:** This error is because of the privacy-level restrictions and the types of data sources you're using.

**Error:** Data source error: We cannot convert the value "[Table]" to type Table.

**Solution:** This error is because of the privacy-level restrictions and the types of data sources you're using.

**Error:** There is not enough space for this row.

**Solution:** This error occurs if you have a single row greater than 4 MB in size. Find the row from your data source, and attempt to filter it out or reduce the size for that row.

## Data sources

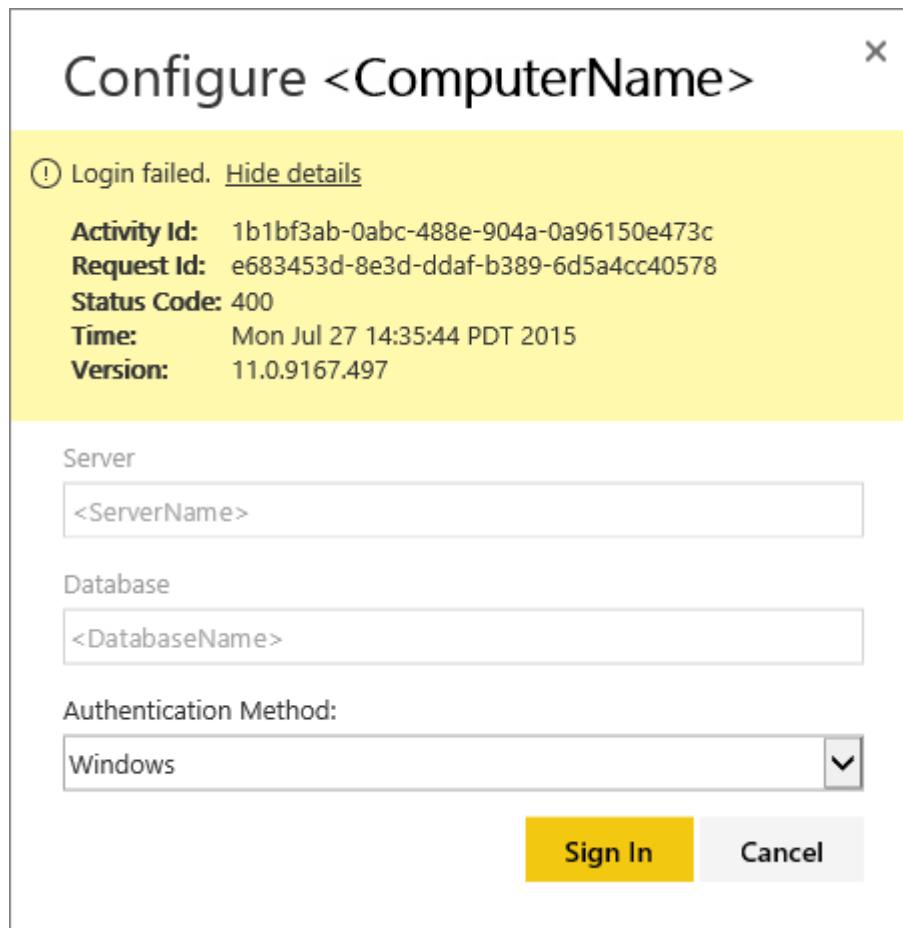
**Missing data provider:** The gateway (personal mode) operates on 64-bit versions only. It requires a 64-bit version of the data providers to be installed on the same computer where the gateway (personal mode) is installed. For example, if the data source in the dataset is Microsoft Access, you must install the 64-bit ACE provider on the same computer where you installed the gateway (personal mode).

### Note

If you have a 32-bit version of Excel, you can't install a 64-bit version ACE provider on the same computer.

**Windows authentication is not supported for Access database:** The Power BI service currently only supports Anonymous authentication for the Access database.

**Error: Sign-in error when you enter credentials for a data source:** If you get an error like this one when you enter Windows credentials for a data source:



You might still be on an older version of the gateway (personal mode).

**Solution:** For more information, see [Install the latest version of Power BI gateway \(personal mode\)](#).

**Error: Sign-in error when you select Windows authentication for a data source using ACE OLEDB:** If you get the following error when you enter data source credentials for a data source using an ACE OLEDB provider:

 Failed to update data source credentials: The credentials you provided for the data source are invalid. Please ensure the credentials you have provided for all the data sources are valid.

[Hide details](#)

Activity Id: 6edb9b25-c65b-4136-90c6-15e0be0d5f88  
Request Id: f270f4fa-6428-b3bb-8ae0-c06cea0bd2d3  
Status Code: 400  
Time: Thu Aug 20 2018 18:28:52 GMT-0700 (Pacific Daylight Time)  
Version: 11.0.9167.799  
Cluster URI: <https://<url>>  
Details: [Permission Error] Driver 'Microsoft.ACE.OLEDB.12.0' doesn't support Windows credentials.

The Power BI service doesn't currently support Windows authentication for a data source using an ACE OLEDB provider.

**Solution:** To work around this error, select **Anonymous authentication**. For a legacy ACE OLEDB provider, anonymous credentials are equal to Windows credentials.

## Tile refresh

If you receive an error when dashboard tiles refresh, see [Troubleshooting tile errors](#).

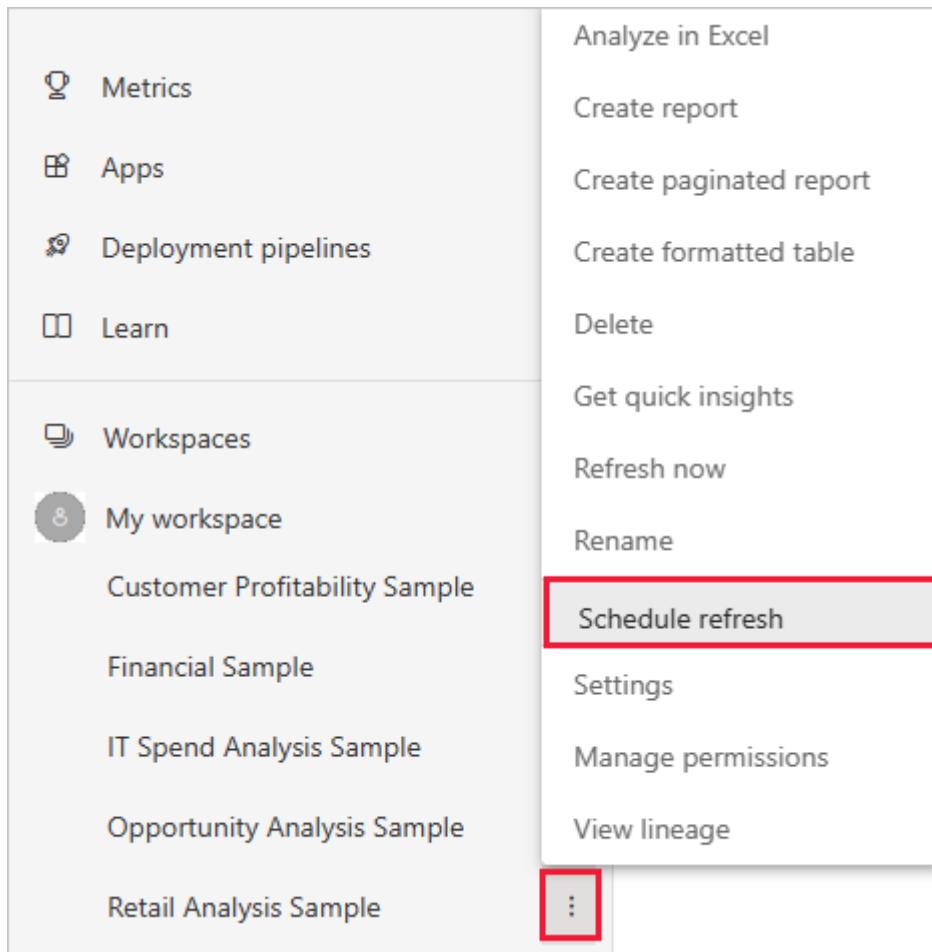
## Tools for troubleshooting

### Refresh history

With **Refresh history**, you can see what errors occurred and find useful data if you need to create a support request. You can view both scheduled and on-demand refreshes.

Here's how you get to **Refresh history**.

1. In the Power BI service navigation pane, in **Datasets**, select a dataset. Open the **More options (...)** menu, and select **Schedule refresh**.



2. In **Settings** for..., select **Refresh history**.

- [Customer Profitability Sample](#)
- [Financial Sample](#)
- [IT Spend Analysis Sample](#)
- [Opportunity Analysis Sample](#)
- [Retail Analysis Sample](#)
- [Sales & Returns](#)
- [Sales & Returns Sample v201912](#)
- [Sales and Marketing Sample PBIX current](#)

## Settings for Retail Analysis Sample

- [View dataset](#)
- [Refresh history](#)
- [Dataset description](#)
- [Sensitivity label](#)
- [Parameters](#)
- [Q&A](#)
- [Featured Q&A questions](#)
- [Endorsement](#)
- [Request access](#)
- [Dataset Image](#)
- [External sharing](#)

## Refresh history

[Scheduled](#) [OneDrive](#)

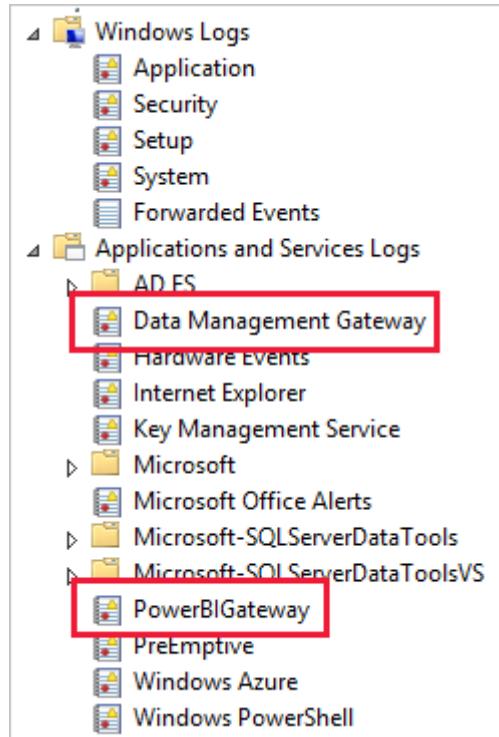
Details	Type	Start	End	Status	Message
	On demand	5/28/2019 9:32:23 AM	5/28/2019 9:32:44 AM	Completed	
<a href="#">Hide</a>	On demand	5/28/2019 9:17:03 AM	5/28/2019 9:17:14 AM	Failed	An error occurred while processing the data in the dataset.
<b>Data Source Error:</b> [Expression error] The name 'Table.TransformColumnNames' wasn't recognized. Make sure it's spelled correctly.					
<b>Cluster URI:</b> WABI-WEST-US-redirect.analysis.windows.net					
<b>Activity Id:</b> 0ec24881-ec30-470c-98a0-1109aa2036ae					
<b>Request Id:</b> 14aaef406-6d1e-32b0-0ad5-4370766897c3					
<b>Time:</b> 2019-05-28 14:17:14Z					
<a href="#">Show</a>	On demand	5/28/2019 9:17:04 AM	5/28/2019 9:17:14 AM	Failed	An error occurred while processing the data in the dataset.

[Close](#)

## Event logs

Several event logs can provide information. The first two, **Data Management Gateway** and **PowerBIGateway**, are present if you're an admin on the machine. If you're not an admin, and you're using the data gateway (personal mode), you'll see the log entries within the **Application** log.

The Data Management Gateway and PowerBIGateway logs are present under Application and Services Logs.



## Fiddler trace

Fiddler [↗](#) is a free tool from Telerik that monitors HTTP traffic. You can see the communication with the Power BI service from the client machine. This communication might show errors and other related information.

8	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/dmm/gateways/discover
{5} 9	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/dmm/aggregateDataSource/147516?testConnection=true
11	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/content/packages/147029/refresh/
{5} 13	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/metadata/models/147516/?modelOptions=Default
14	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/refresh/subscribe
16	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/metadata/dashboard/95433/tiles
{5} 17	200	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/metadata/models/147516/?modelOptions=Default
18	-	HTTPS	wabi-west-us-redirect.analysis.windows.net	/powerbi/refresh/subscribe

## Setup logs

If the gateway (personal mode) fails to install, you'll see a link to show the setup log. The setup log can show you details about the failure. These logs are Windows Install logs, also known as MSI logs. They can be fairly complex and hard to read. Typically, the resulting error is at the bottom, but determining the cause of the error isn't trivial. It could be a result of errors in a different log. It could also be a result of an error higher up in the log.



## Microsoft Power BI Personal Gateway Installation

Completed the installation wizard.

The Power BI Personal Gateway Installation Wizard has failed.

Please uninstall the Power BI Analysis Services Connector before continuing.

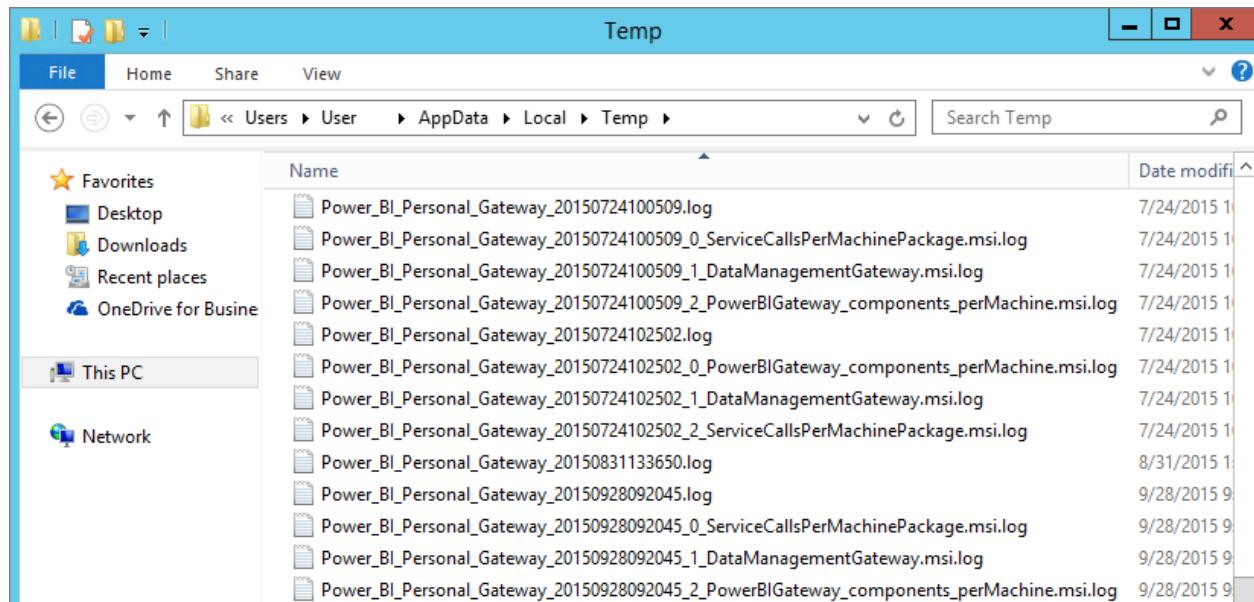
Please visit the [Personal Gateway Installation Troubleshooting Guide](#) for assistance.

For more information, see the [log file](#).

Or, you can go to your Temp folder (%temp%) and look for files that start with *Power\_BI\_*.

### ⓘ Note

Going to %temp% might take you to a subfolder of Temp. The *Power\_BI\_* files are in the root of the Temp directory. You might need to go up a level or two.



## Next steps

- Configure proxy settings for the on-premises data gateway
- Data refresh in Power BI
- Use personal gateways in Power BI

- Troubleshooting tile errors
- Troubleshoot gateways - Power BI

More questions? Try asking the [Power BI Community](#).

# Use custom data connectors with an on-premises data gateway

Article • 03/15/2023

## ⓘ Note

We've split the on-premises data gateway docs into **content that's specific to Power BI** and **general content that applies to all services** that the gateway supports. You're currently in the Power BI content. To provide feedback on this article, or the overall gateway docs experience, scroll to the bottom of the article.

You use Power BI data connectors to connect to and access data from an application, service, or data source. You can develop custom data connectors and use them in Power BI Desktop.

If you build reports in Power BI Desktop that use custom data connectors, you can use an on-premises data gateway to refresh those reports in the Power BI service.

To learn more about how to develop custom data connectors for Power BI, see the [DataConnectors SDK](#) in GitHub. This site includes information on how to get started, and samples for Power BI and Power Query.

## Enable and use custom connectors

To enable using custom connectors, select **Connectors** in the on-premises data gateway app. In **Custom data connectors**, under **Load custom data connectors from folder**, browse to and select a folder that the user running the gateway service can access. The default user is *NT SERVICE\PBIEgwService*. The gateway automatically loads the custom connector files in that folder, and they appear in the list of data connectors.

The screenshot shows the 'On-premises data gateway' settings interface. The left sidebar has a 'Connectors' tab selected, which is highlighted in blue. The main area displays 'Custom data connectors' with a 'Name' input field containing 'MyCustomConnector'. Below this, there's a 'Load custom data connectors from folder:' section with a text input field showing 'C:\Gateway\Custom Connectors' and a browse button ('...'). At the bottom right is a 'Close' button.

**Note**

If you're using an on-premises data gateway (personal mode), you can upload your Power BI report to the Power BI service and use the gateway to refresh it.

For an on-premises data gateway, you need to create a data source for your custom connector. On the gateway settings page in the Power BI service, select the option to enable using custom connectors with this cluster.

## Settings

X

powerbi-docs

Name \*

powerbi-docs

Department

Description

Contact information

admin@contoso.com

### General

- Distribute requests across all active gateways in this cluster. [Learn more.](#)

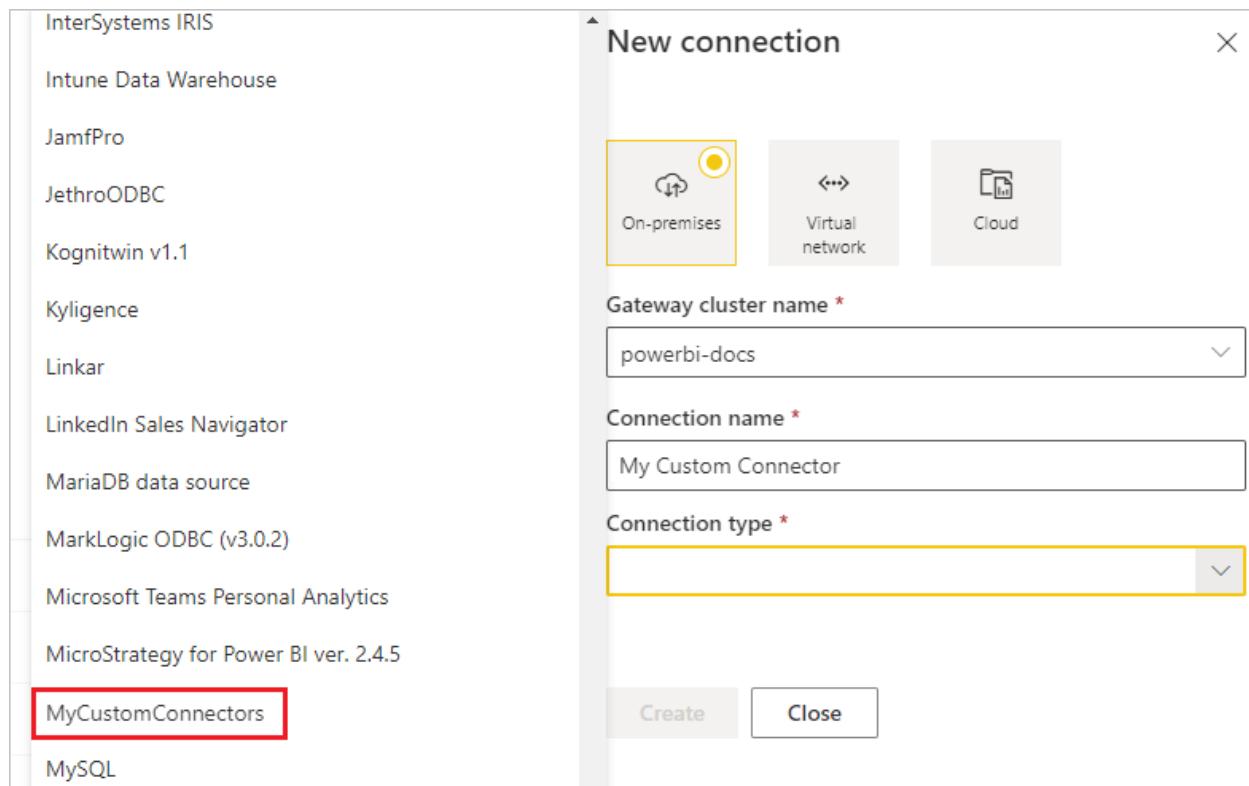
### Power BI

- Allow user's cloud data sources to refresh through this gateway cluster. [Learn more.](#)
- Allow user's custom data connectors to refresh through this gateway cluster. [Learn more.](#)

Save

Close

When you enable this option, you see your custom connectors as available data source connections that you can add to this gateway cluster. After you create a data source that uses your new custom connector, you can refresh Power BI reports by using the custom connector in the Power BI service.



## Considerations and limitations

- Make sure the folder you create is accessible to the background gateway service. Typically, folders under your users' Windows folders or system folders aren't accessible. The on-premises data gateway app shows a message if the folder isn't accessible. This limitation doesn't apply to the on-premises data gateway (personal mode).
- For custom connectors to work with the on-premises data gateway, they need to implement a `TestConnection` section in the custom connector code. This section isn't required when you use custom connectors with Power BI Desktop. For this reason, you can have a connector that works with Power BI Desktop, but not with the gateway. For more information on how to implement a `TestConnection` section, see [TestConnection](#).
- If your custom connector is on a network drive, include the fully qualified path in the on-premises data gateway app.
- You can only use one custom connector data source when working in DirectQuery mode. Multiple custom connector data sources don't work with DirectQuery.

## Next steps

- [Manage your data source - Analysis Services](#)

- Manage your data source - SAP HANA
- Manage your data source - SQL Server
- Manage your data source - Oracle
- Manage your data source - Import/scheduled refresh
- Configure proxy settings for the on-premises data gateway
- Use Kerberos for single sign-on (SSO) from Power BI to on-premises data sources

More questions? Try asking the [Power BI Community](#).

# Troubleshoot Power BI Desktop startup

Article • 06/27/2023

This article describes and provides remedies for several circumstances where Power BI can't open or can't connect to data sources.

## Issues with opening encrypted PBIX files

You can't open encrypted PBIX files by using a Power BI Desktop version that doesn't support information protection. If you need to continue using Power BI Desktop, update to a version that supports information protection.

**Solution:** [Select this link to directly download the latest Power BI Desktop installation executable ↗](#). The latest version of Power BI Desktop supports information protection and can decrypt and open any encrypted PBIX file.

## On-premises data gateway issues

Users who installed and are running earlier versions of the Power BI on-premises data gateway can be blocked from opening Power BI Desktop. Previous versions of the on-premises data gateway placed administrative policy restrictions on named pipes on the local machine.

**Solution:** To resolve the issue associated with the on-premises data gateway and enable Power BI Desktop to open, use one of the following options:

- Install the latest version of the Power BI on-premises data gateway.

The latest version of the Power BI on-premises data gateway doesn't place named pipe restrictions on the local machine, and allows Power BI Desktop to open properly. If you need to continue using the Power BI on-premises data gateway, the recommended resolution is to update it. [Select this link to directly download the latest Power BI on-premises data gateway installation executable ↗](#).

- Uninstall or stop the Power BI on-premises data gateway service. You can uninstall the Power BI on-premises data gateway if you no longer need it. Or you can stop the Power BI on-premises data gateway service, which removes the policy restriction and allows Power BI Desktop to open.
- Run Power BI Desktop with administrator privileges.

You can launch Power BI Desktop as an administrator, which also allows Power BI Desktop to successfully open. It's still recommended to install the latest version of the Power BI on-premises data gateway.

Power BI Desktop is a multiprocess architecture, and several of these processes communicate by using Windows named pipes. Other processes might interfere with those named pipes. The most common reason for such interference is security, including situations where antivirus software or firewalls block the pipes or redirect traffic to a specific port.

Opening Power BI Desktop with administrator privilege might resolve that issue. If you can't open Power BI Desktop with administrator privilege, ask your administrator which security rules are preventing named pipes from properly communicating. Then add Power BI Desktop and its subprocesses to the allowlists.

## Issues connecting to SQL Server

When you attempt to connect to a SQL Server database, you might see a message similar to the following error:

**An error happened while reading data from the provider:**

'Could not load file or assembly 'System.EnterpriseServices, Version=4.0.0.0, Culture=neutral, PublicKeyToken=xxxxxxxxxxxx' or one of its dependencies. Either a required impersonation level was not provided, or the provided impersonation level is invalid. (Exception from HRESULT: 0x80070542)'

**Solution:** You can often resolve the issue if you open Power BI Desktop as an administrator before you make the SQL Server connection. Opening Power BI Desktop as an administrator and establishing the connection registers the required DLLs. After that, you no longer have to open Power BI Desktop as an administrator. However, if you're connecting to SQL server with alternate Windows credentials, you have to open Power BI Desktop as an administrator every time you connect.

## "Unable to sign in" issues

You might see a message similar to the following error:

**Unable to sign in. Sorry, we encountered an error while trying to sign you in. Details: The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.**

**Solution:** Disable the certification revocation check at **Options and settings > Options > Security > Certification Revocation**. For details, see [Certificate revocation check, Power BI Desktop](#).

## Issues starting the Microsoft Store version of Power BI Desktop

You might see a message similar to the following error:

Hmmmm... can't reach this page. ms-pbi.pbi.microsoft.com's server IP address could not be found. Application event log message - The description for Event ID 1 from source

The message might include further information, such as the following details:

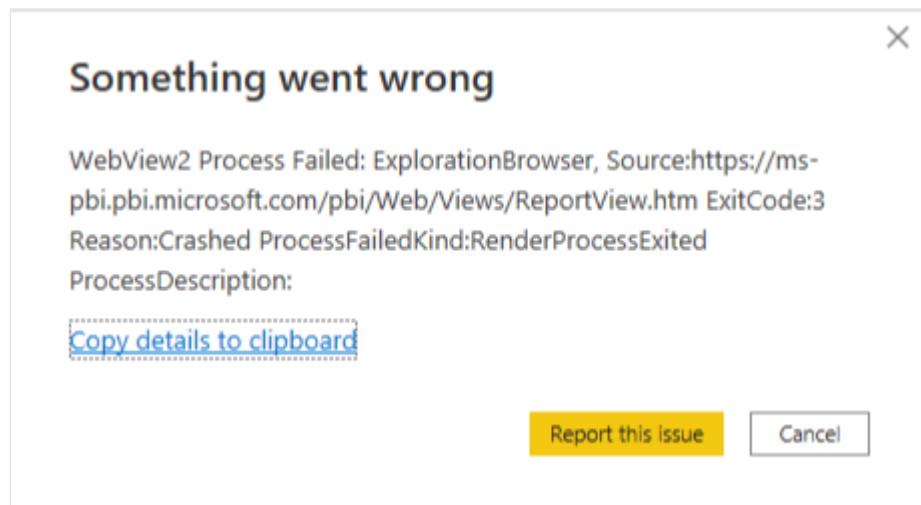
Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.

**Solution:** Reinstall WebView2 by using the following steps, which don't require elevated administrative permissions.

1. Uninstall webview2.
2. Reinstall webview2 by using this [installation link](#).

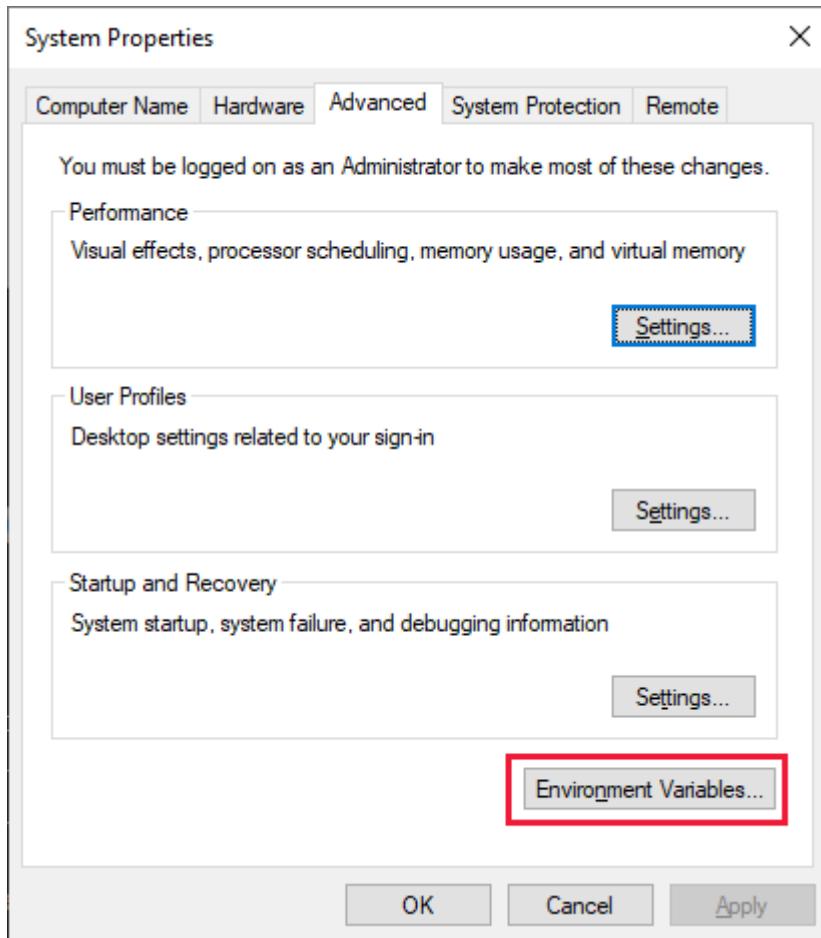
## Issues related to WebView2

Rarely, Power BI Desktop might fail to start and displays a gray window, or an error message that mentions **WebView2**.

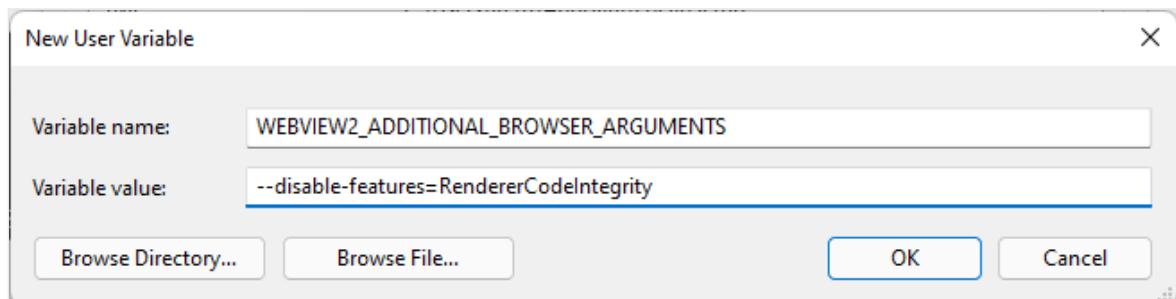


Most cases are caused by a program on your machine, usually antivirus software. To verify whether a program is causing the issue, take the following steps:

1. Close Power BI Desktop.
2. Open Windows **Settings** > **About** > **Advanced system settings** and select **Environment Variables**.



Select **New** under **User variables** and add variable name **WEBVIEW2\_ADDITIONAL\_BROWSER\_ARGUMENTS** with the value **--disable-features=RendererCodeIntegrity**.



3. Start Power BI Desktop and verify that it starts successfully this time.
4. Delete the environment variable you set.

**Solution:** If the preceding steps fixed the issue, disable any software that might be interfering with Power BI Desktop startup, or provide an exemption for the WebView2 process.

If you still have issues, submit a support incident to [Power BI support](#), and provide the following information:

- **WebView2 error reports.** If you use the Microsoft Store version of Power BI Desktop, the error reports are at `c:\Users\<username>\Microsoft\Power BI Desktop Store App\WebView2\EBWebView\Crashpad\reports` or `c:\Users\<username>\Microsoft\Power BI Desktop Store App\WebView2Elevated\EBWebView\Crashpad\reports`.

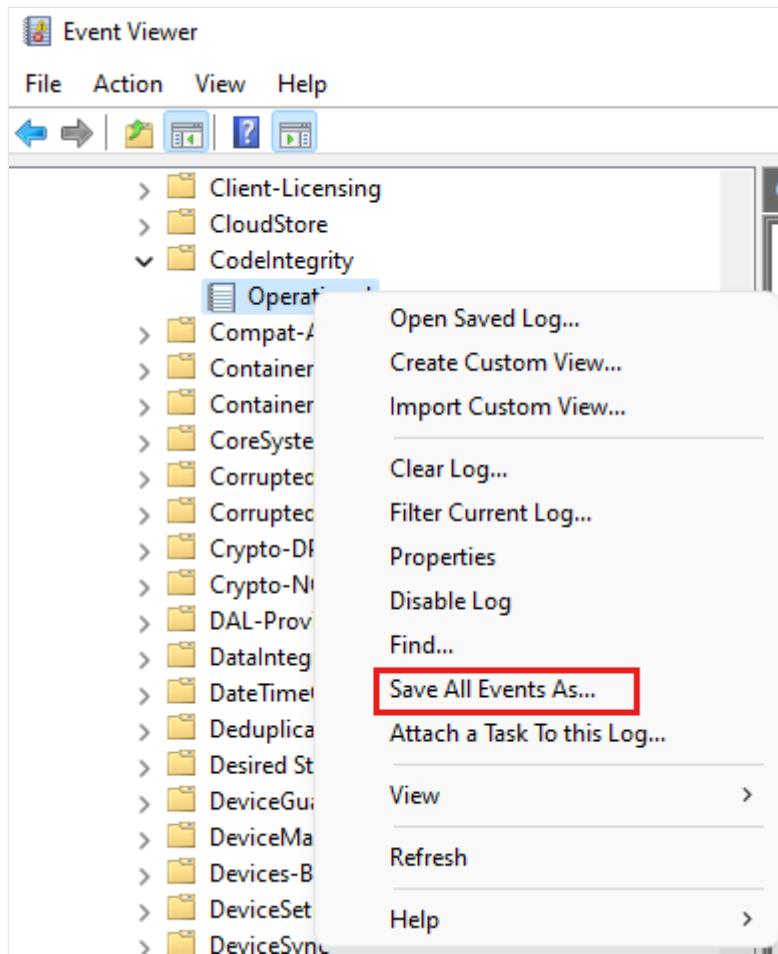
If you use the downloaded .exe version of Power BI Desktop, the error reports are at `c:\Users\<username>\AppData\Local\Microsoft\Power BI Desktop\WebView2\EBWebView\Crashpad\reports` or `c:\Users\<username>\AppData\Local\Microsoft\Power BI Desktop\WebView2Elevated\EBWebView\Crashpad\reports`.

- **Your machine's Device ID**, from Windows **Settings > System > About**.
- **Installer and update logs.** Collect the following files from the following locations by copying and pasting the paths into File Explorer and then copying the files to another location. Some files have the same names, so be sure not to overwrite them but instead rename them when copying.

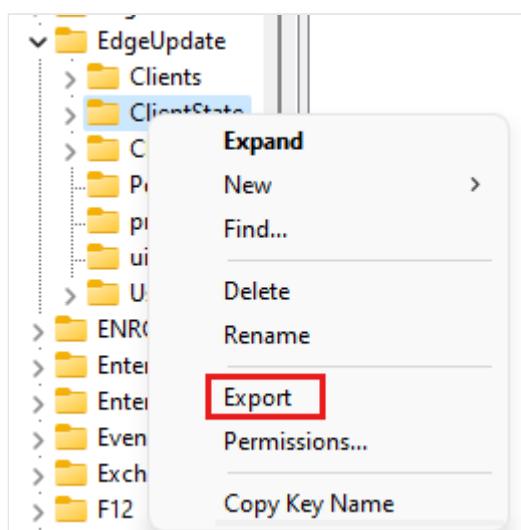
Path	File
<code>%temp%\</code>	<code>msedge_installer.log</code>
<code>%ProgramData%\Microsoft\EdgeUpdate\Log\</code>	<code>MicrosoftEdgeUpdate.log</code>
<code>%windir%\Temp\</code>	<code>MicrosoftEdgeUpdate.log</code>
<code>%allusersprofile%\Microsoft\EdgeUpdate\Log\</code>	<code>MicrosoftEdgeUpdate.log</code>
<code>%systemroot%\Temp\</code>	<code>msedge_installer.log</code>
<code>%localappdata%\Temp\</code>	<code>msedge_installer.log</code>
<code>%localappdata%\Temp\</code>	<code>MicrosoftEdgeUpdate.log</code>

- **Event Viewer logs.** Start Event Viewer from the **Start** menu. In Event Viewer, go to **Applications and Services log > Microsoft > Windows > CodeIntegrity > Operational**. Right-click **Operational** in the left pane and choose **Save All Events**

As. Store the file somewhere you can retrieve it. Do the same for Windows Logs > Application.



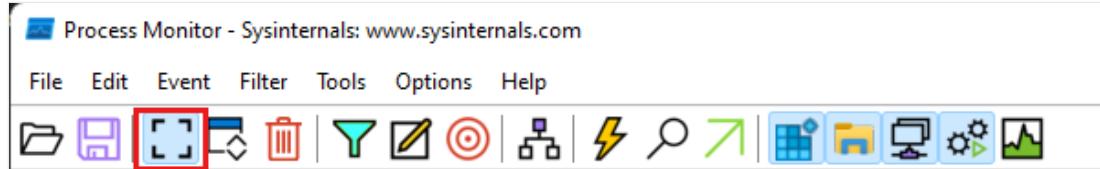
- The ClientState key from Registry Editor. Open Registry Editor by searching for `regedit` in Windows Search or the Start menu. In Registry Editor, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\ClientState`. Right-click ClientState in the left pane, choose Export, and save the file.



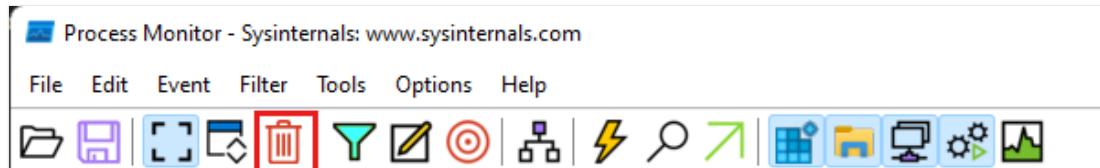
- Process traces. Follow these steps to collect process traces by using Process Monitor:

1. Download [Process Monitor](#), extract the downloaded file, and run *Procmon.exe*.

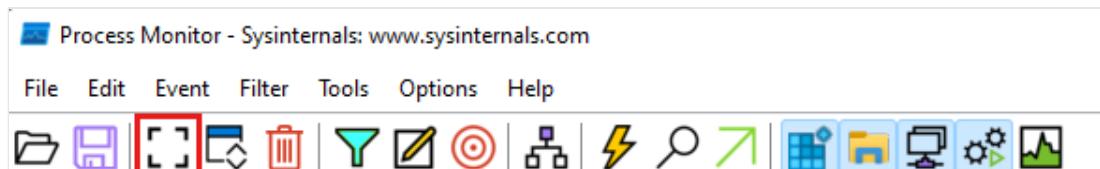
2. Stop capturing by selecting the open-square **Capture** button.



3. Clear all traces by selecting the **Clear garbage** can button.

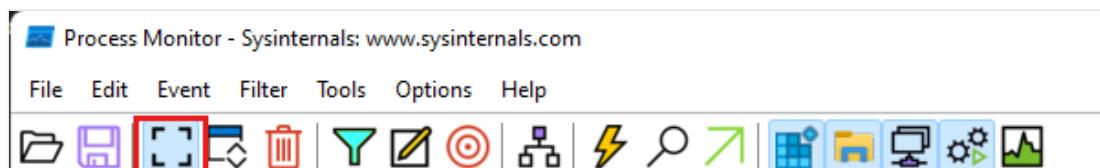


4. Start capturing by selecting the **Capture** button.

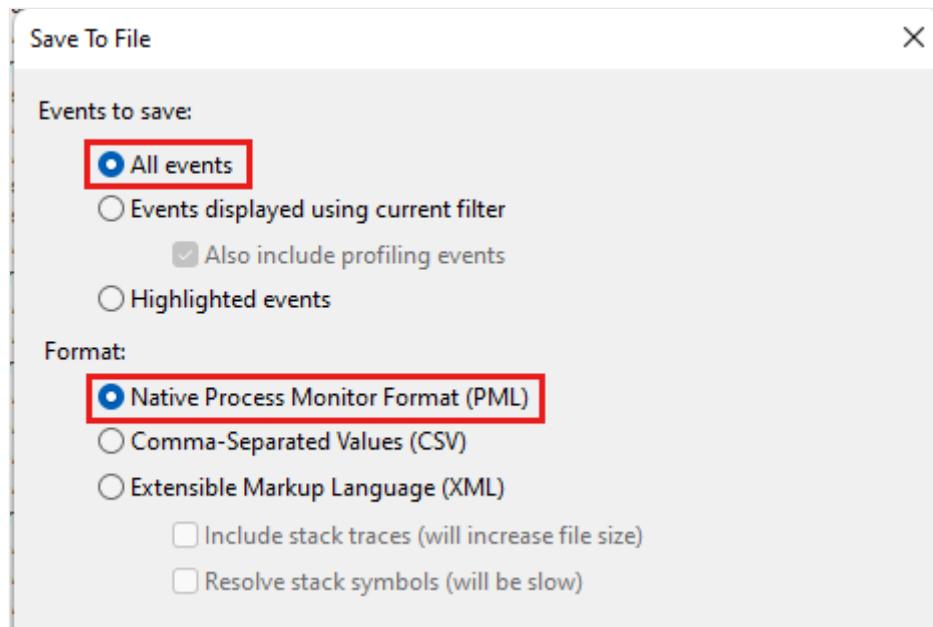


5. Launch Power BI Desktop and wait for the error to appear.

6. Stop the capture by selecting the **Capture** button.



7. Save the traces by choosing **File > Save**. In the **Save to File** dialog box, select **All events** and **Native Process Monitor Format (PML)**, provide a path for the file, and then select **OK**.

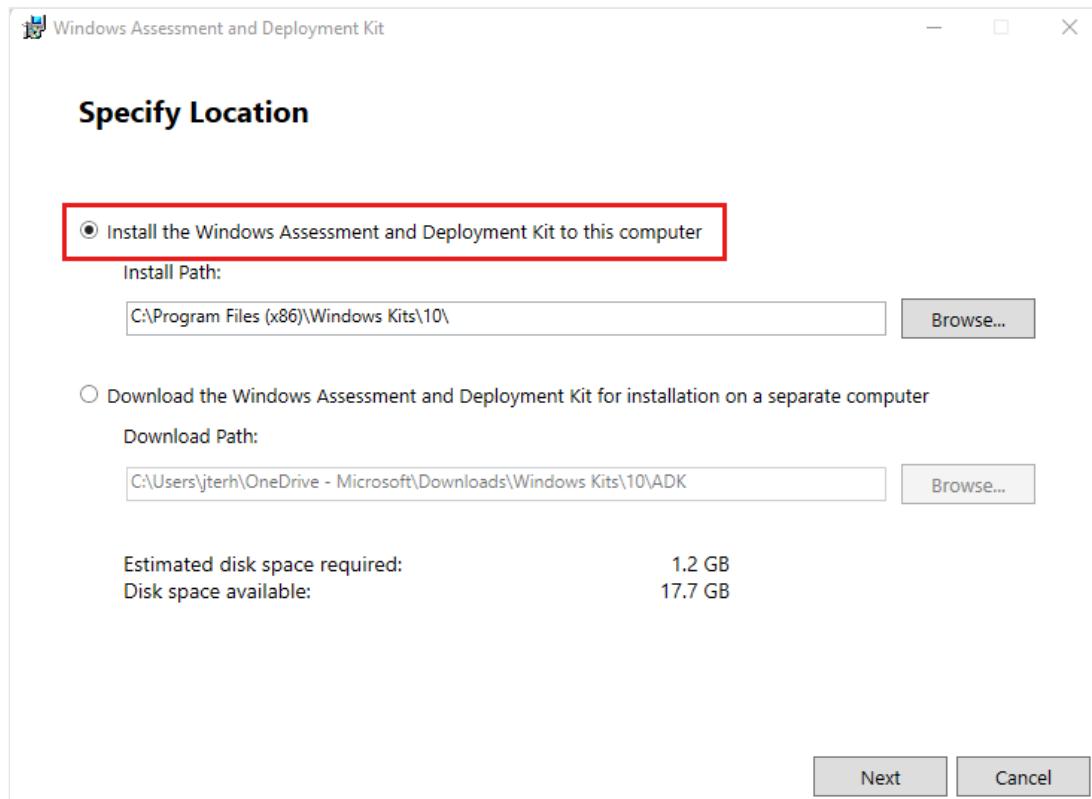


8. Share the traces with the support team on request.

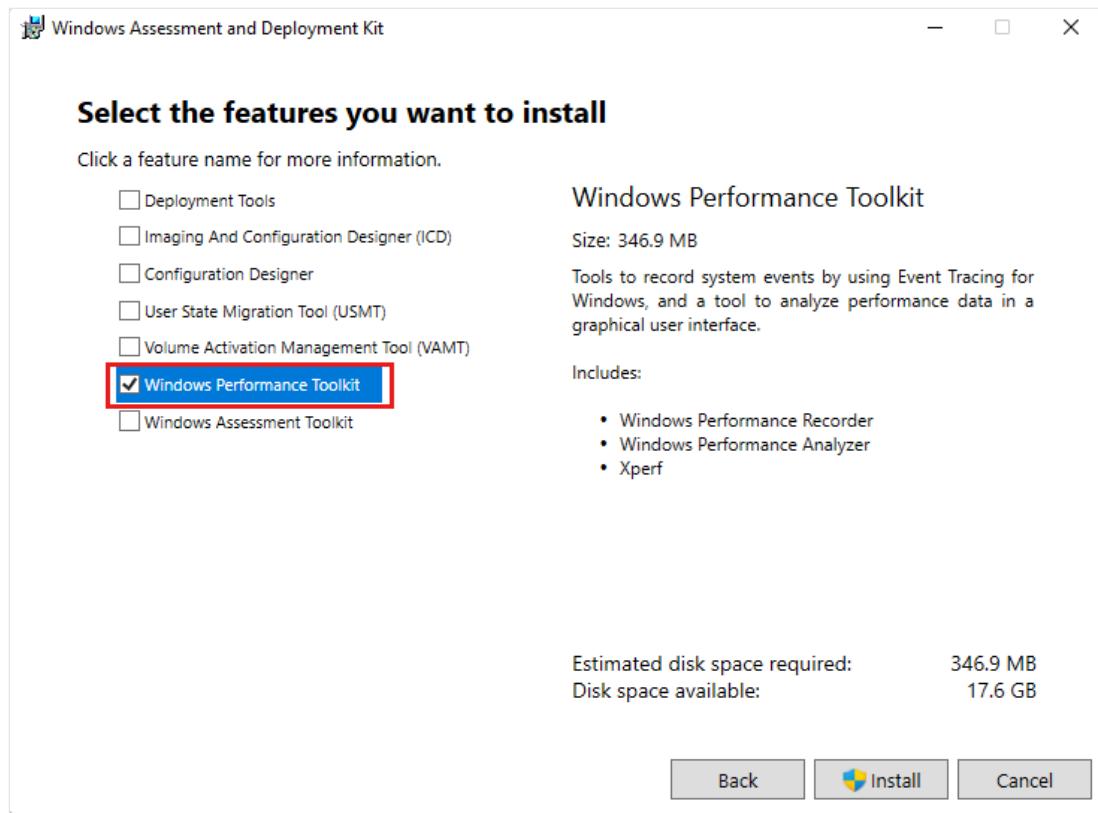
- **Extra diagnostic information.** Use the Windows Assessment and Deployment Kit to collect extra information.

1. Download the [Windows Assessment and Deployment Kit](#).

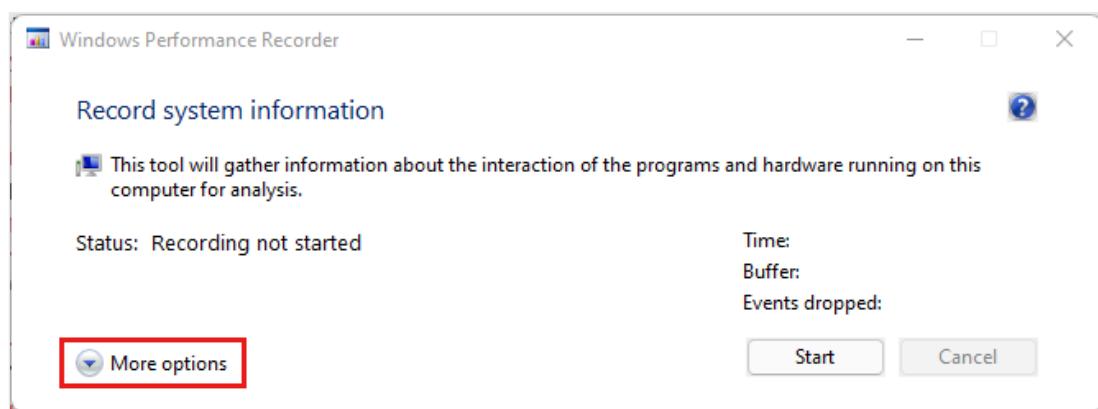
2. After downloading, start *adksetup.exe*, select **Install the Windows Assessment and Development Kit to this computer**, and then select **Next**:



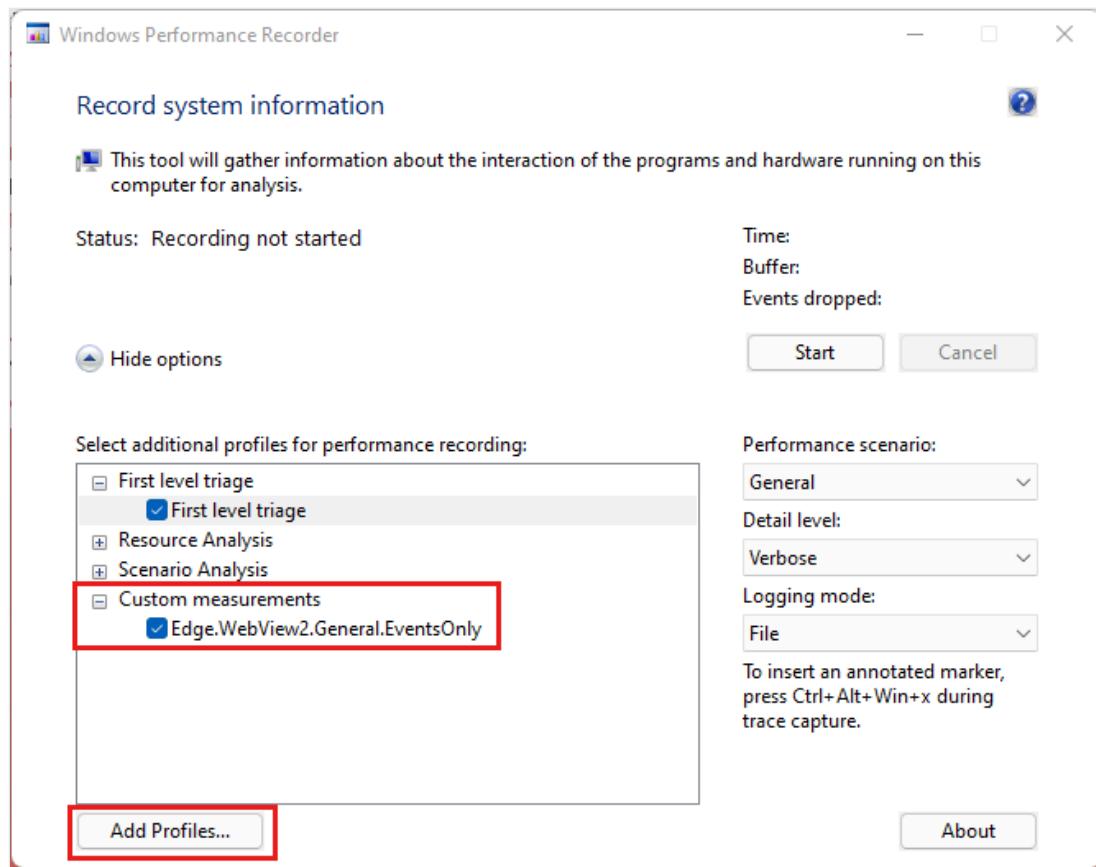
3. Continue the wizard. On the **Select the features you want to install** page, select **Windows Performance Toolkit**, and then select **Install**:



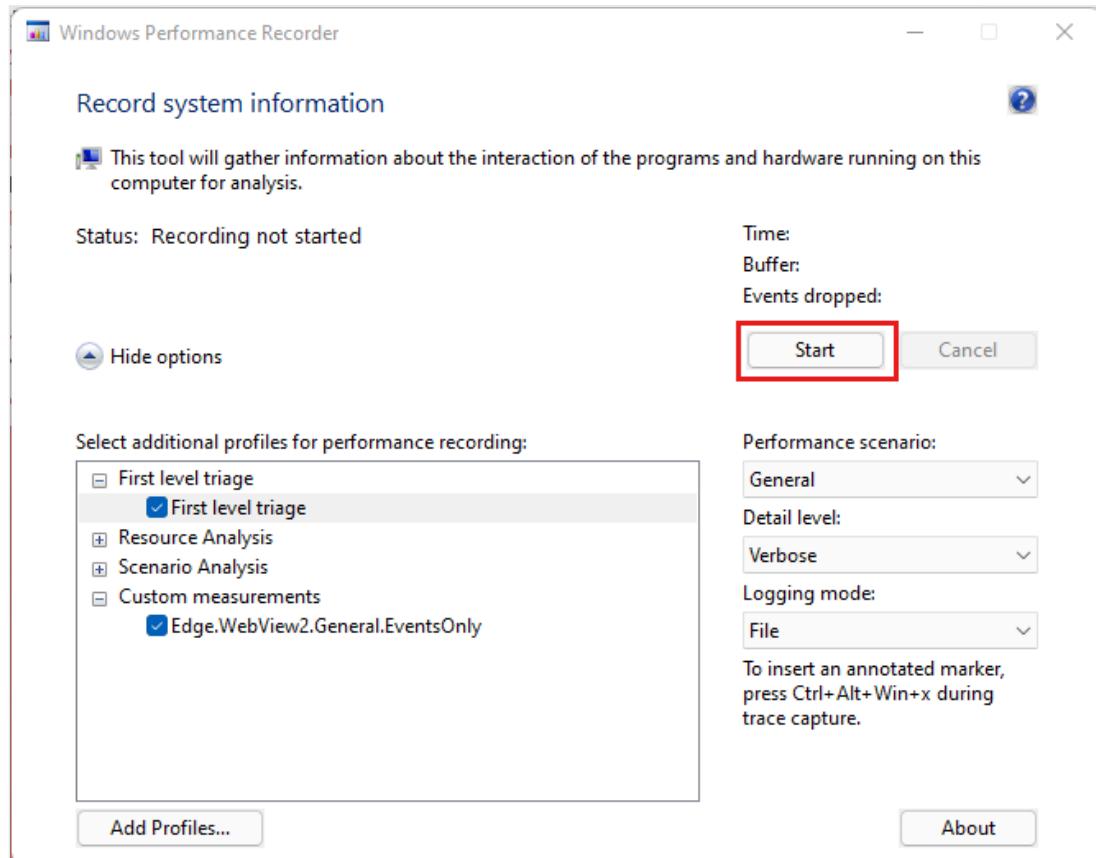
4. Complete the installation, and then start Windows Performance Recorder.
5. Download the [EdgeWebView2\\_General\\_EventsOnly.wprp](#) file to your machine and unpack it.
6. In Windows Performance Recorder, choose **More options**.



7. Choose **Add Profiles** to add the [EdgeWebView2\\_General\\_EventsOnly.wprp](#) profile that you downloaded in the previous step.

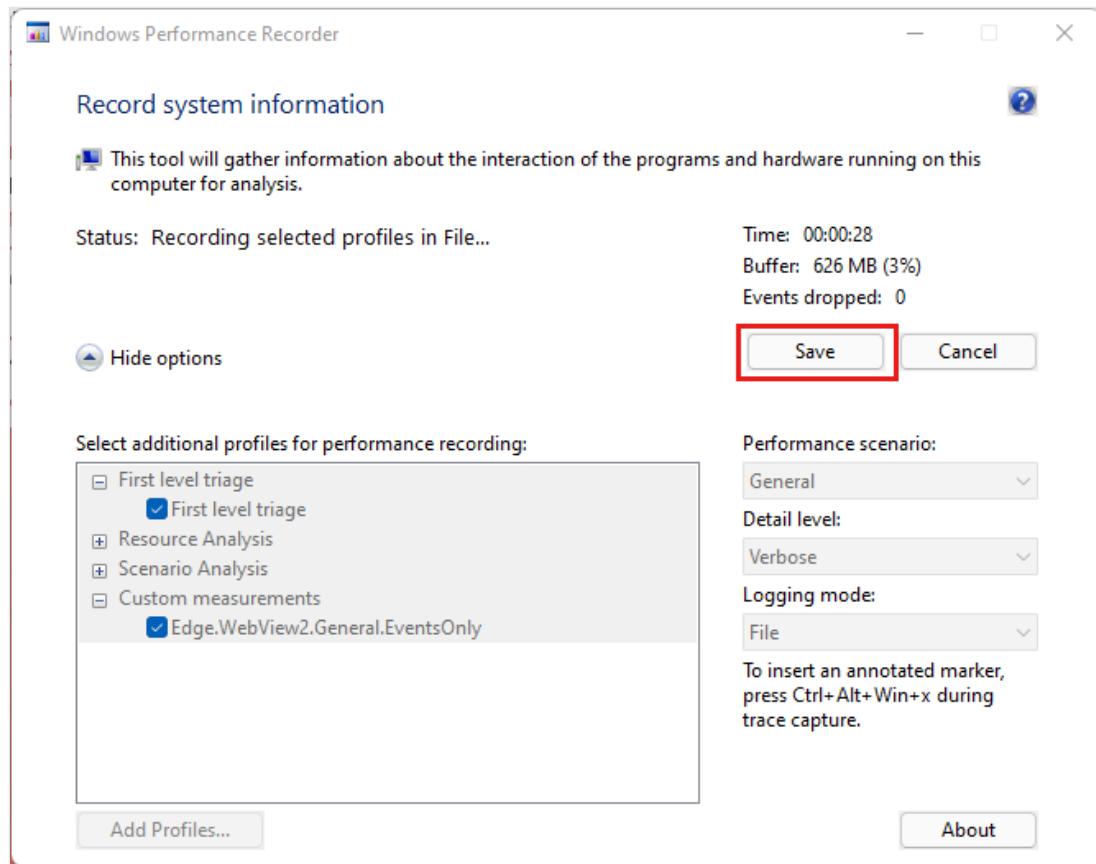


8. Choose **Start** to start recording.



9. With the recording running, start Power BI Desktop and make sure the startup issue occurs.

10. When you're done, choose **Save** to stop the recording and save the results to your computer.



11. Provide all information you collected to the support team on request.

## Data connection time-outs

When you try to create a new connection or connect to an existing Power BI dataset, Power BI Desktop might time out without establishing the connection. The connection spinner might continue to turn, but the connection never completes.

This situation can happen if your machine has a security product such as Digital Guardian or other security products installed. In some cases, the installed security product can interfere with outgoing network connection request calls, causing the connection attempt to time out or fail.

**Solution:** Try disabling the security product, and then attempt the connection again. If the connection succeeds after you disable the security product, you know that the security product was probably the cause of the connection failure.

## Other launch issues

The Power BI documentation team strives to cover as many Power BI Desktop issues as possible. The team regularly looks at issues that might affect many customers, and includes them in articles.

If your issue isn't related to an on-premises data gateway, or if the resolutions in this article don't work, you can submit a support incident to [Power BI support](#).

Whenever you experience issues with Power BI Desktop, it's helpful to turn on tracing and gather log files. Log files can help isolate and identify the issue. To turn on tracing in Power BI Desktop, choose **File > Options and settings > Options**, select **Diagnostics**, and then select **Enable tracing**. Power BI Desktop must be running to set this option, but it's helpful for any future issues associated with opening Power BI Desktop.

## Next steps

- [Get Power BI Desktop](#)

# Troubleshoot refresh scenarios

Article • 06/05/2023

This article describes different scenarios you might encounter when refreshing data within the Power BI service.

## Note

If you encounter a scenario that's not listed in this article, and if it's causing issues, you can ask for further assistance on the [community site](#), or you can create a [support ticket](#).

You should always ensure that basic requirements for refresh are met and verified:

- Verify the gateway version is up to date.
- Verify the report has a gateway selected. If there is no gateway selected, the data source might have changed or might be missing.

After you've confirmed these requirements are met, take a look through the following sections for more troubleshooting.

## Email notifications

If you're coming to this article from an email notification, and you no longer want to receive emails about refresh issues, contact your Power BI admin. Ask them to remove your email, or an email list you're subscribed to, from the appropriate datasets in Power BI. An admin uses the following area in the Power BI admin portal.

#### ► Scheduled refresh

Keep your data up to date

Configure a data refresh schedule to import data from the data source into the dataset. [Learn more](#)



Refresh frequency



Time zone



Time

[Add another time](#)

Send refresh failure notifications to



Enter email addresses

[Apply](#)

[Discard](#)

## Refresh using Web connector doesn't work properly

If you have a Web connector script that's using the [Web.Page](#) function, and you've updated your dataset or report after November 18, 2016, you must use a gateway for refresh to work properly.

## Unsupported data source for refresh

When you configure a dataset, you might get an error indicating the dataset uses an unsupported data source for refresh. For details, see [Troubleshooting unsupported data source for refresh](#).

## Dashboard doesn't reflect changes after refresh

Wait 10-15 minutes for a refresh to be reflected in the dashboard tiles. If it still doesn't show up, repin the visualization to the dashboard.

# GatewayNotReachable when setting credentials

You might encounter a `GatewayNotReachable` error when you try to set credentials for a data source, which can be the result of an outdated gateway. Install the latest gateway and try again.

## Processing Error: The following system error occurred: Type Mismatch

This error could be an issue with your [M script](#) within your Power BI Desktop file or Excel workbook. It can also be due to an out-of-date Power BI Desktop version.

## Tile refresh errors

For a list of errors you might encounter with dashboard tiles, and explanations, see [Troubleshooting tile errors](#).

## Refresh fails when updating data from sources that use Azure AD OAuth

The Azure Active Directory (Azure AD) OAuth token, used by many different data sources, expires in approximately one hour. Sometimes that token expires before the data has finished loading, since the Power BI service waits for up to two hours when loading data. In that situation, the data loading process can fail with a credentials error.

Data sources that use Azure AD OAuth include [Microsoft Dynamics CRM Online](#), [SharePoint Online \(SPO\)](#), and others. If you're connecting to such data sources, and get a credentials failure when loading data takes more than an hour, OAuth might be the reason.

Microsoft is investigating a solution that allows the data loading process to refresh the token and continue. However, if your Dynamics CRM Online or SharePoint Online instance is so large that it runs over the two-hour data-load threshold, the Power BI service might report a data load time-out. This data load time-out also applies to other Azure AD OAuth data sources.

For refresh to work properly when connecting to a [SharePoint Online](#) data source by using Azure AD OAuth, you must use the same account that you use to sign in to the [Power BI service](#).

If you want to connect to a data source from Power BI service by using OAuth2, the data source must be in the same tenant as the Power BI service. Currently, multi-tenant connection scenarios aren't supported with OAuth2.

## Uncompressed data limits for refresh

The maximum size for datasets imported into the **Power BI service** is 1 GB. These datasets are heavily compressed to ensure high performance. In addition, in shared capacity, the service places a limit of 10 GB on the amount of uncompressed data that's processed during refresh. This limit accounts for the compression, and therefore is much higher than the 1-GB maximum dataset size. Datasets in Power BI Premium aren't subject to these limits. If refresh in the Power BI service fails for this reason, reduce the amount of data being imported to Power BI and try again.

## Scheduled refresh time-out

Scheduled refresh for imported datasets time out after two hours. This time-out is increased to five hours for datasets in **Premium** workspaces. If you encounter this limit, consider reducing the size or complexity of your dataset, or consider refactoring the large dataset into multiple smaller datasets.

## Scheduled refresh disabled

If a scheduled refresh fails four times in a row, Power BI disables the refresh. Address the underlying problem, and then re-enable the scheduled refresh.

However, if the dataset resides in a workspace under Embedded capacity, and that capacity is switched off off, the *first* attempt at refresh will fail(since the capacity is switched off), and in this circumstance its scheduled refresh is immediately disabled.

## Access to the resource is forbidden

This error can occur because of expired cached credentials. Clear your internet browser cache, then sign in to Power BI and go to <https://app.powerbi.com?alwaysPromptForContentProviderCreds=true> to force an update of your credentials.

## Data refresh failure because of password change or expired credentials

Data refresh can also fail due to expired cached credentials. Clear your internet browser cache, then sign in to Power BI and go to <https://app.powerbi.com?alwaysPromptForContentProviderCreds=true>, which forces an update of your credentials.

## Refresh a column of the ANY type that contains TRUE or FALSE results in unexpected values

When you create a report in Power BI Desktop that has an ANY data type column containing TRUE or FALSE values, the values of that column can differ between Power BI Desktop and the Power BI service after a refresh. In Power BI Desktop, the underlying engine converts the boolean values to strings, retaining TRUE or FALSE values. In the Power BI service, the underlying engine converts the values to objects, and then converts the values to -1 or 0.

Visuals created in Power BI Desktop by using such columns might behave or appear as designed prior to a refresh event, but might change (due to TRUE/FALSE being converted to -1/0) after the refresh event.

## Resolve the error: Container exited unexpectedly with code 0x0000DEAD

If you get the **Container exited unexpectedly with code 0x0000DEAD** error, try to disable the scheduled refresh and republish the dataset.

## Refresh operation throttled by Power BI Premium

A Premium capacity might throttle data refresh operations when too many datasets are being processed concurrently. Throttling can occur in Power BI Premium capacities. When a refresh operation is canceled, the following error messages are logged into the refresh history:

*You've exceeded the capacity limit for dataset refreshes. Try again when fewer datasets are being processed.*

If the error occurs frequently, use the [schedule view](#) to determine whether the scheduled refresh events are properly spaced. To understand the maximum number of concurrent refreshes allowed per SKU, review the [Capacities and SKUs](#) table.

To resolve this error, you can modify your refresh schedule to perform the refresh operation when fewer datasets are being processed. You can also increase the time between refresh operations for all datasets in your refresh schedule on the affected Premium capacity. You can retry the operation if you're using custom [XMLA operations](#).

#### *Capacity level limit exceeded.*

This error indicates you have too many datasets running refresh at the same time, based on the capacity your organization has purchased. You can retry the refresh operation, or reschedule the refresh time to address this error.

#### *Node level limit exceeded.*

This error indicates a system error in Power BI Premium based on datasets residing on a given physical node. You can retry the refresh operation, or reschedule the refresh time to address this error.

## Dataflows or datamart failures in Premium workspaces

Some connectors are not supported for dataflows and datamarts in Premium workspaces. When using an unsupported connector, you may receive the following error: *Expression.Error: The import "<"connector name">" matches no exports. Did you miss a module reference?*

The following connectors are not supported for dataflows and datamarts in Premium workspaces:

- Linkar
- Actian
- AmazonAthena
- AmazonOpenSearchService
- BIConnector
- DataVirtuality
- DenodoForPowerBI
- Exasol
- Foundry
- Indexima
- IRIS
- JethroODBC
- Kyligence
- MariaDB

- MarkLogicODBC
- OpenSearchProject
- QubolePresto
- SingleStoreODBC
- StarburstPresto
- TibcoTdv

The use of the previous list of connectors with dataflows or datamarts is only supported workspaces that are not Premium.

## Next steps

- [Data refresh in Power BI](#)
- [Troubleshoot the On-premises data gateway](#)
- [Troubleshooting the Power BI Gateway - Personal](#)

More questions? [Try asking the Microsoft Power BI Community](#)↗

# Troubleshoot tile errors

Article • 02/02/2023

This article lists and explains the common errors that can occur with tile refresh in Power BI. If an error that's not listed causes you problems, you can ask for assistance on the [Power BI community site](#) or file a [support ticket](#).

## Error list

The following list explains and offers solutions for common tile refresh errors.

- **Power BI encountered an unexpected error while loading the model. Please try again later.**

or

**Couldn't retrieve the data model. Please contact the dashboard owner to make sure the data sources and model exist and are accessible.**

Power BI couldn't access your data because the data source wasn't reachable. This issue can happen if the data source was removed, renamed, or moved, if the source is offline, or if permissions have changed. Check that the source is still in the specified location and you still have permission to access it. If that isn't the problem, the source might be slow. Try again later during a time when the load on the source is less. If it's an on-premises source, the data source owner might be able to provide more information.

- **You don't have permission to view this tile or open the workbook.**

Contact the dashboard owner to make sure the data sources and model exist and are accessible for your account.

- **Power BI visuals have been disabled by your administrator.**

Your Power BI administrator has disabled using Power BI visuals for your organization or your security group. You can't use Power BI visuals from the [Microsoft marketplace](#) or import private visuals from a file. You can use only the pre-packed set of visuals.

- **Data shapes must contain at least one group or calculation that outputs data. Please contact the dashboard owner.**

There's no data to display because the query is empty. Try adding some fields from the field list to the visual and repinning it.

- **Can't display the data because Power BI can't determine the relationship between two or more fields.**

You're trying to use two or more fields from tables that aren't related. You need to remove the unrelated fields from the visual and then create a relationship between the tables. Once you create the relationship, you can add the fields back to the visual. You can use Power BI Desktop or Power Pivot for Excel for this process. For more information, see [Create and manage relationships in Power BI Desktop](#).

- **The groups in the primary axis and the secondary axis overlap. Groups in the primary axis can't have the same keys as groups in the secondary axis.**

This issue is usually transient, and typically happens when you're moving groups from rows to columns. The error should disappear when you finish moving all the groups. If you still see the message, try switching fields between the rows, columns, or axis legend, or removing fields from the visual.

- **This visual has exceeded the available resources. Try filtering to decrease the amount of data displayed.**

The visual has tried to query too much data for Power BI to complete the result with available resources. Try filtering the visual to reduce the amount of data in the result.

- **We are not able to identify the following fields: {0}. Please update the visual with fields that exist in the dataset.**

The field was probably deleted or renamed. You can remove the broken field from the visual, add a different field, and repin it.

- **Couldn't retrieve the data for this visual. Please try again later.**

This issue is usually transient. If you try again later and still see this message, [contact support ↗](#).

- **Tiles continue to show unfiltered data after you enable single-sign on (SSO).**

This issue can happen if the underlying dataset uses DirectQuery mode or a Live Connection to Analysis Services through an on-premises data gateway. In this issue, the tiles continue to show unfiltered data after you enable SSO for the data source, until the next tile refresh. At the next tile refresh, Power BI uses SSO as configured, and the tiles show the data filtered according to the user identity.

To see the filtered data immediately, you can force a tile refresh. Select the **Refresh** icon at the upper right of a Power BI dashboard.

As a dataset owner, you can also increase the tile refresh frequency to 15 minutes to accelerate tile refresh. Select the gear icon in the upper right corner of the Power BI service, and then select **Settings**. On the **Datasets** tab, expand **Scheduled refresh**, and under **Automatic dashboard tile and metric refresh**, change **Refresh frequency**. Make sure you reset the configuration to the original refresh frequency after Power BI does the next tile refresh.

ⓘ Note

**Automatic dashboard tile and metric refresh** is available only for datasets in DirectQuery or Live Connection modes. Datasets in Import mode don't need a separate tile refresh because the tiles refresh automatically during the next scheduled data refresh.

## Support contact

If you're still having problems, [contact support ↗](#) and ask them to investigate further.

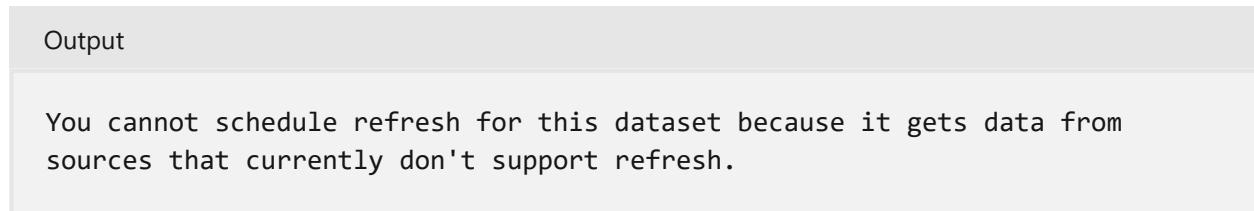
## Next steps

- [Troubleshoot the on-premises data gateway](#)
- [Troubleshoot Power BI personal gateway](#)
- More questions? [Try the Power BI community site. ↗](#)

# Troubleshooting unsupported data source for refresh

Article • 01/12/2023

You might see an error when trying to configure a dataset for scheduled refresh.

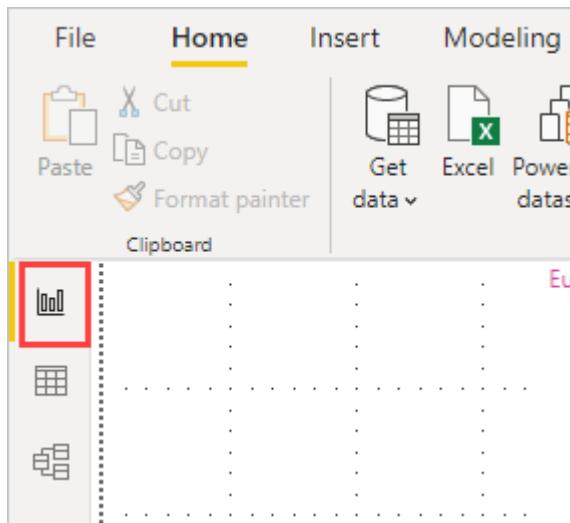


This issue happens when the data source you used, within Power BI Desktop, isn't supported for refresh. You need to find the data source that you're using and compare that against the list of supported data sources at [Refresh data in Power BI](#).

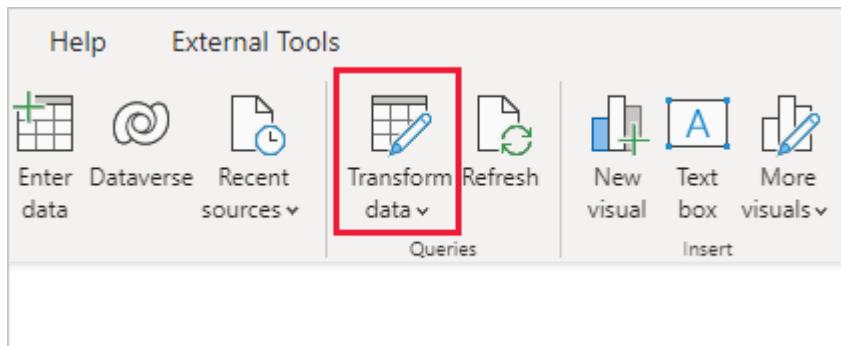
## Find the data source

If you aren't sure what data source was used, you can find that using the following steps within Power BI Desktop.

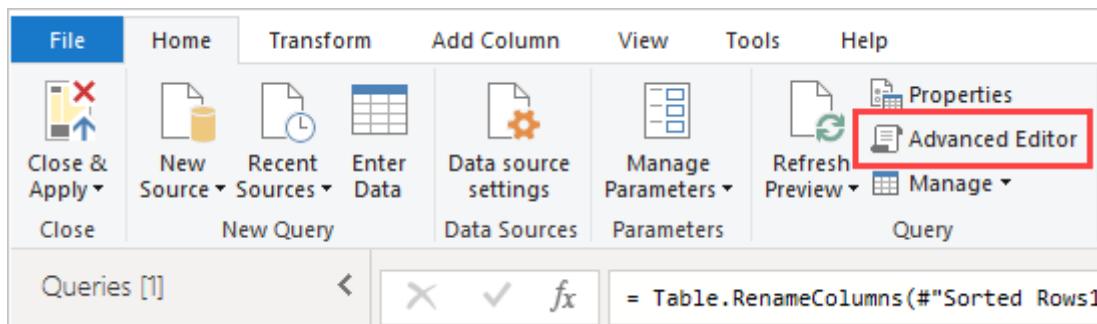
1. In Power BI Desktop, make sure you are on the **Report** pane.



2. Select **Transform data** from the ribbon bar.



### 3. Select Advanced Editor.



### 4. Make note of the provider listed for the source. In this example, the provider is ActiveDirectory.

The screenshot shows the 'Advanced Editor' dialog box. The title bar says 'Advanced Editor'. The main area contains a Power BI query:

```
let
 Source = ActiveDirectory Domains("contoso.com"),
 contoso.com = source{[Domain="contoso"]}[#Object Categories],
 user1 = contoso.com{[Category="user"]}[Objects]
in
 user1
```

A red box highlights the word 'ActiveDirectory' in the first line of the query. At the bottom of the editor, a green checkmark indicates: 'No syntax errors have been detected.' There are 'Done' and 'Cancel' buttons at the bottom right.

### 5. Compare the provider with the list of supported data sources found in [Power BI data sources](#).

 **Note**

For refresh issues related to dynamic data sources, including data sources that include hand-authored queries, see [Refresh and dynamic data sources](#).

## Next steps

- [Data Refresh](#)
- [Power BI Gateway - Personal](#)
- [On-premises data gateway](#)
- [Troubleshooting the On-premises data gateway](#)
- [Troubleshooting the Power BI Gateway - Personal](#)

More questions? [Try asking the Power BI Community](#) 

# Troubleshoot scheduled refresh for Azure SQL databases in Power BI

Article • 02/20/2023

For detailed information about refresh, see [Refresh data in Power BI](#) and [Configure scheduled refresh](#).

While you set up scheduled refresh for an Azure SQL database, if you get an error with error code 400 when editing the credentials, try these steps to configure the correct firewall rule:

1. Sign in to the [Azure portal](#).
2. Go to the Azure SQL database for which you're configuring refresh.
3. In the **Overview** page, select **Set server firewall**.
4. On the **Networking** page, select **Allow Azure services and resources to access this server** and choose **Save**.

The screenshot shows the 'Networking' page for an Azure SQL database. The 'Firewall rules' section displays a single rule named 'Clientip-2022-6-10\_22-40-24' with the 'Start IPv4 address' set to 76.110.122.109 and the 'End IPv4 address' also set to 76.110.122.109. Below the table, there is an 'Exceptions' section containing a single checkbox: 'Allow Azure services and resources to access this server'. This checkbox is highlighted with a red border.

More questions? [Ask the Power BI Community](#).

# Troubleshoot the connection from Excel to Power BI data

Article • 12/02/2022

There may be times when connecting Excel to Power BI data that you get an unexpected result, or the feature doesn't work as you expected. This page provides solutions for common issues when analyzing Power BI data in Excel.

## Note

There are separate articles for different connection types. Those articles are as follows:

- [Start in Power BI with Analyze in Excel](#).
- [Start in Excel to connect to Power BI datasets](#).

If you encounter a scenario that's not listed below, ask for assistance on the [Power BI community site](#), or create a support ticket.

If you need to troubleshoot an issue with Power BI data in Excel, see the following sections:

- [Forbidden error](#)
- [Unable to access on-premises Analysis Services](#)
- [Can't drag anything to the PivotTable Values area \(no measures\)](#)

If you need to troubleshoot an issue in Power BI with Analyze in Excel, see the following sections:

- [Connection cannot be made error](#)
- [Can't find OLAP cube model error](#)
- [Token expired error](#)

## Forbidden error

A user may have more than one Power BI account. When Excel tries to connect to Power BI by using credentials from one of those accounts, it may attempt to use credentials that don't have access to the desired dataset or report.

When this situation occurs, you may receive an error titled **Forbidden**. This error means you may be signed into Power BI with credentials that don't have permission to access the dataset. After encountering the **Forbidden** error and when you see the prompt, type the credentials that have permission to access the dataset you're trying to use.

If you still run into errors, log into Power BI with the account that has permission. Then, verify that you can view and access the dataset in Power BI that you're attempting to access in Excel.

## Unable to access on-premises Analysis Services

If you're trying to access a dataset that has a live connection to SQL Server Analysis Services or Azure Analysis Services data, you may receive an error message. This error may occur because a user can't connect to Power BI datasets. This situation may happen when you build datasets on live connections to Analysis Services unless the user has read access to the data in Analysis Services in addition to the datasets permissions in Power BI.

## Can't drag anything to the PivotTable Values area

Excel connects to Power BI through an external OLAP model. When these applications connect, the *PivotTable* requires you to define **measures** in the external model because all calculations are performed on the server. This requirement is different from working with a local data source, such as tables in Excel and working with datasets in **Power BI Desktop** or the **Power BI service**). In those cases, the tabular model is available locally, and [you can use implicit measures](#). Implicit measures are generated dynamically, and not stored in the data model. In these cases, the behavior in Excel is different from the behavior in **Power BI Desktop** or the **Power BI service**. For instance, there may be columns in the data that can be treated as measures in Power BI, but can't be used as measures, or values, in Excel.

To address this issue, you have a few options:

- [Create measures in your data model in Power BI Desktop](#). Then, publish the data model to the **Power BI service** and access that published dataset from Excel.
- [Create measures in your data model from Excel PowerPivot](#).
- If you imported data from an Excel workbook that had only tables and no data model, then you can [add the tables to the data model](#). Then, follow the steps in the previous step to create measures in your data model.

Once you define your measures in the model in the Power BI service, you can use them in the **Values** area in Excel PivotTables.

## Connection cannot be made

The primary cause for a **Connection cannot be made** error is that your computer's OLE DB provider client libraries aren't current.

## Can't find OLAP cube model

The primary cause for a **Can't find OLAP cube model** error is that the dataset you're trying to access has no data model, and therefore the dataset can't be analyzed in Excel.

## Token expired error

The primary cause for a **Token expired** error is that you haven't recently used the **Analyze in Excel** feature on the computer you're using. To resolve this error, reenter your credentials or reopen the file, and the error should go away.

## Next steps

[Analyze in Excel](#)

[Tutorial: Create your own measures in Power BI Desktop](#)

[Measures in PowerPivot ↗](#)

[Create a Measure in PowerPivot ↗](#)

[Add worksheet data to a Data Model using a linked table ↗](#)

# Error: We couldn't find any data in your Excel workbook

Article • 02/22/2023

## ⓘ Note

This article applies to Excel 2007 and later.

When you import an Excel workbook into the Power BI service, you might see the following error:

Output

Error: We couldn't find any data formatted as a table. To import from Excel into the Power BI service, you need to format the data as a table. Select all the data you want in the table and press Ctrl+T.

## We couldn't find any data formatted as a table ✖

To import from Excel into the Power BI service, you need to format the data as a table. Select all the data you want in the table and press Ctrl + T.

[Learn more about solving this issue](#)

Close

## Quick solution

1. Edit your workbook in Excel.
2. Select the range of cells that contain your data. The first row should contain your column headers, the column names.
3. Press **Ctrl + T** to create a table.
4. Save your workbook.
5. Return to the Power BI service and import your workbook again, or if you're working in Excel 2016 and you've saved your workbook to OneDrive for work or school, in Excel, select **File > Publish**.

## Details

### Cause

In Excel, you can create a *table* out of a range of cells, which makes it easier to sort, filter, and format data.

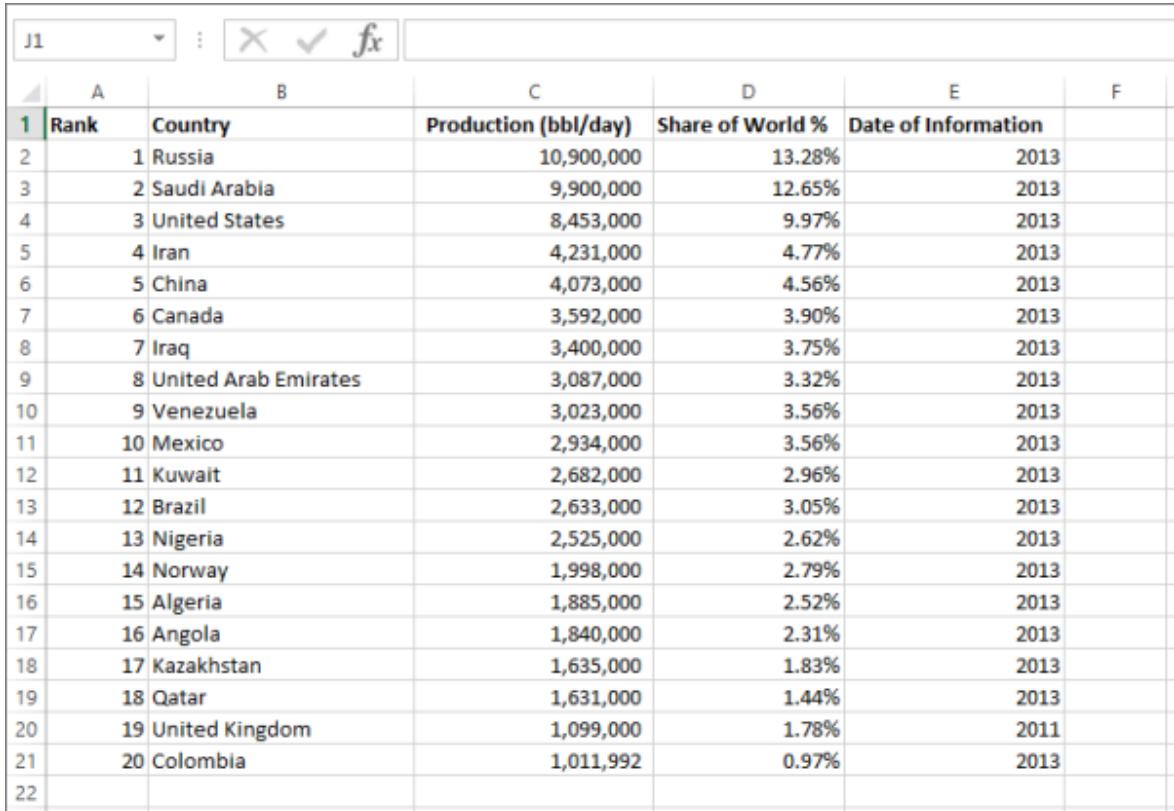
When you import an Excel workbook, Power BI looks for these tables and imports them into a dataset. If it doesn't find any tables, you see this error message.

## Solution

1. Open your workbook in Excel.

 Note

The pictures here are of Excel 2013. If you're using a different version, things might look a little different, but the steps are the same.



	A	B	C	D	E	F
1	Rank	Country	Production (bbl/day)	Share of World %	Date of Information	
2	1	Russia	10,900,000	13.28%	2013	
3	2	Saudi Arabia	9,900,000	12.65%	2013	
4	3	United States	8,453,000	9.97%	2013	
5	4	Iran	4,231,000	4.77%	2013	
6	5	China	4,073,000	4.56%	2013	
7	6	Canada	3,592,000	3.90%	2013	
8	7	Iraq	3,400,000	3.75%	2013	
9	8	United Arab Emirates	3,087,000	3.32%	2013	
10	9	Venezuela	3,023,000	3.56%	2013	
11	10	Mexico	2,934,000	3.56%	2013	
12	11	Kuwait	2,682,000	2.96%	2013	
13	12	Brazil	2,633,000	3.05%	2013	
14	13	Nigeria	2,525,000	2.62%	2013	
15	14	Norway	1,998,000	2.79%	2013	
16	15	Algeria	1,885,000	2.52%	2013	
17	16	Angola	1,840,000	2.31%	2013	
18	17	Kazakhstan	1,635,000	1.83%	2013	
19	18	Qatar	1,631,000	1.44%	2013	
20	19	United Kingdom	1,099,000	1.78%	2011	
21	20	Colombia	1,011,992	0.97%	2013	
22						

2. Select the range of cells that contain your data. The first row should contain your column headers, the column names.

A1 : X ✓ fx Rank

	A	B	C	D	E	F
1	Rank	Country	Production (bbl/day)	Share of World %	Date of Information	
2	1	Russia	10,900,000	13.28%	2013	
3	2	Saudi Arabia	9,900,000	12.65%	2013	
4	3	United States	8,453,000	9.97%	2013	
5	4	Iran	4,231,000	4.77%	2013	
6	5	China	4,073,000	4.56%	2013	
7	6	Canada	3,592,000	3.90%	2013	
8	7	Iraq	3,400,000	3.75%	2013	
9	8	United Arab Emirates	3,087,000	3.32%	2013	
10	9	Venezuela	3,023,000	3.56%	2013	
11	10	Mexico	2,934,000	3.56%	2013	
12	11	Kuwait	2,682,000	2.96%	2013	
13	12	Brazil	2,633,000	3.05%	2013	
14	13	Nigeria	2,525,000	2.62%	2013	
15	14	Norway	1,998,000	2.79%	2013	
16	15	Algeria	1,885,000	2.52%	2013	
17	16	Angola	1,840,000	2.31%	2013	
18	17	Kazakhstan	1,635,000	1.83%	2013	
19	18	Qatar	1,631,000	1.44%	2013	
20	19	United Kingdom	1,099,000	1.78%	2011	
21	20	Colombia	1,011,992	0.97%	2013	
22						

3. In the ribbon on the **INSERT** tab, select **Table**. Or, as a shortcut, press **Ctrl + T**.

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER POWER QU

PivotTable Recommended PivotTables Tables Table Pictures Online SmartArt Pictures Screenshot Illustrations Store Bing Maps My Apps People Graph Recommended Charts Charts

A1 Table (Ctrl+T)

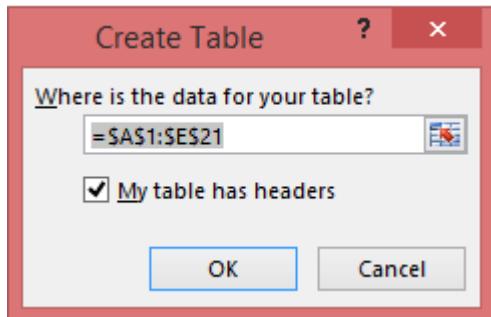
Create a table to organize and analyze related data.

Tables make it easy to sort, filter, and format data within a sheet.

Microsoft Power Query for Excel Tell me more

	A	B	C	D	E	F
1	Rank	Country	Production (bbl/day)	Share of World %	Date of Information	
2	1	Russia	10,900,000	13.28%	2013	
3	2	Saudi Arabia	9,900,000	12.65%	2013	
4	3	United States	8,453,000	9.97%	2013	
5	4	Iran	4,231,000	4.77%	2013	
6	5	China	4,073,000	4.56%	2013	
7	6	Canada	3,592,000	3.90%	2013	
8	7	Iraq	3,400,000	3.75%	2013	
9	8	United Arab Emirates	3,087,000	3.32%	2013	
10	9	Venezuela	3,023,000	3.56%	2013	
11	10	Mexico	2,934,000	3.56%	2013	
12	11	Kuwait	2,682,000	2.96%	2013	
13	12	Brazil	2,633,000	3.05%	2013	
14	13	Nigeria	2,525,000	2.62%	2013	
15	14	Norway	1,998,000	2.79%	2013	
16	15	Algeria	1,885,000	2.52%	2013	
17	16	Angola	1,840,000	2.31%	2013	
18	17	Kazakhstan	1,635,000	1.83%	2013	
19	18	Qatar	1,631,000	1.44%	2013	
20	19	United Kingdom	1,099,000	1.78%	2011	
21	20	Colombia	1,011,992	0.97%	2013	
22						

4. You see the following dialog. Make sure **My table has headers** is selected, then choose **OK**.

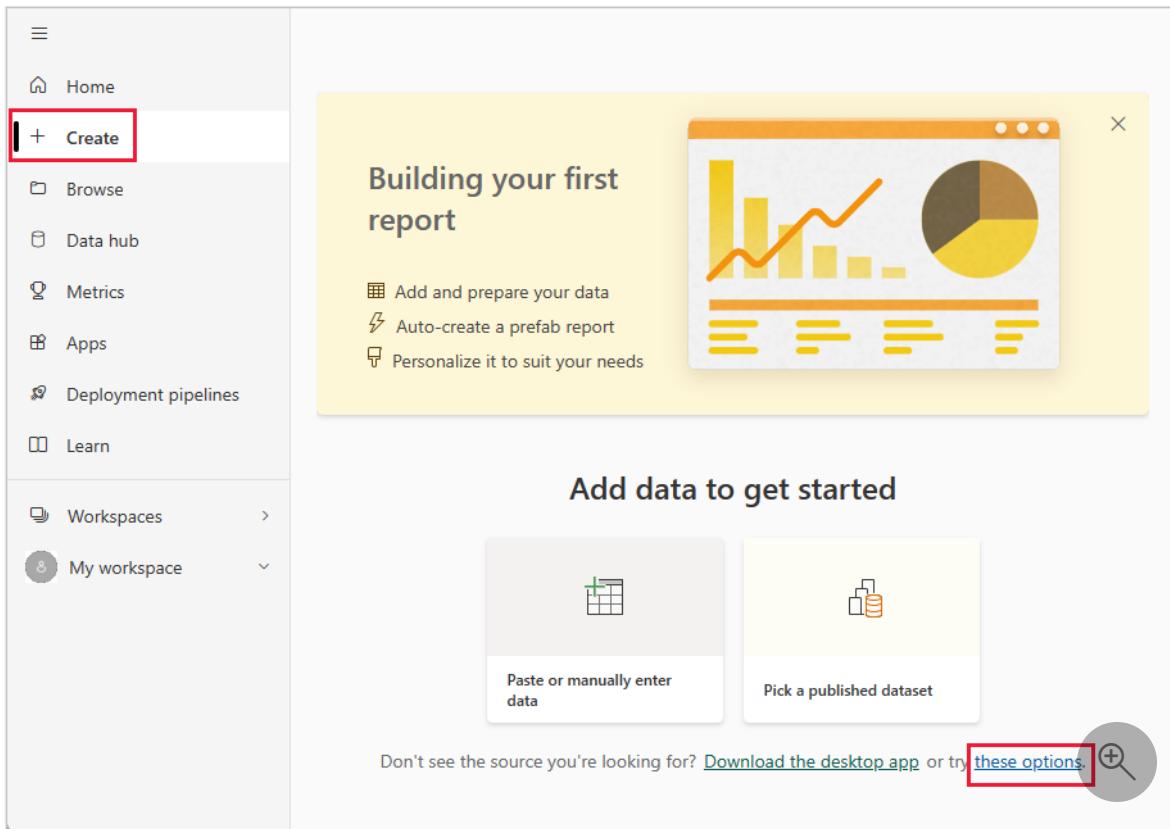


Now your data is formatted as a table.

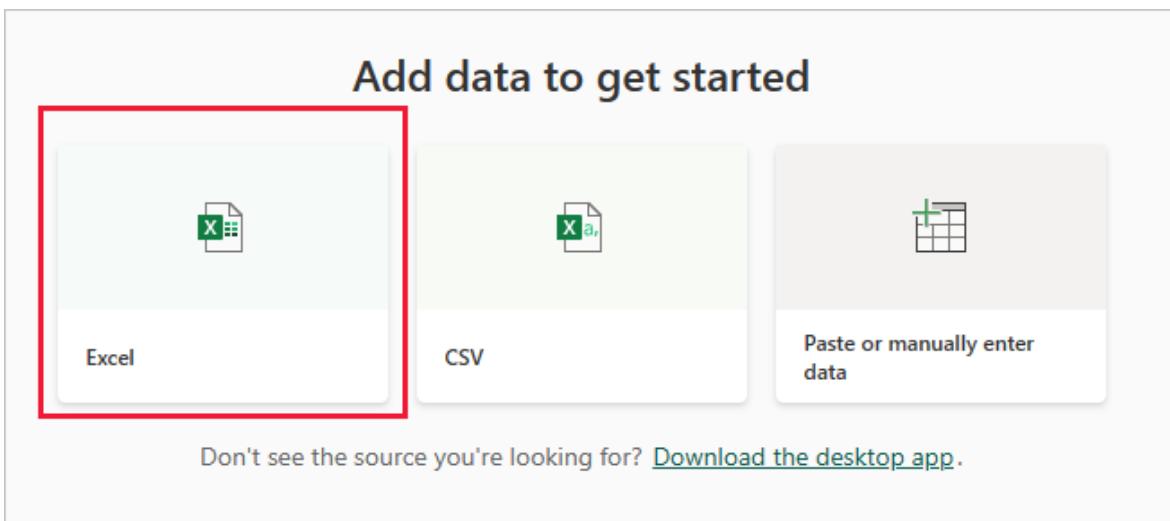
A1	B	C	D	E	F
1	Rank	Country	Production (bbl/day)	Share of World %	Date of Information
2	1	Russia	10,900,000	13.28%	2013
3	2	Saudi Arabia	9,900,000	12.65%	2013
4	3	United States	8,453,000	9.97%	2013
5	4	Iran	4,231,000	4.77%	2013
6	5	China	4,073,000	4.56%	2013
7	6	Canada	3,592,000	3.90%	2013
8	7	Iraq	3,400,000	3.75%	2013
9	8	United Arab Emirates	3,087,000	3.32%	2013
10	9	Venezuela	3,023,000	3.56%	2013
11	10	Mexico	2,934,000	3.56%	2013
12	11	Kuwait	2,682,000	2.96%	2013
13	12	Brazil	2,633,000	3.05%	2013
14	13	Nigeria	2,525,000	2.62%	2013
15	14	Norway	1,998,000	2.79%	2013
16	15	Algeria	1,885,000	2.52%	2013
17	16	Angola	1,840,000	2.31%	2013
18	17	Kazakhstan	1,635,000	1.83%	2013
19	18	Qatar	1,631,000	1.44%	2013
20	19	United Kingdom	1,099,000	1.78%	2011
21	20	Colombia	1,011,992	0.97%	2013
22					

5. Save your workbook.

6. Return to the Power BI service. Select **Create**, then choose **these options**.



7. In the Add data to get started window, select Excel.



8. Import your Excel workbook again. This time, the import should find the table and succeed.

If the import still fails, let us know by selecting **Community** in the help menu:



Getting started

Community

Feature Requests

Help topics

Power BI for developers

About Power BI

# Troubleshoot Access and Excel XLS import issues in Power BI Desktop

Article • 01/23/2023

In Power BI Desktop, imported Access databases and Excel 97-2003 XLS files both use the *Access Database Engine*. Three common situations can prevent the Access Database Engine from working properly:

- No Access Database Engine is installed.
- The Access Database Engine bit version, 32-bit or 64-bit, is different from the Power BI Desktop bit version.
- You're using Access or XLS files with a Microsoft 365 subscription.

## No Access Database Engine installed

If a Power BI Desktop error message indicates the Access Database Engine isn't installed, [install the Access Database Engine](#) from the downloads page. Install the version, either 32-bit or 64-bit, that matches your Power BI Desktop version.

If you work with dataflows and use a gateway to connect to the data, you must install the Access Database Engine on the computer that runs the gateway.

### Note

If the Access Database Engine bit version you install is different from your Microsoft Office bit version, your Office applications won't be able to use the Access Database Engine.

## Access Database Engine bit version is different from Power BI Desktop bit version

This situation usually occurs when the installed Microsoft Office version is 32-bit and the installed Power BI Desktop version is 64-bit. The opposite can also happen, and the bit version mismatch occurs in either case.

Any of the following solutions can remedy this bit-version mismatch error. You can also apply these solutions to other mismatches, for example other 32-bit COM applications like Visual Studio SSDT.

If you're using Access or XLS files with a Microsoft 365 subscription, see [Access or XLS files with Microsoft 365](#) for a different issue and resolution.

## Solution 1: Change Power BI Desktop bit version to match Microsoft Office bit version

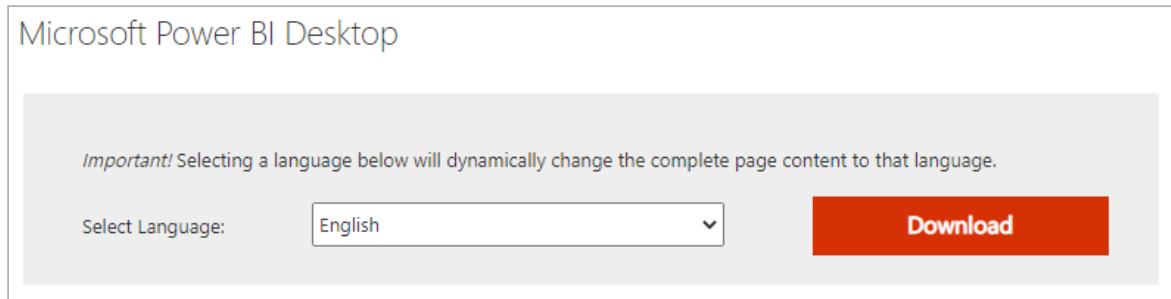
To change the bit version of Power BI Desktop, uninstall Power BI Desktop, and then install the version of Power BI Desktop that matches your Office installation.

### Note

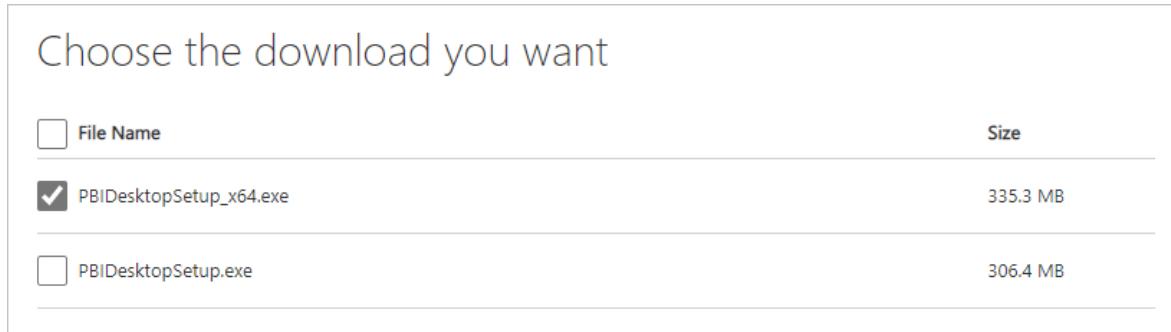
If you use the 32-bit version of Power BI Desktop to create very large data models, you might experience out-of-memory issues.

To select a version of Power BI Desktop:

1. On the [Power BI Desktop download page](#), choose your language, and then select **Download**.



2. On the next screen, select the checkbox next to **PBIDesktop.msi** for the 32-bit version, or **PBIDesktop\_x64.msi** for the 64-bit version, and then select **Next**.



## Solution 2: Change Microsoft Office bit version to match Power BI Desktop bit version

To change the bit version of Microsoft Office to match the bit version of your Power BI Desktop installation:

1. Uninstall Microsoft Office.
2. Install the version of Office that matches your Power BI Desktop installation.

## Solution 3: Save the XLS file as XLSX

If the error occurs with an Excel 97-2003 XLS workbook, you can avoid using the Access Database Engine by opening the XLS file in Excel and saving it as an XLSX file.

## Solution 4: Install both versions of the Access Database Engine

You can install both versions of the Access Database Engine to resolve the issue for Power Query for Excel and Power BI Desktop. This workaround isn't recommended, because it can introduce errors and issues for applications that use the Access Database Engine bit version you installed first.

To use both Access Database Engine bit versions:

1. [Install both bit versions of the Access Database Engine](#) from the download page.
2. Run each version of the Access Database Engine by using the `/passive` switch. For example:

Console

```
c:\users\joe\downloads\AccessDatabaseEngine.exe /passive
c:\users\joe\downloads\AccessDatabaseEngine_x64.exe /passive
```

## You use Access or XLS files with Microsoft 365

Office 2013 and Office 2016 Microsoft 365 subscriptions register the Access Database Engine provider in a virtual registry location that only Microsoft Office processes can access. The Mashup Engine, which is responsible for running non-Microsoft 365 Excel and Power BI Desktop, isn't an Office process, so it can't use the Access Database Engine provider.

To fix this situation, download and install the [Access Database Engine Redistributable](#) that matches the bit version of your Power BI Desktop installation, 32-bit or 64-bit.

## Other import issues

The Power BI team regularly looks for issues that might affect many users, and tries to include them in documentation. If you encounter an issue that this article doesn't cover, submit a question about the issue to [Power BI Support](#).

# Troubleshoot DirectQuery models in Power BI Desktop

Article • 02/28/2023

This article helps you diagnose performance issues with Power BI DirectQuery data models you develop in Power BI Desktop or the Power BI service. The article also describes how to get detailed information to help you optimize reports.

You should start any diagnosis of performance issues in Power BI Desktop, rather than in the Power BI service or Power BI Report Server. Performance issues often depend on the performance level of the underlying data source. You can more easily identify and diagnose these issues in the isolated Power BI Desktop environment, without involving components like an on-premises gateway.

If you don't find the performance issues in Power BI Desktop, you can focus your investigation on the specifics of the report in the Power BI service.

You should also try to isolate issues to an individual visual before you look at many visuals on a page.

## Performance Analyzer

[Performance Analyzer](#) is a useful tool for identifying performance issues throughout the troubleshooting process. If you can identify a single sluggish visual on a page in Power BI Desktop, you can use Performance Analyzer to determine what queries Power BI Desktop sends to the underlying source.

You also might be able to view traces and diagnostic information that the underlying data sources emit. Such traces can contain useful information about the details of how the query executed, and how to improve it.

Even without traces from the source, you can view the queries Power BI sent, along with their execution times.

### Note

For DirectQuery SQL-based sources, Performance Analyzer shows queries only for SQL Server, Oracle, and Teradata data sources.

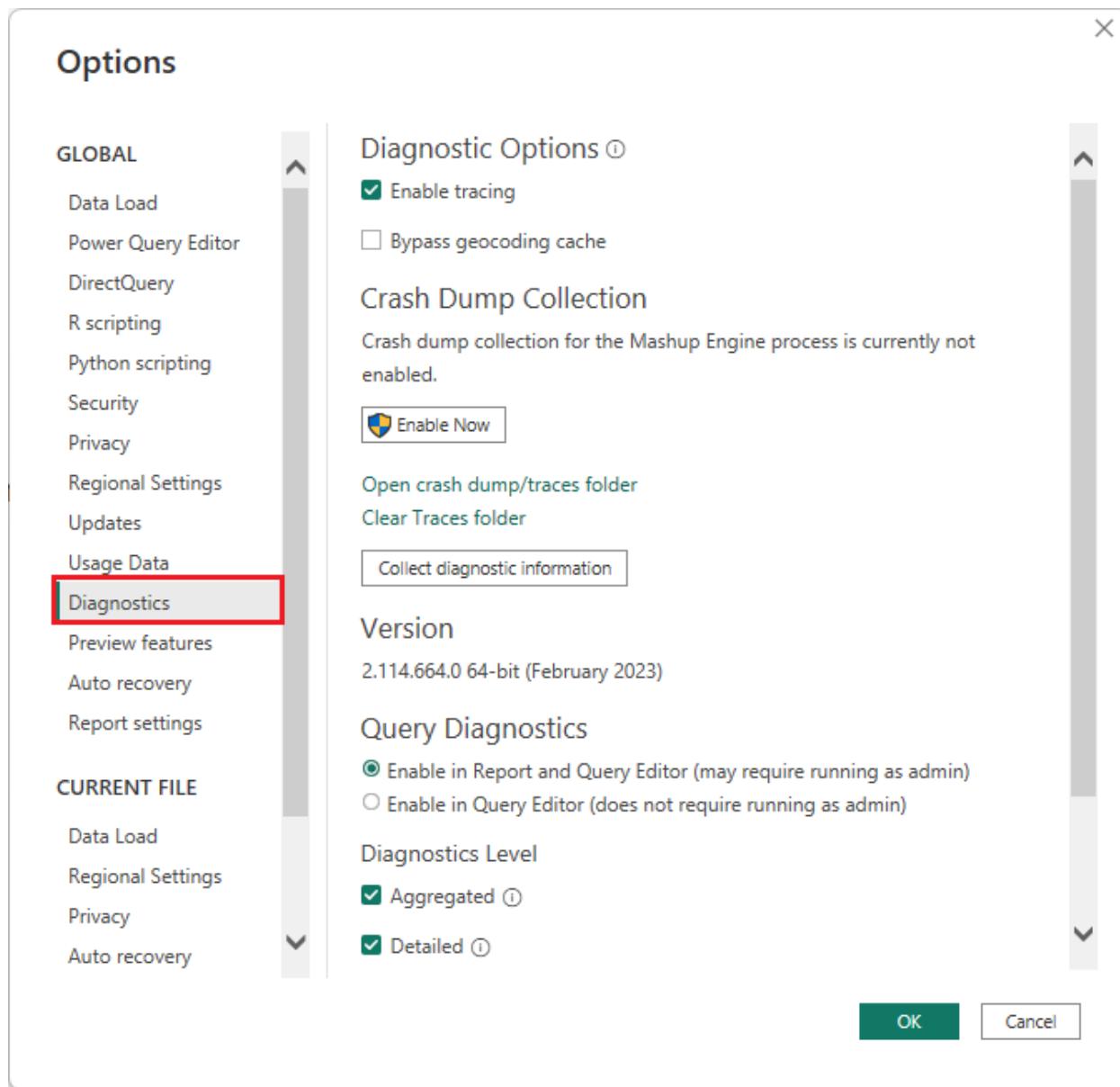
# Trace file

By default, Power BI Desktop logs events during a given session to a trace file called *FlightRecorderCurrent.trc*. You can find the trace file for the current session in the *AppData* folder for the current user, at *<User>\AppData\Local\Microsoft\Power BI Desktop\AnalysisServicesWorkspaces*.

The following DirectQuery data sources write all the queries that Power BI sends them to the trace file. The log might support other DirectQuery sources in the future.

- SQL Server
- Azure SQL Database
- Azure Synapse Analytics (formerly SQL Data Warehouse)
- Oracle
- Teradata
- SAP HANA

To easily get to the trace file folder in Power BI Desktop, select **File > Options and settings > Options**, and then select **Diagnostics**.



Under **Crash Dump Collection**, select the [Open crash dump/traces folder](#) link to open the `<User>\AppData\Local\Microsoft\Power BI Desktop\Traces` folder.

Navigate to that folder's parent folder, and then open the `AnalysisServicesWorkspaces` folder, which contains one workspace subfolder for every open instance of Power BI Desktop. The subfolder names have integer suffixes, such as `AnalysisServicesWorkspace2058279583`.

Each `AnalysisServicesWorkspace` folder includes a `Data` subfolder that contains the trace file `FlightRecorderCurrent.trc` for the current Power BI session. This folder disappears when the associated Power BI Desktop session ends.

You can open the trace files by using the SQL Server Profiler tool, which you can get as part of the free [SQL Server Management Studio \(SSMS\) download](#). After you download and install SQL Server Management Studio, open SQL Server Profiler.

SQL Server Profiler - [C:\Data\FlightRecorderCurrent.trc]

The screenshot shows the SQL Server Profiler application window. The title bar reads "SQL Server Profiler - [C:\Data\FlightRecorderCurrent.trc]". The menu bar includes File, Edit, View, Replay, Tools, Window, Help. The toolbar has icons for Open, Save, Print, and others. The main area is a large table with columns: EventClass, EventSubClass, CurrentTime, StartTime, C, Database, NTUser, NID, Os, ApplicationName, SessionID, NTConn, SP, TextData, ServerName, R, RequestID, AdhvID, EndTime, Duration, C, L, S, Error, E. The table contains numerous rows of event data, primarily related to Power BI sessions (e.g., Command, DirectQuery, Query). The last few rows show DAX query events starting with "3 - DAXQuery". The bottom of the table shows some DDL definitions for variables and a note about Done.

```

EventClass EventSubClass CurrentTime StartTime C Database NTUser NID Os ApplicationName SessionID NTConn SP TextData ServerName R RequestID AdhvID EndTime Duration C L S Error E
Command End 12 - Batch 2022-11-17 ... 6 3b8... RED... R... 9824 Power BI... ED11... RED... 14 <batch Transaction="f..." LAPTOP-... <Prop... F34F7E800-64C1-478D... CCC1... 2022-11-17 17:43... 5 0 13 1
Command Begin 12 - Batch 2022-11-17 ... 81 3b8... RED... R... 9824 Power BI... A0CF... RED... 301 <batch Transaction="f..." LAPTOP-... <Prop... 733BF6D0-8267-45B6... 30A9... 2022-11-17 17:43... 4 0 13 1
Command End 12 - Batch 2022-11-17 ... 6 3b8... RED... R... 9824 Power BI... A0CF... RED... 301 <batch Transaction="f..." LAPTOP-... <Prop... F94F6880-64C1-478D... 2DE9A... 2022-11-17 17:43... 8 0 13 1
Command End 12 - Batch 2022-11-17 ... 6 3b8... RED... R... 9824 Power BI... ED11... RED... 14 <batch Transaction="f..." LAPTOP-... <Prop... E87E3039-A413-4E8E... B3188... 2022-11-17 17:43... 8 0 13 1
Command End 12 - Batch 2022-11-17 ... 84 3b8... RED... R... 9824 Power BI... BDAC... RED... 111 <batch Transaction="f..." LAPTOP-... <Prop... E2E72030-AA13-4E8E... B3188... 2022-11-17 17:43... 5 0 13 1
Query Begin 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 DEFINE VAR __DSOCO... LAPTOP-... <Prop... D710944C-40CF-4420... 57905... 2022-11-17 17:44... 50 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (62) [...] LAPTOP-... <Prop... D710944C-40CF-4420... 57905... 2022-11-17 17:44... 50 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (62) [...] LAPTOP-... <Prop... D710944C-40CF-4420... 57905... 2022-11-17 17:44... 50 0 1 0
Query Begin 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001) [...] LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 0 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001) [...] LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 51 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT COUNT_BIG(DI...) LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 0 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT COUNT_BIG(DI...) LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 50 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001) [...] LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 0 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001) [...] LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 0 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1000001) [...] LAPTOP-... <Prop... 400BDAA4-0A8E-40CC... AD247... 2022-11-17 17:44... 0 0 1 0
Query End 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE VAR __DSOCO... LAPTOP-... <Prop... D710944C-40CF-4420... AD247... 2022-11-17 17:44... 276 0 0 1
Command Begin 12 - Batch 2022-11-17 ... 88 3b8... RED... R... 9824 Power BI... A090... RED... 328 <batch Transaction="f..." LAPTOP-... <Prop... FC1D0458-5DA2-49F6... 8850F... 2022-11-17 17:44... 3 0 13 1
Command End 12 - Batch 2022-11-17 ... 88 3b8... RED... R... 9824 Power BI... A090... RED... 328 <batch Transaction="f..." LAPTOP-... <Prop... 8850F... 2022-11-17 17:44... 3 0 13 1
Command Begin 12 - Batch 2022-11-17 ... 92 3b8... RED... R... 9824 Power BI... A463... RED... 318 <batch Transaction="f..." LAPTOP-... <Prop... D004118A-4C33-4E33... B0281... 2022-11-17 17:44... 4 0 13 1
Command End 12 - Batch 2022-11-17 ... 91 3b8... RED... R... 9824 Power BI... A463... RED... 318 <batch Transaction="f..." LAPTOP-... <Prop... D004118A-4C33-4E33... B0281... 2022-11-17 17:44... 4 0 13 1
Query Begin 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE VAR __DSOCO... LAPTOP-... <Prop... 2E53511B-21FC-407E... 05B08... 2022-11-17 17:44... 16 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 SELECT TOP (1001) [...] LAPTOP-... <Prop... 2E53511B-21FC-407E... 05B08... 2022-11-17 17:45... 52 16 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE VAR __DSOCO... LAPTOP-... <Prop... 2E53511B-21FC-407E... 05B08... 2022-11-17 17:45... 69 16 0 1 0
Query End 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE VAR __DSOCO... LAPTOP-... <Prop... 2E53511B-21FC-407E... 05B08... 2022-11-17 17:45... 69 16 0 1 0
Query Begin 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE SUM([T1].[SUB... LAPTOP-... <Prop... 2E44A56F-6200-40E9... D036F... 2022-11-17 17:46... 0 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <SELECT SUM([T1].[SUB... LAPTOP-... <Prop... 2E44A56F-6200-40E9... D036F... 2022-11-17 17:46... 56 0 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <SELECT TOP (1000001)... LAPTOP-... <Prop... 2E44A56F-6200-40E9... D036F... 2022-11-17 17:46... 0 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <SELECT TOP (1000001)... LAPTOP-... <Prop... 2E44A56F-6200-40E9... D036F... 2022-11-17 17:46... 47 0 0 1 0
Query End 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE VAR __DSOCO... LAPTOP-... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:46... 116 0 0 1 0
Query Begin 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE SUM([T1].[SUB... LAPTOP-... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:46... 0 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <SELECT SUM([T1].[SUB... LAPTOP-... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:46... 55 0 0 1 0
DirectQuery Begin 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <SELECT TOP (1000001)... LAPTOP-... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:46... 0 0 1 0
DirectQuery End 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <SELECT TOP (1000001)... LAPTOP-... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:46... 49 0 0 1 0
query End 3 - DAXQuery 2022-11-17 ... 80 3b8... RED... R... 9824 Power BI... D93C... RED... 292 <DEFINE VAR __DSOCO... LAPTOP-... <Prop... 499600C-0154-49B8... 7430B... 2022-11-17 17:46... 116 0 0 1 0

```

Done.

Ln 137, Col 1 | Row 229 | Connections 0

To open a trace file:

1. In SQL Server Profiler, select File > Open > Trace File.
2. Navigate to or enter the path to the trace file for the current Power BI session, such as `<User>\AppData\Local\Microsoft\Power BI Desktop\AnalysisServicesWorkspaces\AnalysisServicesWorkspace2058279583\Data`, and open `FlightRecorderCurrent.trc`.

SQL Server Profiler displays all events from the current session. The following screenshot highlights a group of events for a query. Each query group has the following events:

- A `Query Begin` and `Query End` event, which represent the start and end of a DAX query generated by changing a visual or filter in the Power BI UI, or from filtering or transforming data in the Power Query Editor.
- One or more pairs of `DirectQuery Begin` and `DirectQuery End` events, which represent queries sent to the underlying data source as part of evaluating the DAX query.

SQL Server Profiler - [C:\Data\FlightRecorderCurrent.trc]

File Edit View Replay Tools Window Help

EventClass EventSubClass CurrentTime StartTime C Database NTJobID ApplicationName SessionID NTConnID SP\_TextData ServerName R\_ RequestID RequestID EndTime Duration C L S Error

EventClass	EventSubClass	CurrentTime	StartTime	C	Database	NTJobID	ApplicationName	SessionID	NTConnID	SP_TextData	ServerName	R_	RequestID	RequestID	EndTime	Duration	C	L	S	Error
Command End	12 - Batch	2022-11-17 ...	6 3b8... RED...	R....	9824 Power BI...	ED11...	RED...	14	<batch Transaction=1>,...	LAPTOP-...	F34FEB08-E4C1-478D...	CCCL...	2022-11-17 17:43...	\$ 0 13 1						
Command Begin	12 - Batch	2022-11-17 ...	6 3b8... RED...	R....	9824 Power BI...	A0CF...	RED...	301	<batch Transaction=1>,...	LAPTOP-...	<Prop...>	738FB0D8-B267-45B6...	30A9...	2022-11-17 17:43...	4 0 13 1					
Command End	12 - Batch	2022-11-17 ...	6 3b8... RED...	R....	9824 Power BI...	A0CF...	RED...	301	<batch Transaction=1>,...	LAPTOP-...	<Prop...>	F94FE88E-E4C1-478D...	20E9A...	2022-11-17 17:43...	8 0 13 1					
Command End	12 - Batch	2022-11-17 ...	6 3b8... RED...	R....	9824 Power BI...	ED11...	RED...	14	<batch Transaction=1>,...	LAPTOP-...	<Prop...>	E87E3039-AA13-4E8E...	B138E...	2022-11-17 17:43...	8 0 13 1					
Command End	12 - Batch	2022-11-17 ...	6 3b8... RED...	R....	9824 Power BI...	EDAC...	RED...	111	<batch Transaction=1>,...	LAPTOP-...	<Prop...>	E2E72039-AA13-4E8E...	B138E...	2022-11-17 17:43...	8 0 13 1					
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	F710944C-40CF-4420...	57905...	2022-11-17 17:44...	50 0 1 0					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (62) [1]...	LAPTOP-...	<Prop...>	D710944C-40CF-4420...	57905...	2022-11-17 17:44...	50 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFIN...	LAPTOP-...	<Prop...>	D710944C-40CF-4420...	57905...	2022-11-17 17:44...	50 0 1 0					
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	50 0 1 0					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (62) [1]...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	50 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT COUNT(BIGD1I...)...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	50 0 1 0					
Query Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT COUNT(BIGD1I...)...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	50 0 1 0					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (100001) [1]...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	51 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (100001) [1]...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	51 0 1 0					
Query End	3 - DAXQuery	2022-11-17 ...	80 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	40BDAA4A-0A8E-40CC...	A0247...	2022-11-17 17:44...	47 0 1 0					
Command Begin	12 - Batch	2022-11-17 ...	88 3b8... RED...	R....	9824 Power BI...	O490...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	FC1D045B-5DA2-49F6...	8850F...	2022-11-17 17:44...	3 0 13 1					
Command End	12 - Batch	2022-11-17 ...	88 3b8... RED...	R....	9824 Power BI...	O490...	RED...	292	<batch Transaction=1>,...	LAPTOP-...	<Prop...>	DD04158A-4C33-4E33...	B0281...	2022-11-17 17:44...	4 0 13 1					
Command Begin	12 - Batch	2022-11-17 ...	91 3b8... RED...	R....	9824 Power BI...	4463...	RED...	338	<batch Transaction=1>,...	LAPTOP-...	<Prop...>	DD04158A-4C33-4E33...	B0281...	2022-11-17 17:44...	4 0 13 1					
Command End	12 - Batch	2022-11-17 ...	91 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:44...	16 0 1 0					
Query Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (1001) [1]...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:45...	52 16 0 1 0					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT SUM([T1].[Sub])...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:45...	69 16 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:45...	69 16 0 1 0					
Query Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT SUM([T1].[Sub])...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:45...	69 16 0 1 0					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (100001) [1]...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:45...	56 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (100001) [1]...	LAPTOP-...	<Prop...>	2E53511B-21FC-407E...	05B08...	2022-11-17 17:45...	47 0 1 0					
Query Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	116 0 0 1					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT SUM([T1].[Sub])...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	0 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT SUM([T1].[Sub])...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	55 0 1 0					
Query Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (100001) [1]...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	0 0 1 0					
DirectQuery Begin	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT TOP (100001) [1]...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	49 0 1 0					
DirectQuery End	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	DEFINE VAR __DSOCO...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	116 0 0 1					
Query End	3 - DAXQuery	2022-11-17 ...	90 3b8... RED...	R....	9824 Power BI...	D93C...	RED...	292	SELECT SUM([T1].[Sub])...	LAPTOP-...	<Prop...>	49K600C-0154-49B8...	7430B...	2022-11-17 17:46...	0 0 1 0					
Define					VAR __DSOCO = SUM( SELECT ProductCategory FROM [SalesLT].[ProductCategory] WHERE ProductCategory.[Name] = @CategoryName );  VAR __DSOPRIMARYBASE = SUMMARIZEE(_ __DSOCO, [SalesLT].[ProductCategory].[ModifiedDate])															
Done.																				

Ln 137, Col 1 | Row 229 | Connections 0

Multiple DAX queries can run in parallel, so events from different groups can interleave. You can use the value of the `ActivityID` to determine which events belong to the same group.

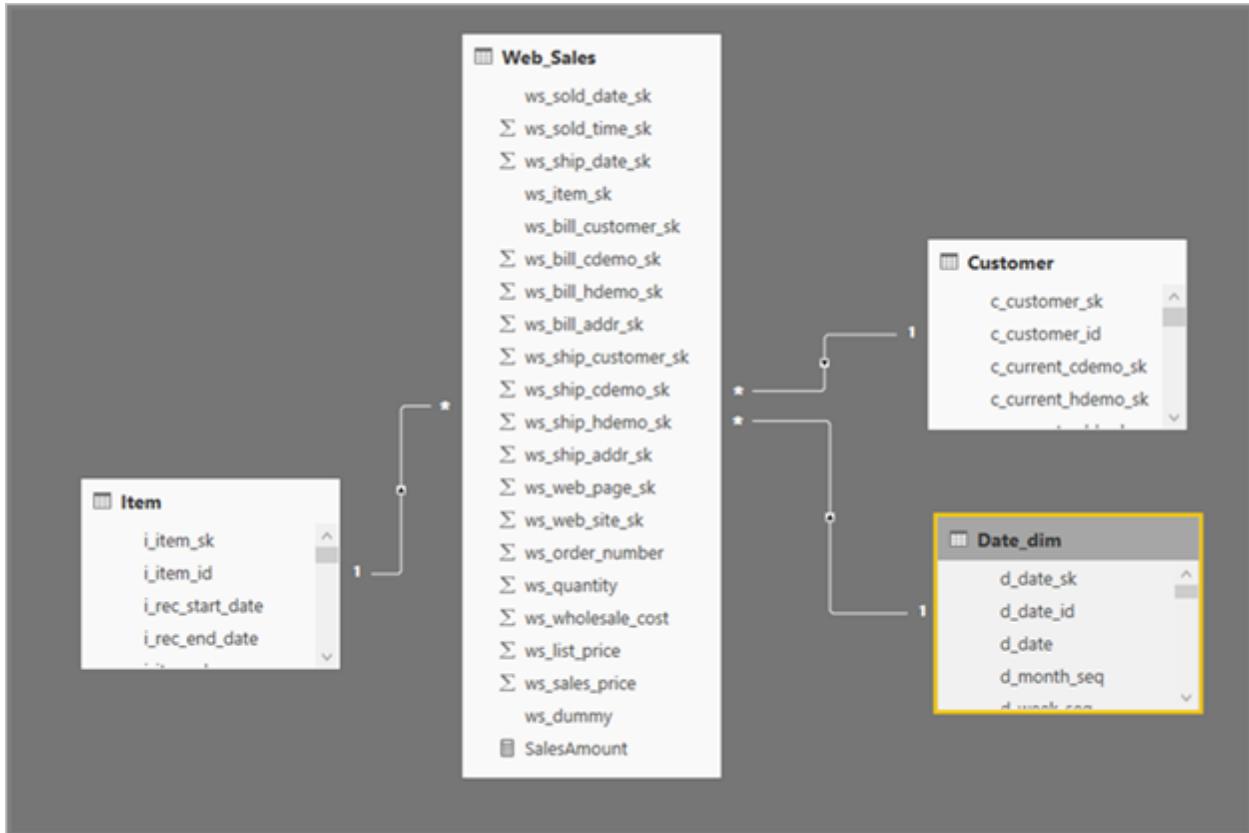
The following columns are also of interest:

- TextData:** The textual detail of the event. For `Query Begin` and `Query End` events, the detail is the DAX query. For `DirectQuery Begin` and `DirectQuery End` events, the detail is the SQL query sent to the underlying source. The `TextData` value for the currently selected event also appears in the pane at the bottom of the screen.
  - EndTime:** The time when the event completed.
  - Duration:** The duration, in milliseconds, it took to run the DAX or SQL query.
  - Error:** Whether an error occurred, in which case the event also displays in red.
- The preceding image narrows some of the less interesting columns, so you can see the more interesting columns more easily.
- Follow this approach to capture a trace to help diagnose a potential performance issue:
- Open a single Power BI Desktop session, to avoid the confusion of multiple workspace folders.
  - Do the set of actions of interest in Power BI Desktop. Include a few more actions, to ensure that the events of interest flush into the trace file.
  - Open SQL Server Profiler and examine the trace. Remember that closing Power BI Desktop deletes the trace file. Also, further actions in Power BI Desktop don't immediately appear. You must close and reopen the trace file to see new events.

Keep individual sessions reasonably small, perhaps 10 seconds of actions, not hundreds. This approach makes it easier to interpret the trace file. There's also a limit on the size of the trace file, so for long sessions, there's a chance of early events dropping.

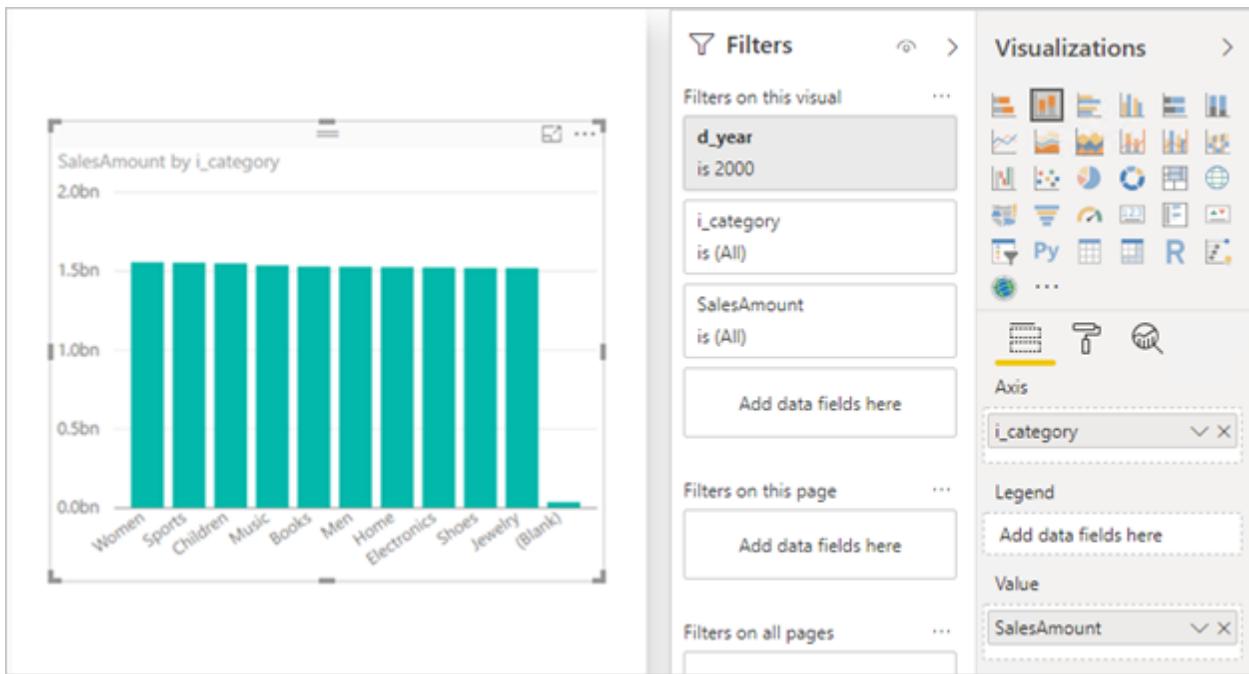
## Query and subquery format

The general format of Power BI Desktop queries is to use subqueries for each model table the queries reference. The Power Query Editor query defines the subselect queries. For example, assume you have the following [TPC-DS](#) tables in a SQL Server relational database:



In the Power BI visual, the following expression defines the `SalesAmount` measure:

```
DAX
SalesAmount = SUMX(Web_Sales, [ws_sales_price] * [ws_quantity])
```



Refreshing the visual produces the T-SQL query in the following image. There are three subqueries for the `Web_Sales`, `Item`, and `Date_dim` model tables. Each query returns all the model table columns, even though the visual references only four columns.

These shaded subqueries are the exact definition of the Power Query queries. This use of subqueries doesn't affect performance for the data sources DirectQuery supports. Data sources like SQL Server optimize away the references to the other columns.

One reason Power BI uses this pattern is so you can define a Power Query query to use a specific query statement. Power BI uses the query as provided, without an attempt to rewrite it. This pattern restricts using query statements that use Common Table Expressions (CTEs) and stored procedures. You can't use these statements in subqueries.

```

SELECT
TOP (1000000) [c13],SUM([a0])
AS [a0]
FROM
(

```

  SELECT [t0].[i\_category] AS [c13],[t1].[ws\_quantity] AS [c40],[t1].[ws\_sales\_price] AS  
[c43],[t3].[d\_year] AS [c72],  
([t1].[ws\_sales\_price] \* [t1].[ws\_quantity])  
AS [a0]  
FROM  
(  
(select [\$Table].[ws\_sold\_date\_sk] as [ws\_sold\_date\_sk],  
[\$Table].[ws\_sold\_time\_sk] as [ws\_sold\_time\_sk],  
[\$Table].[ws\_ship\_date\_sk] as [ws\_ship\_date\_sk],  
[\$Table].[ws\_item\_sk] as [ws\_item\_sk],  
[\$Table].[ws\_bill\_customer\_sk] as [ws\_bill\_customer\_sk],  
[\$Table].[ws\_bill\_cdemo\_sk] as [ws\_bill\_cdemo\_sk],  
[\$Table].[ws\_bill\_hdemo\_sk] as [ws\_bill\_hdemo\_sk],  
... 8 other columns  
[\$Table].[ws\_quantity] as [ws\_quantity],  
[\$Table].[ws\_wholesale\_cost] as [ws\_wholesale\_cost],  
[\$Table].[ws\_list\_price] as [ws\_list\_price],  
[\$Table].[ws\_sales\_price] as [ws\_sales\_price],  
[\$Table].[ws\_dummy] as [ws\_dummy]  
from [dbo].[Web\_Sales] as [\$Table]) AS [t1]

left outer join

(select [\$Table].[i\_item\_sk] as [i\_item\_sk],  
[\$Table].[i\_item\_id] as [i\_item\_id],  
[\$Table].[i\_rec\_start\_date] as [i\_rec\_start\_date],  
[\$Table].[i\_rec\_end\_date] as [i\_rec\_end\_date],  
[\$Table].[i\_item\_desc] as [i\_item\_desc],  
[\$Table].[i\_current\_price] as [i\_current\_price],  
[\$Table].[i\_wholesale\_cost] as [i\_wholesale\_cost],  
[\$Table].[i\_brand\_id] as [i\_brand\_id],  
[\$Table].[i\_brand] as [i\_brand],  
[\$Table].[i\_class\_id] as [i\_class\_id],  
[\$Table].[i\_class] as [i\_class],  
[\$Table].[i\_category\_id] as [i\_category\_id],  
[\$Table].[i\_category] as [i\_category],  
... 8 other columns  
[\$Table].[i\_product\_name] as [i\_product\_name],  
[\$Table].[i\_dummy] as [i\_dummy]  
from [dbo].[Item] as [\$Table]) AS [t0] on  
(  
[t1].[ws\_item\_sk] = [t0].[i\_item\_sk]  
)  
)

left outer join

(select [\$Table].[d\_date\_sk] as [d\_date\_sk],  
[\$Table].[d\_date\_id] as [d\_date\_id],  
[\$Table].[d\_date] as [d\_date],  
[\$Table].[d\_month\_seq] as [d\_month\_seq],  
[\$Table].[d\_week\_seq] as [d\_week\_seq],  
[\$Table].[d\_quarter\_seq] as [d\_quarter\_seq],  
[\$Table].[d\_year] as [d\_year],  
[\$Table].[d\_dow] as [d\_dow],  
[\$Table].[d\_moy] as [d\_moy],  
[\$Table].[d\_dom] as [d\_dom],  
[\$Table].[d\_ooy] as [d\_ooy],  
... 16 other columns  
[\$Table].[d\_current\_year] as [d\_current\_year],  
[\$Table].[d\_dummy] as [d\_dummy]  
from [dbo].[Date\_dim] as [\$Table]) AS [t3] on  
(  
[t1].[ws\_sold\_date\_sk] = [t3].[d\_date\_sk]  
)  
)

)

AS [t0]  
WHERE  
(  
[c72] = 2000  
)  
GROUP BY [c13]

## Gateway performance

For information about troubleshooting gateway performance, see [Troubleshoot gateways - Power BI](#).

# Next steps

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

- [Use DirectQuery in Power BI Desktop](#)
- [Data sources supported by DirectQuery](#)
- [DirectQuery models in Power BI Desktop](#)
- [DirectQuery model guidance in Power BI Desktop](#)

Questions? Try asking the Power BI Community 

# Troubleshooting nested values returned as text in the Power BI service

Article • 02/10/2023

In the past, there have been cases where a report refreshes in Power BI Desktop, but fails on the Power BI service with an error like this text:

Output

We cannot convert the value "[Table]" to type Table

## Cause

One of the causes of this error involves nested non-scalar values, such as tables, records, lists, and functions. When the Data Privacy Firewall buffers a data source, nested non-scalar values are converted to text values, such as "[Table]" or "[Record]".

The Power BI service now supports the setting of privacy levels or turning off the Firewall entirely. The errors can be avoided by [configuring the data source privacy settings](#) in the Power BI service to be *non-Private*.

For more recent versions of Power BI, when the Firewall buffers a nested table, record, or list, it doesn't silently convert non-scalar values to text. Instead, it shows an error:

Output

We cannot return a value of type Table in this context

## Effect on Load/Refresh

This change motivated by Firewall buffering extends to Load/Refresh, as well. The behavior of loading nested tables, records, and lists to the Power BI Model and the Excel Data Model in Power Query for Excel has changed. Before, nested items were loaded as text values, such as "[Table]" or "[Record]". Now, they're treated as errors. A `null` value is in the loaded table and error count increments in the load results.

Since these errors only occur during Load/Refresh, they don't appear in the Power Query Editor.

## Before

- Load/Refresh with no errors
- Loaded table contains "[Table]", "[Record]", and so forth.

## After

- Load/Refresh with errors
- Loaded table contains null, instead of "[Table]", "[Record]", and so forth.

## Resolution

Are you loading a column that contains non-scalar values, for example, tables, lists, or records? If so, you should be able to eliminate the errors by removing the column.

If you can't remove the column, try to replicate the old behavior by adding a custom column and using logic like the following sample:

### Output

```
if [MyColumn] is table then "[Table]" else if [MyColumn] is record then "[Record]"
else if [MyColumn] is list then "[List]" else if [MyColumn] is function
then "[Function]" else [MyColumn]
```

Does the issue reproduce in Power BI Desktop if you set all your data source privacy settings to Private? If so, try to resolve the error by [configuring their data source privacy settings](#) in the Power BI service to be non-Private.

# Data types in Power BI Desktop

Article • 07/12/2023

This article describes data types that Power BI Desktop and Data Analysis Expressions (DAX) support.

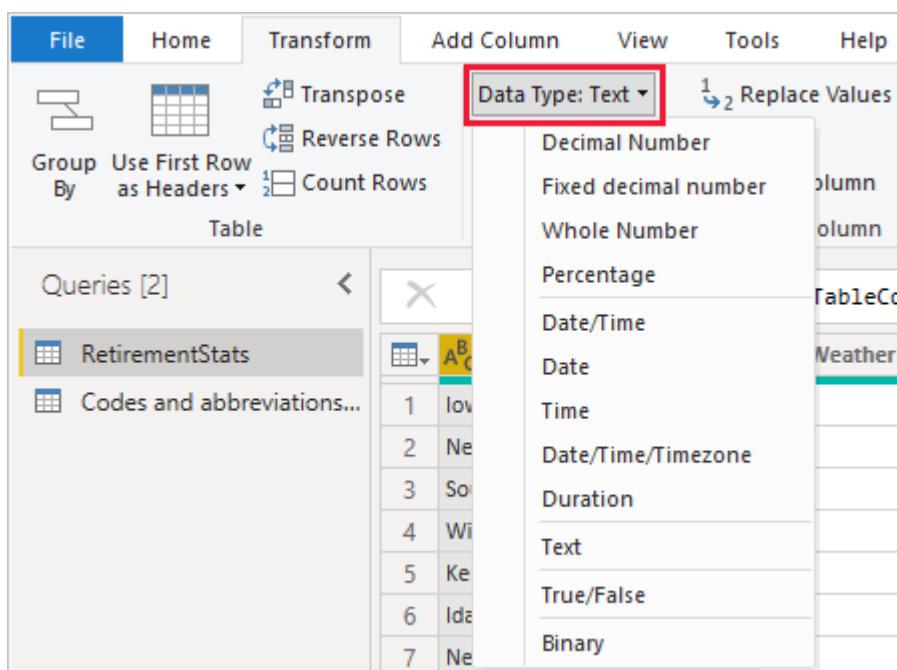
When Power BI loads data, it tries to convert the data types of source columns into data types that support more efficient storage, calculations, and data visualization. For example, if a column of values you import from Excel has no fractional values, Power BI Desktop converts the data column to a **Whole number** data type, which is better suited for storing integers.

This concept is important because some DAX functions have special data type requirements. In many cases DAX [implicitly converts data types](#), but in some cases it doesn't. For instance, if a DAX function requires a **Date** data type, but the data type for your column is **Text**, the DAX function won't work correctly. So it's important and useful to use the correct data types for columns.

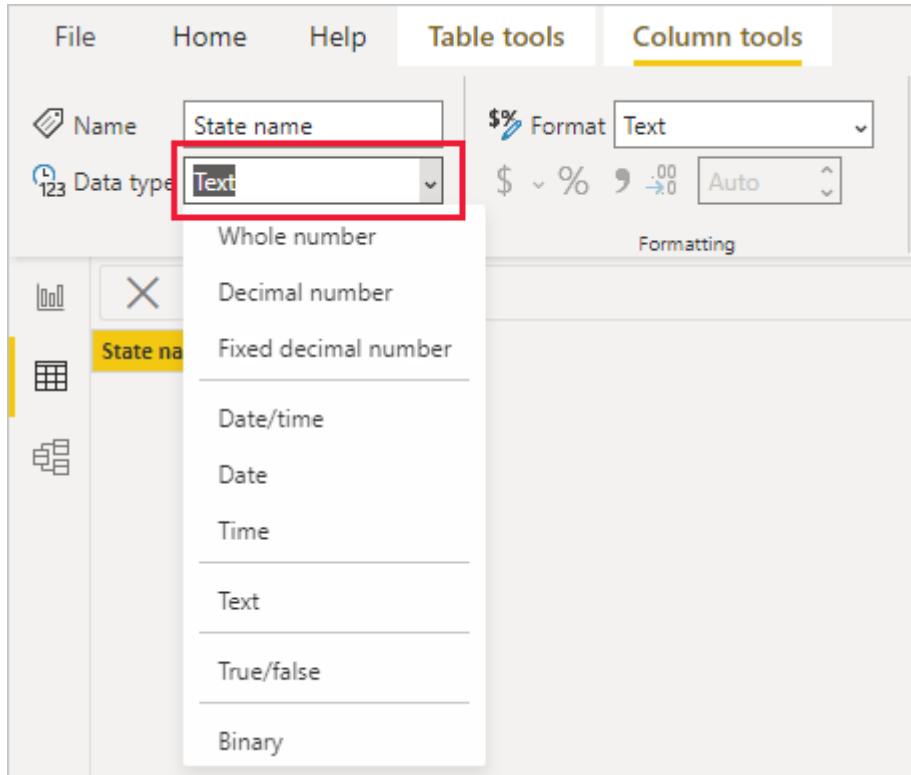
## Determine and specify a column's data type

In Power BI Desktop, you can determine and specify a column's data type in the Power Query Editor, in Data View, or in Report View:

- In Power Query Editor, select the column and then select **Data Type** in the **Transform** group of the ribbon.



- In Data View or Report View, select the column, and then select the dropdown arrow next to **Data type** on the **Column tools** tab of the ribbon.



The Data Type dropdown selection in Power Query Editor has two data types not present in Data View or Report View: **Date/Time/Timezone** and **Duration**. When you load a column with these data types into the Power BI model, a **Date/Time/Timezone** column converts into a **Date/time** data type, and a **Duration** column converts into a **Decimal number** data type.

The **Binary** data type isn't supported outside of the Power Query Editor. In the Power Query Editor, you can use the **Binary** data type when you load binary files if you convert it to other data types before loading it into the Power BI model. The **Binary** selection exists in the Data View and Report View menus for legacy reasons, but if you try to load **Binary** columns into the Power BI model, you might run into errors.

## Number types

Power BI Desktop supports three number types: **Decimal number**, **Fixed decimal number**, and **Whole number**.

You can use the Tabular Object Model (TOM) Column [DataType](#) property to specify the [DataType](#) Enums for number types. For more information about programmatically modifying objects in Power BI, see [Program Power BI datasets with the Tabular Object Model](#).

## Decimal number

**Decimal number** is the most common number type, and can handle numbers with fractional values and whole numbers. **Decimal number** represents 64-bit (eight-byte) floating point numbers with negative values from  $-1.79E +308$  through  $-2.23E -308$ , positive values from  $2.23E -308$  through  $1.79E +308$ , and 0. Numbers like 34, 34.01, and 34.000367063 are valid decimal numbers.

The highest precision that the **Decimal number** type can represent is 15 digits. The decimal separator can occur anywhere in the number. This type corresponds to how Excel stores its numbers, and TOM specifies this type as `DataType.Double` Enum.

## Fixed decimal number

The **Fixed decimal number** data type has a fixed location for the decimal separator. The decimal separator always has four digits to its right, and allows for 19 digits of significance. The largest value the **Fixed decimal number** can represent is positive or negative 922,337,203,685,477.5807.

The **Fixed decimal number** type is useful in cases where rounding might introduce errors. Numbers that have small fractional values can sometimes accumulate and force a number to be slightly inaccurate. The **Fixed decimal number** type can help you avoid these kinds of errors by truncating the values past the four digits to the right of decimal separator.

This data type corresponds to SQL Server's **Decimal (19,4)**, or the **Currency** data type in Analysis Services and Power Pivot in Excel. TOM specifies this type as `DataType.Decimal` Enum.

## Whole number

**Whole number** represents a 64-bit (eight-byte) integer value. Because it's an integer, **Whole number** has no digits to the right of the decimal place. This type allows for 19 digits of positive or negative whole numbers between  $-9,223,372,036,854,775,807$  ( $-2^{63}+1$ ) and  $9,223,372,036,854,775,806$  ( $2^{63}-2$ ), so can represent the largest possible numbers of the numeric data types.

As with the **Fixed decimal** type, the **Whole number** type can be useful when you need to control rounding. TOM represents the **Whole number** data type as `DataType.Int64` Enum.

## Note

The Power BI Desktop data model supports 64-bit integer values, but due to JavaScript limitations, the largest number Power BI visuals can safely express is 9,007,199,254,740,991 ( $2^{53}-1$ ). If your data model has larger numbers, you can reduce their size through calculations before you add them to visuals.

## Accuracy of number type calculations

Column values of **Decimal number** data type are stored as *approximate* data types, according to the IEEE 754 Standard for floating point numbers. Approximate data types have inherent precision limitations, because instead of storing exact number values, they may store extremely close, or rounded, approximations.

Precision loss, or *imprecision*, can occur if the floating-point value can't reliably quantify the number of floating point digits. Imprecision can potentially appear as unexpected or inaccurate calculation results in some reporting scenarios.

Equality-related comparison calculations between values of **Decimal number** data type can potentially return unexpected results. Equality comparisons include equals `=`, greater than `>`, less than `<`, greater than or equal to `>=`, and less than or equal to `<=`.

This issue is most apparent when you use the [RANKX function](#) in a DAX expression, which calculates the result twice, resulting in slightly different numbers. Report users might not notice the difference between the two numbers, but the rank result can be noticeably inaccurate. To avoid unexpected results, you can change the column data type from **Decimal number** to either **Fixed decimal number** or **Whole number**, or do a forced rounding by using [ROUND](#). The **Fixed decimal number** data type has greater precision, because the decimal separator always has four digits to its right.

Rarely, calculations that sum the values of a column of **Decimal number** data type can return unexpected results. This result is most likely with columns that have large amounts of both positive numbers and negative numbers. The sum result is affected by the distribution of values across rows in the column.

If a required calculation sums most of the positive numbers before summing most of the negative numbers, the large positive partial sum at the beginning can potentially skew the results. If the calculation happens to add balanced positive and negative numbers, the query retains more precision, and therefore returns more accurate results. To avoid unexpected results, you can change the column data type from **Decimal number** to **Fixed decimal number** or **Whole number**.

# Date/time types

Power BI Desktop supports five **Date/Time** data types in Power Query Editor. Both **Date/Time/Timezone** and **Duration** convert during load into the Power BI Desktop data model. The model supports **Date/Time**, or you can format the values as **Date** or **Time** independently.

- **Date/Time** represents both a date and time value. The underlying **Date/Time** value is stored as a **Decimal number** type, so you can convert between the two types. The time portion stores as a fraction to whole multiples of 1/300 seconds (3.33 ms). The data type supports dates between years 1900 and 9999.
- **Date** represents just a date with no time portion. A **Date** converts into the model as a **Date/Time** value with zero for the fractional value.
- **Time** represents just a time with no date portion. A **Time** converts into the model as a **Date/Time** value with no digits to the left of the decimal point.
- **Date/Time/Timezone** represents a UTC date/time with a timezone offset, and converts into **Date/Time** when loaded into the model. The Power BI model doesn't adjust the timezone based on a user's location or locale. A value of 09:00 loaded into the model in the USA displays as 09:00 wherever the report is opened or viewed.
- **Duration** represents a length of time, and converts into a **Decimal Number** type when loaded into the model. As **Decimal Number** type, you can add or subtract the values from **Date/Time** values with correct results, and easily use the values in visualizations that show magnitude.

# Text type

The **Text** data type is a Unicode character data string, which can be letters, numbers, or dates represented in a text format. The practical maximum limit for string length is approximately 32,000 Unicode characters, based on Power BI's underlying Power Query engine, and its limits on **text** data type lengths. Text data types beyond the practical maximum limit are likely to result in errors.

The way Power BI stores text data can cause the data to display differently in certain situations. The next sections describe common situations that can cause **Text** data to change appearance slightly between querying data in Power Query Editor and loading it into Power BI.

## Case sensitivity

The engine that stores and queries data in Power BI is *case insensitive*, and treats different capitalization of letters as the same value. "A" is equal to "a". However, Power Query is *case sensitive*, where "A" isn't the same as "a". The difference in case sensitivity can lead to situations where text data changes capitalization seemingly inexplicably after loading into Power BI.

The following example shows order data: An **OrderNo** column that's unique for each order, and an **Addressee** column that shows the addressee name entered manually at order time. Power Query Editor shows several orders with the same **Addressee** names entered into the system with varying capitalizations.

	OrderNo	Addressee
1	1001	Taina Hasu
2	1002	Ellis Turner
3	1003	MURALI DAS
4	1004	TAINA HASU
5	1005	Taina HASU
6	1006	Ellis TURNER
7	1007	Murali Das

After Power BI loads the data, capitalization of the duplicate names in the **Data** tab changes from the original entry into one of the capitalization variants.

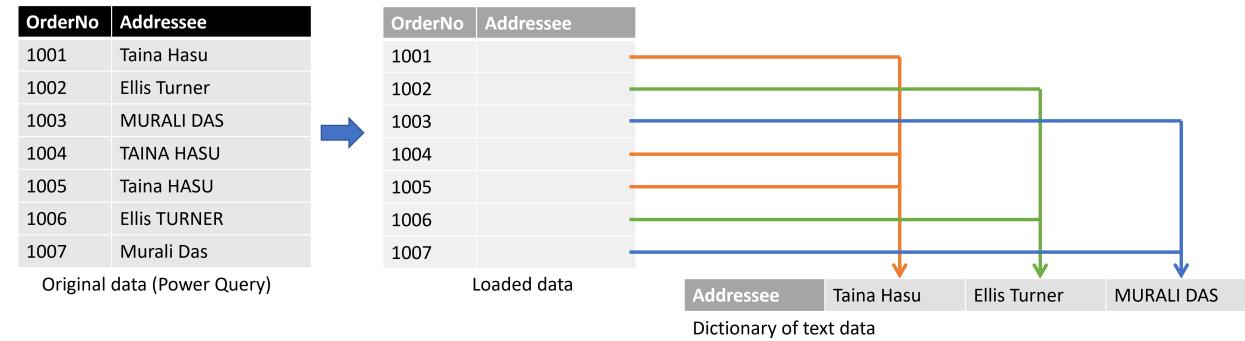
OrderNo	Addressee
1001	Taina Hasu
1002	Ellis Turner
1003	MURALI DAS
1004	Taina Hasu
1005	Taina Hasu
1006	Ellis Turner
1007	MURALI DAS

This change happens because Power Query Editor is case sensitive, so it shows the data exactly as stored in the source system. The engine that stores data in Power BI is case insensitive, so treats the lowercase and uppercase versions of a character as identical. Power Query data loaded into the Power BI engine can change accordingly.

The Power BI engine evaluates each row individually when it loads data, starting from the top. For each text column, such as **Addressee**, the engine stores a dictionary of unique values, to improve performance through data compression. The engine sees the first three values in the **Addressee** column as unique and stores them in the dictionary. After that, because the engine is case insensitive, it evaluates the names as identical.

The engine sees the name "Taina Hasu" as identical to "TAINA HASU" and "Taina HASU", so it doesn't store those variations, but refers to the first variation it stored. The name "MURALI DAS" appears in uppercase letters, because that's how the name appeared the first time the engine evaluated it when loading the data from top to bottom.

This image illustrates the evaluation process:



In the preceding example, the Power BI engine loads the first row of data, creates the **Addressee** dictionary, and adds *Taina Hasu* to it. The engine also adds a reference to that value in the **Addressee** column on the table it loads. The engine does the same for the second and third rows, because these names aren't equivalent to the others when ignoring case.

For the fourth row, the engine compares the value against the names in the dictionary and finds the name. Since the engine is case insensitive, "TAINA HASU" and "Taina Hasu" are the same. The engine doesn't add a new name to the dictionary, but refers to the existing name. The same process happens for the remaining rows.

### ⓘ Note

Because the engine that stores and queries data in Power BI is case insensitive, take special care when you work in DirectQuery mode with a case-sensitive source.

Power BI assumes that the source has eliminated duplicate rows. Because Power BI is case insensitive, it treats two values that differ only by case as duplicate, whereas the source might not treat them as such. In such cases, the final result is undefined.

To avoid this situation, if you use DirectQuery mode with a case-sensitive data source, normalize casing in the source query or in Power Query Editor.

## Leading and trailing spaces

The Power BI engine automatically trims any trailing spaces that follow text data, but doesn't remove leading spaces that precede the data. To avoid confusion, when you work with data that contains leading or trailing spaces, you should use the `Text.Trim`

function to remove spaces at the beginning or end of the text. If you don't remove leading spaces, a relationship might fail to create because of duplicate values, or visuals might return unexpected results.

The following example shows data about customers: a **Name** column that contains the name of the customer and an **Index** column that's unique for each entry. The names appear within quotes for clarity. The customer name repeats four times, but each time with different combinations of leading and trailing spaces. These variations can occur with manual data entry over time.

Row	Leading space	Trailing space	Name	Index	Text length
1	No	No	"Dylan Williams"	1	14
2	No	Yes	"Dylan Williams "	10	15
3	Yes	No	" Dylan Williams"	20	15
4	Yes	Yes	" Dylan Williams "	40	16

In Power Query Editor, the resulting data appears as follows.

	Name	Index	Length
1	Dylan Williams	1	14
2	Dylan Williams	10	15
3	Dylan Williams	20	15
4	Dylan Williams	40	16

When you go to the **Data** tab in Power BI after you load the data, the same table looks like the following image, with the same number of rows as before.

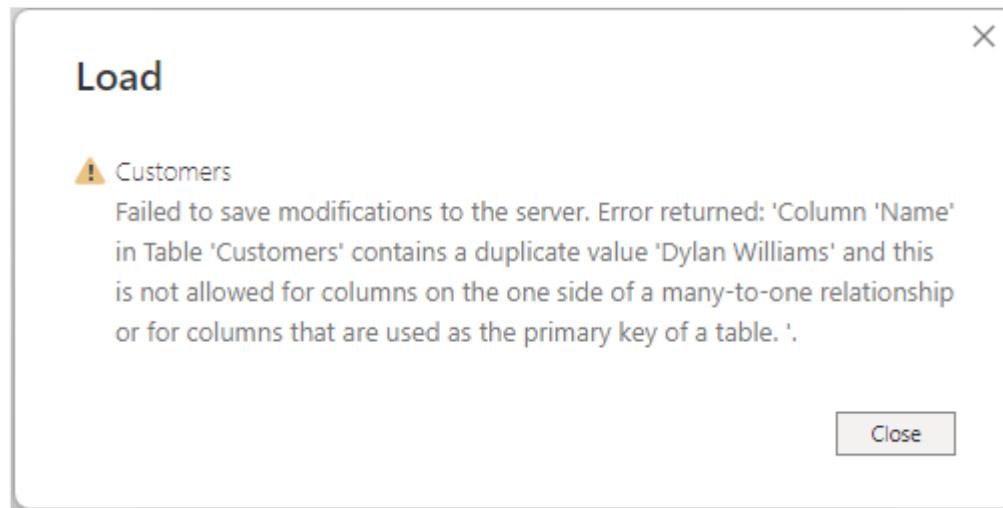
Name	Length	Index
Dylan Williams	14	1
Dylan Williams	15	10
Dylan Williams	15	20
Dylan Williams	16	40

However, a visual based on this data returns just two rows.

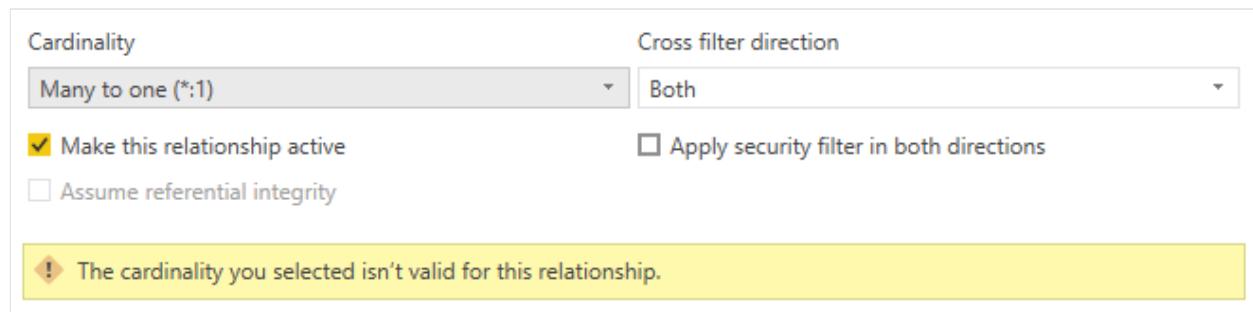
Name	Index
Dylan Williams	60
Dylan Williams	11
<b>Total</b>	<b>71</b>

In the preceding image, the first row has a total value of 60 for the **Index** field, so the first row in the visual represents the last two rows of the loaded data. The second row with total **Index** value of 11 represents the first two rows. The difference in the number of rows between the visual and the data table is caused by the engine automatically removing or trimming trailing spaces, but not leading spaces. So the engine evaluates the first and second rows, and the third and fourth rows, as identical, and the visual returns these results.

This behavior can also cause error messages related to relationships, because duplicate values are detected. For example, depending on the configuration of your relationships, you might see an error similar to the following image:



In other situations, you might be unable to create a many-to-one or one-to-one relationship because duplicate values are detected.



You can trace these errors back to leading or trailing spaces, and resolve them by using [Text.Trim](#), or **Trim** under **Transform**, to remove the spaces in Power Query Editor.

## True/false type

The **True/false** data type is a Boolean value of either *True* or *False*. For the best and most consistent results, when you load a column that contains Boolean true/false information into Power BI, set the column type to **True/False**.

Power BI converts and displays data differently in certain situations. This section describes common cases of converting Boolean values, and how to address conversions that create unexpected results in Power BI.

In this example, you load data about whether your customers have signed up for your newsletter. A value of *TRUE* indicates the customer has signed up for the newsletter, and a value of *FALSE* indicates the customer hasn't signed up.

However, when you publish the report to the Power BI service, the newsletter signup status column shows 0 and -1 instead of the expected values of *TRUE* or *FALSE*. The following steps describe how this conversion occurs, and how to prevent it.

The simplified query for this table appears in the following image:

	Customer Number	Subscribed To Newsletter
1	1001	FALSE
2	1002	TRUE
3	1003	TRUE
4	1004	FALSE
5	1005	TRUE

The data type of the **Subscribed To Newsletter** column is set to **Any**, and as a result, Power BI loads the data into the model as **Text**.

The screenshot shows the Power BI Data Editor interface. At the top, there are two tabs: "Structure" and "Formatting". The "Formatting" tab is active, showing the "Name" (Subscribed To New...) and "Data type" (Text) fields. The "Data type" field is highlighted with a red box. Below the tabs, there is a preview pane showing the data in a table format. The columns are "Customer Number" and "Subscribed To Newsletter". The data rows are: 1001 (FALSE), 1002 (TRUE), 1003 (TRUE), 1004 (FALSE), and 1005 (TRUE). The "Customer Number" column is also highlighted with a red box.

When you add a simple visualization that shows the detailed information per customer, the data appears in the visual as expected, both in Power BI Desktop and when

published to the Power BI service.

Customer Number	Subscribed To Newsletter
1001	FALSE
1002	TRUE
1003	TRUE
1004	FALSE
1005	TRUE

However, when you refresh the dataset in the Power BI service, the **Subscribed To Newsletter** column in the visuals displays values as *-1* and *0*, instead of displaying them as *TRUE* or *FALSE*:

Customer Number	Subscribed To Newsletter
1001	0
1002	-1
1003	-1
1004	0
1005	-1

If you republish the report from Power BI Desktop, the **Subscribed To Newsletter** column again shows *TRUE* or *FALSE* as you expect, but once a refresh occurs in the Power BI service, the values again change to show *-1* and *0*.

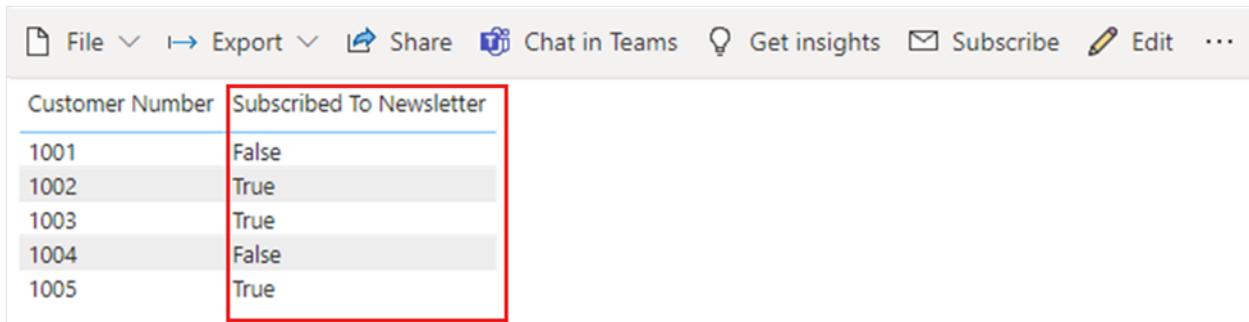
The solution to prevent this situation is to set any Boolean columns to type **True/False** in Power BI Desktop, and republish your report.

	Customer Number	Subscribed To Newsletter
1	1001	FALSE
2	1002	TRUE
3	1003	TRUE
4	1004	FALSE
5	1005	TRUE

When you make the change, the visualization shows the values in the **Subscribed To Newsletter** column slightly differently. Rather than the text being all capital letters as entered in the table, only the first letter is capitalized. This change is one result of changing the column's data type.

Customer Number	Subscribed To Newsletter
1001	False
1002	True
1003	True
1004	False
1005	True

Once you change the data type, republish to the Power BI service, and a refresh occurs, the report displays the values as *True* or *False*, as expected.



The screenshot shows a Power BI report interface. At the top, there's a navigation bar with File, Export, Share, Chat in Teams, Get insights, Subscribe, Edit, and more. Below the navigation bar is a table with two columns: 'Customer Number' and 'Subscribed To Newsletter'. The 'Subscribed To Newsletter' column contains the values False, True, True, False, and True. This column is highlighted with a red rectangular box. The table rows have alternating light gray and white backgrounds.

To summarize, when working with Boolean data in Power BI, make sure your columns are set to the **True/False** data type in Power BI Desktop.

## Blank type

**Blank** is a DAX data type that represents and replaces SQL nulls. You can create a blank by using the **BLANK** function, and test for blanks by using the **ISBLANK** logical function.

## Binary type

You can use the **Binary** data type to represent any data with a binary format. In the Power Query Editor, you can use this data type when loading binary files if you convert it to other data types before you load it into the Power BI model.

Binary columns aren't supported in the Power BI data model. The **Binary** selection exists in the Data View and Report View menus for legacy reasons, but if you try to load binary columns to the Power BI model, you might run into errors.

### Note

If a binary column is in the output of the steps of a query, attempting to refresh the data through a gateway can cause errors. It's recommended that you explicitly

remove any binary columns as the last step in your queries.

## Table type

DAX uses a table data type in many functions, such as aggregations and time intelligence calculations. Some functions require a reference to a table. Other functions return a table that you can then use as input to other functions.

In some functions that require a table as input, you can specify an expression that evaluates to a table. Some functions require a reference to a base table. For information about the requirements of specific functions, see the [DAX Function Reference](#).

## Implicit and explicit data type conversion

Each DAX function has specific requirements for the types of data to use as inputs and outputs. For example, some functions require integers for some arguments and dates for others. Other functions require text or tables.

If the data in the column you specify as an argument is incompatible with the data type the function requires, DAX may return an error. However, wherever possible DAX attempts to implicitly convert the data to the required data type.

For example:

- If you type a date as a string, DAX parses the string and tries to cast it as one of the Windows date and time formats.
- You can add *TRUE* + 1 and get the result 2, because DAX implicitly converts *TRUE* to the number 1, and does the operation 1+1.
- If you add values in two columns with one value represented as text ("12") and the other as a number (12), DAX implicitly converts the string to a number, and then does the addition for a numeric result. The expression = "22" + 22 returns 44.
- If you try to concatenate two numbers, DAX presents them as strings, and then concatenates. The expression = 12 & 34 returns "1234".

## Tables of implicit data conversions

The operator determines the type of conversion DAX performs by casting the values it requires before doing the requested operation. The following tables list the operators, and the conversion DAX does on each data type when it pairs with the data type in the intersecting cell.

### Note

These tables don't include **Text** data type. When a number is represented in a text format, in some cases Power BI tries to determine the number type and represent the data as a number.

## Addition (+)

	INTEGER	CURRENCY	REAL	Date/time
INTEGER	INTEGER	CURRENCY	REAL	Date/time
CURRENCY	CURRENCY	CURRENCY	REAL	Date/time
REAL	REAL	REAL	REAL	Date/time
Date/time	Date/time	Date/time	Date/time	Date/time

For example, if an addition operation uses a real number in combination with currency data, DAX converts both values to REAL and returns the result as REAL.

## Subtraction (-)

In the following table, the row header is the minuend (left side) and the column header is the subtrahend (right side).

	INTEGER	CURRENCY	REAL	Date/time
INTEGER	INTEGER	CURRENCY	REAL	REAL
CURRENCY	CURRENCY	CURRENCY	REAL	REAL
REAL	REAL	REAL	REAL	REAL
Date/time	Date/time	Date/time	Date/time	Date/time

For example, if a subtraction operation uses a date with any other data type, DAX converts both values to dates, and the return value is also a date.

### Note

Data models support the unary operator, - (negative), but this operator doesn't change the data type of the operand.

## Multiplication (\*)

	INTEGER	CURRENCY	REAL	Date/time
INTEGER	INTEGER	CURRENCY	REAL	INTEGER
CURRENCY	CURRENCY	REAL	CURRENCY	CURRENCY
REAL	REAL	CURRENCY	REAL	REAL

For example, if a multiplication operation combines an integer with a real number, DAX converts both numbers to real numbers, and the return value is also REAL.

## Division (/)

In the following table, the row header is the numerator and the column header is the denominator.

	INTEGER	CURRENCY	REAL	Date/time
INTEGER	REAL	CURRENCY	REAL	REAL
CURRENCY	CURRENCY	REAL	CURRENCY	REAL
REAL	REAL	REAL	REAL	REAL
Date/time	REAL	REAL	REAL	REAL

For example, if a division operation combines an integer with a currency value, DAX converts both values to real numbers, and the result is also a real number.

## Comparison operators

In comparison expressions, DAX considers Boolean values greater than string values, and string values greater than numeric or date/time values. Numbers and date/time values have the same rank.

DAX doesn't do any implicit conversions for Boolean or string values. BLANK or a blank value is converted to 0, "", or *False*, depending on the data type of the other compared value.

The following DAX expressions illustrate this behavior:

- `=IF(FALSE()>"true","Expression is true", "Expression is false")` returns "Expression is true".

- `=IF("12">12,"Expression is true", "Expression is false")` returns "Expression is true".
- `=IF("12"=12,"Expression is true", "Expression is false")` returns "Expression is false".

DAX does implicit conversions for numeric or date/time types as the following table describes:

<b>Comparison Operator</b>	<b>INTEGER</b>	<b>CURRENCY</b>	<b>REAL</b>	<b>Date/time</b>
INTEGER	INTEGER	CURRENCY	REAL	REAL
CURRENCY	CURRENCY	CURRENCY	REAL	REAL
REAL	REAL	REAL	REAL	REAL
Date/time	REAL	REAL	REAL	Date/Time

## Blanks, empty strings, and zero values

DAX represents a null, blank value, empty cell, or missing value by the same new value type, a BLANK. You can also generate blanks by using the BLANK function, or test for blanks by using the ISBLANK function.

How operations such as addition or concatenation handle blanks depends on the individual function. The following table summarizes the differences between how DAX and Microsoft Excel formulas handle blanks.

<b>Expression</b>	<b>DAX</b>	<b>Excel</b>
BLANK + BLANK	BLANK	0 (zero)
BLANK + 5	5	5
BLANK * 5	BLANK	0 (zero)
5/BLANK	Infinity	Error
0/BLANK	NaN	Error
BLANK/BLANK	BLANK	Error
FALSE OR BLANK	FALSE	FALSE
FALSE AND BLANK	FALSE	FALSE

Expression	DAX	Excel
TRUE OR BLANK	TRUE	TRUE
TRUE AND BLANK	FALSE	TRUE
BLANK OR BLANK	BLANK	Error
BLANK AND BLANK	BLANK	Error

## Next steps

You can do all sorts of things with Power BI Desktop and data. For more information on Power BI capabilities, see the following resources:

- [What is Power BI Desktop?](#)
- [Query overview with Power BI Desktop](#)
- [Data sources in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)

# Connector extensibility in Power BI

Article • 12/22/2022

Power BI can connect to data by using existing connectors and generic data sources, like ODBC, OData, OLE DB, Web, CSV, XML, and JSON. Developers can also enable new data sources with custom data extensions called *custom connectors*. Microsoft certifies and distributes some custom connectors as *certified connectors*.

To use non-certified custom connectors that you or another party develop, you must adjust your Power BI Desktop security settings to allow extensions to load without validation or warning. These extensions can ignore privacy levels and handle credentials, including sending them over HTTP, so you should use this setting only if you completely trust your custom connectors.

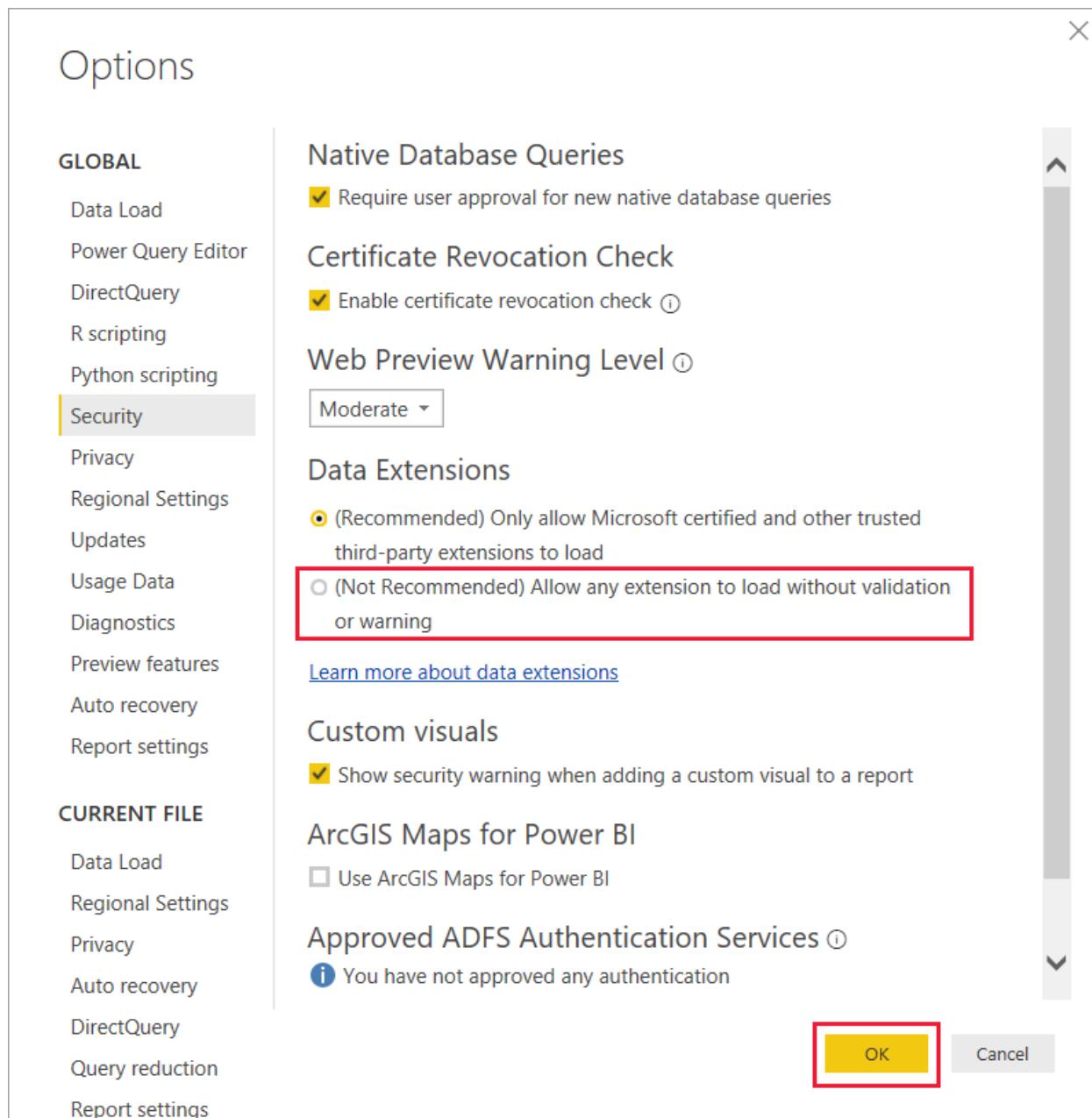
Another option is for the developer to sign the connector with a certificate, and provide the information you need to use the connector without changing your security settings. For more information, see [Trusted third-party connectors](#).

## Custom connectors

Non-certified custom connectors can range from small business-critical APIs to large industry-specific services that Microsoft hasn't released a connector for. Many connectors are distributed by vendors. If you need a connector for a specific industry or business, contact the vendor.

To use a non-certified custom connector:

1. Put the connector *.pq*, *.pqx*, *.m*, or *.mez* file in your local *[Documents]\Power BI Desktop\Custom Connectors* folder. If the folder doesn't exist, create it.
2. To adjust the data extension security settings, in Power BI Desktop, select **File > Options and settings > Options > Security**.
3. Under **Data Extensions**, select **(Not Recommended) Allow any extension to load without validation or warning**.
4. Select **OK**, and then restart Power BI Desktop.



The default Power BI Desktop data extension security setting is **(Recommended) Only allow Microsoft certified and other trusted third-party extensions to load**. With this setting, if there are non-certified custom connectors on your system, the **Uncertified Connectors** dialog box appears at Power BI Desktop startup, listing the connectors that can't securely load.

## Uncertified Connectors

The following connector has not been certified, and we are unable to verify that it is secure to use:

TripPin

If it was provided by a vendor, please request that they contact Microsoft about the Certified Connector program. If you would still like to use it, please enable a lower data extension setting under the Security tab in the Options dialog, and restart Power BI Desktop.

OK

To clear the error if you don't need to use the connectors in this session, select **OK**.

To prevent the error, either change your **Data Extensions** security setting, or remove the uncertified connectors from your *Custom Connectors* folder.

### ⓘ Important

You can use only one custom connector data source when you work in DirectQuery mode. Multiple custom connector data sources won't work with DirectQuery.

## Certified connectors

Microsoft certifies a limited subset of custom data extensions. While Microsoft distributes these connectors, Microsoft isn't responsible for their performance or continued function. The third-party developer who created the connector is responsible for its maintenance and support.

In Power BI Desktop, certified third-party connectors appear in the list in the **Get Data** dialog box, along with generic and common connectors. You don't need to adjust security settings to use the certified connectors.

## Next steps

To get a custom connector certified, see [Power Query Connector Certification](#).

# Trusted third-party connectors

Article • 01/23/2023

In Power BI Desktop, we generally recommend keeping your *Data extension security* level at the higher level, which prevents loading of code not certified by Microsoft. However, there might be many cases in which you want to load specific connectors. These connectors include ones you've written and ones provided to you by a consultant or vendor outside the Microsoft certification path.

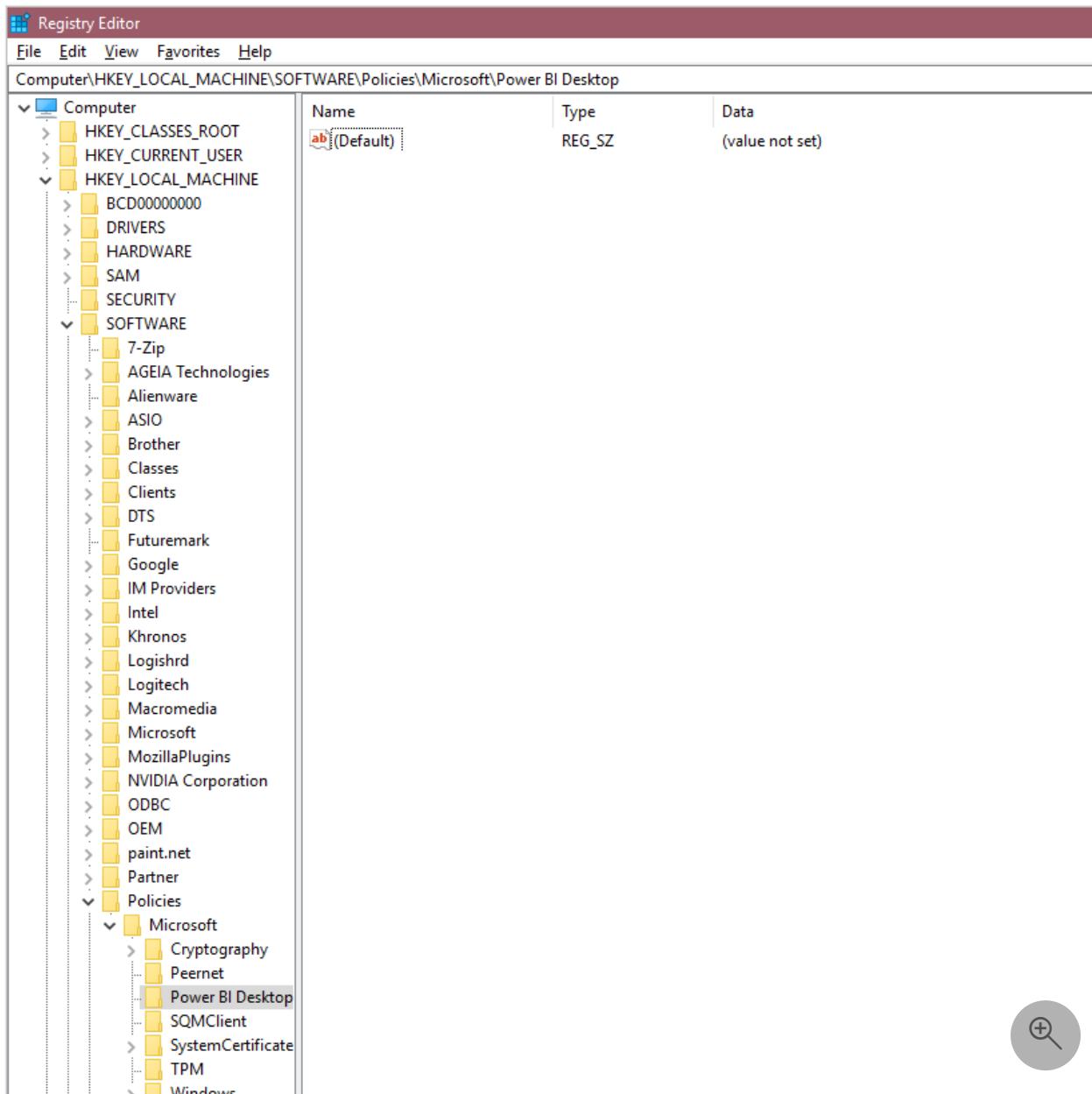
The developer of a given connector can sign it with a certificate and provide you with the information you need to securely load it without lowering your security settings.

For more information about the security settings, see [Connector extensibility in Power BI](#).

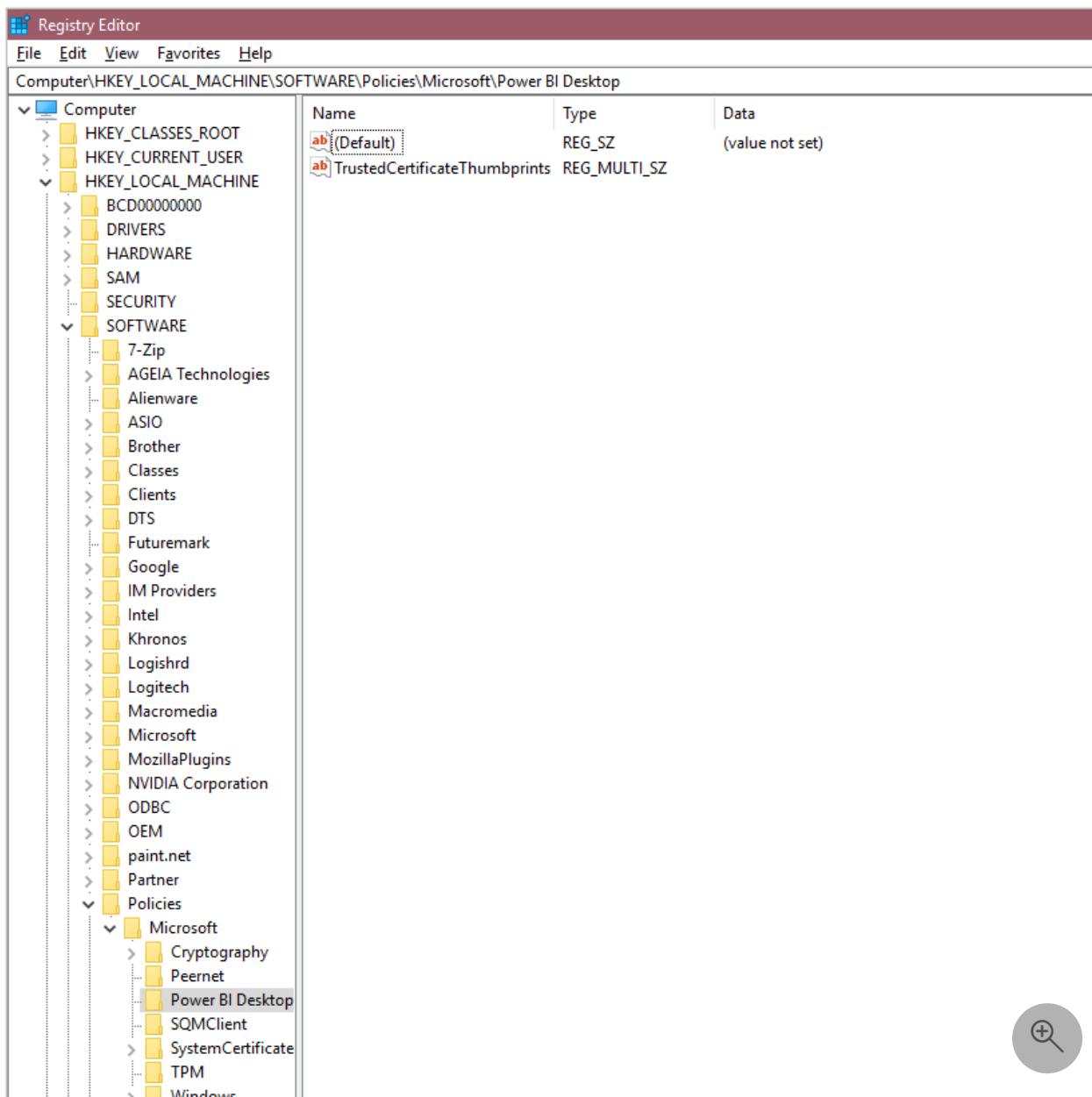
## Using the registry to trust third-party connectors

Trusting third-party connectors in Power BI Desktop is done by listing the thumbprint of the certificate you want to trust in a specified registry value. If this thumbprint matches the thumbprint of the certificate on the connector you want to load, you can load it in the *Recommended* security level of Power BI Desktop.

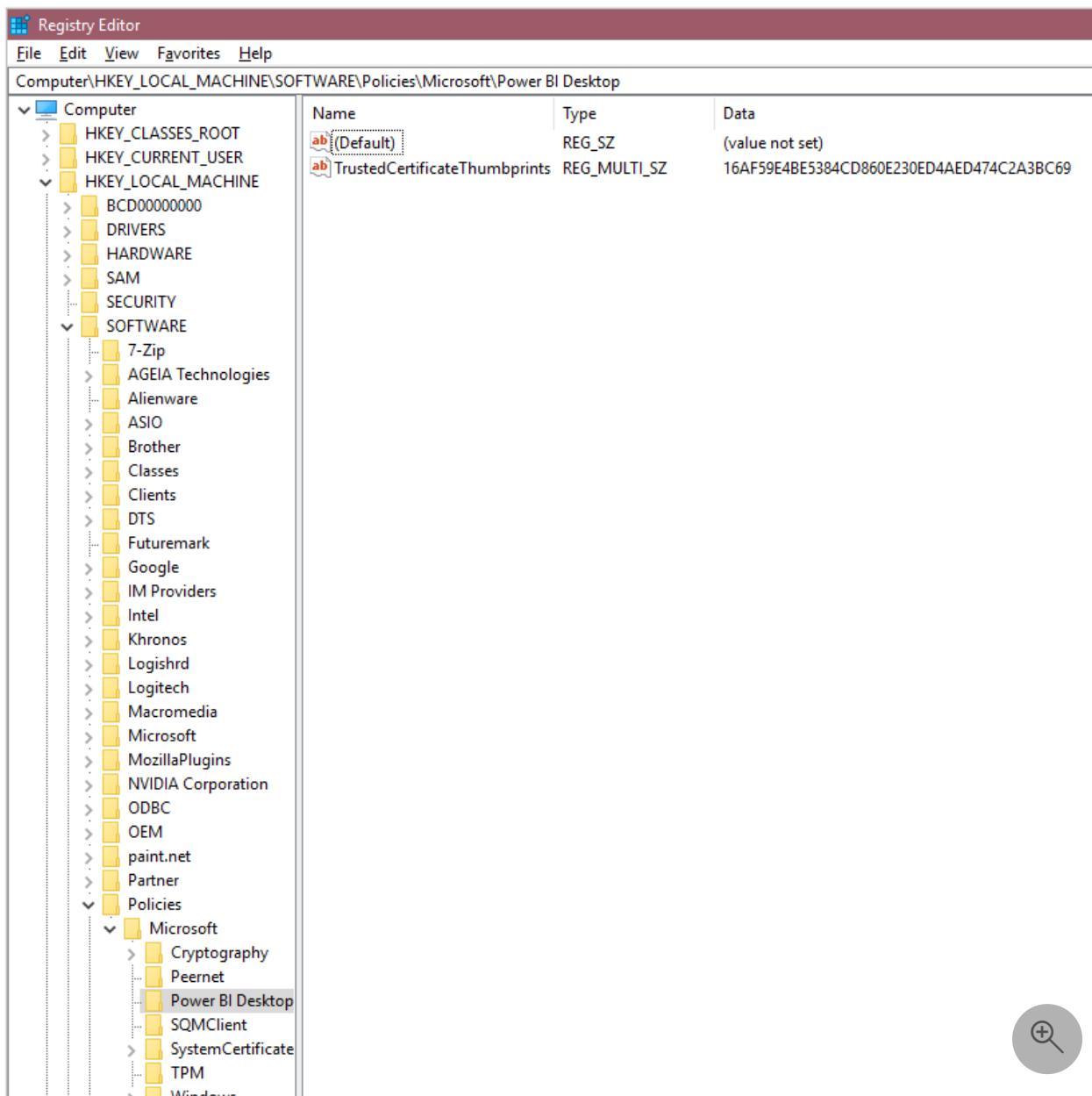
The registry path is `HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Power BI Desktop`. Make sure the path exists, or create it. This location is chosen due to it being primarily controlled by IT policy and requiring local machine administration access to edit.



Add a new value under the path specified in the previous image. The type should be Multi-String Value: `REG_MULTI_SZ`. It should be called `TrustedCertificateThumbprints`.



Add the thumbprints of the certificates you want to trust. You can add multiple certificates by using `\0` as a delimiter, or in the Registry Editor, right-click key, then select **Modify** to put each thumbprint on a new line. This example thumbprint is taken from a self-signed certificate.



If you have the proper thumbprint from your developer, you should now be able to securely trust connectors signed with the associated certificate.

## How to sign connectors

If you have a connector you or a developer need to sign, see [Handling Power Query Connector Signing](#).

# Manage DirectQuery connections to a published dataset

Article • 04/24/2023

By default, when you publish a dataset to the Power BI service, you can make a DirectQuery connection to it, assuming you have proper permissions. You can use this connection to create new composite models on top of the dataset.

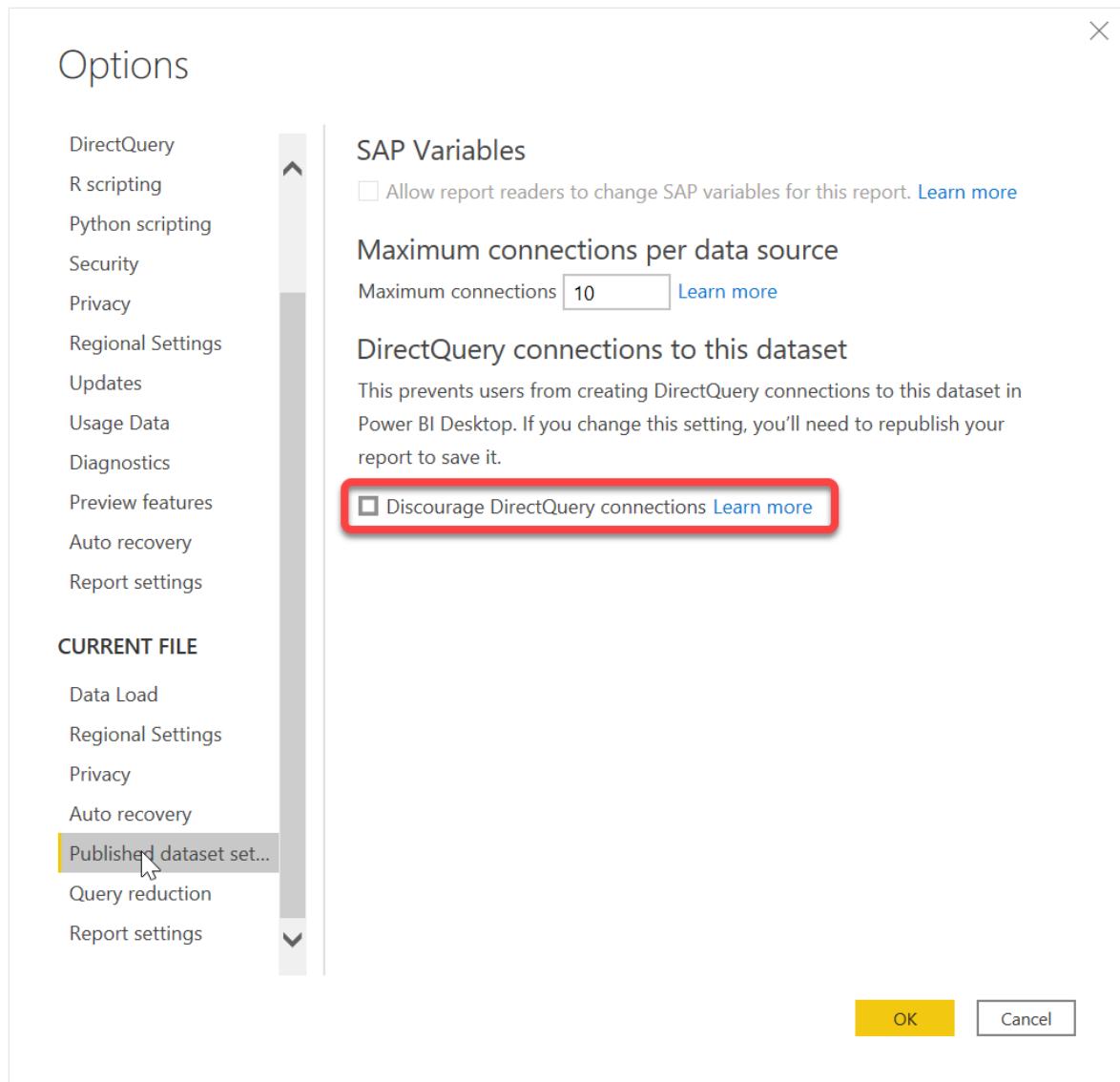
In some situations, however, you need to discourage these connections from happening. Discouraging these connections is especially important in the composite models scenario, where you might want to prohibit creation of new composite models on top of the dataset (so-called *chaining*). By discouraging DirectQuery connections to a dataset, you're effectively ending the chain or stopping it from forming in the first place.

## Note

Power BI honors this setting and disables making DirectQuery connections to a dataset, but third-party tools might not. Third-party tools might allow users to make DirectQuery connections to a dataset even if you disabled it.

## Use Power BI Desktop to discourage DirectQuery connections to a dataset

1. To discourage DirectQuery connections to a dataset, go to **File > Options and settings > Options > Current File > Published dataset settings**.
2. On this page, choose the **Discourage DirectQuery connections** option, and select **OK**.



## Use third-party tools to discourage DirectQuery connections to a dataset

By using third-party tools, you can discourage DirectQuery connections to a dataset by setting the `DiscourageCompositeModels` property on a model to `True`.

## Next steps

- [Using DirectQuery in Power BI](#)
- [Datasets in the Power BI service](#)
- [Use composite models in Power BI Desktop](#)
- More questions? [Ask the Power BI Community](#)

# Troubleshooting sign-in with OData feed

Article • 01/31/2023

This article contains troubleshooting options when signing in using an organization account for an OData feed:

- Credential type not supported error
- Access denied errors

The following sections describe each error, and the steps to remedy them, in turn.

## Credential type not supported

You might see the following error, indicating the credential type isn't supported:

Output

```
We are unable to connect because this credential type is not supported
by this resource. Please choose another credential type.
```

You need to ensure your service is sending auth headers as follows:

- First Oauth request without any *authorization* header should send the following header in response:

Output

```
www-authenticate: Bearer realm=https://login.microsoftonline.com/<Your
Active Directory Tenant Id>
```

- Redirect request to the service with *authorization* header set to *Bearer* should send the following header in response:

Output

```
www-authenticate: Bearer
authorization_uri=https://login.microsoftonline.com/<Your Active
Directory Tenant Id>/oauth2/authorize
```

After a successful redirect call, calls to your service have the right access token in the authorization header. If you still see an error, clear the **Global Permissions** for the OData

service URI and try again. To clear **Global Permissions**, go to **File > Options and Settings > Data Source Settings > Global Permissions**.

## Access denied

You might see one of the following errors, indicating access is denied:

### Output

```
access_denied: AADSTS650053: The application 'Microsoft Power Query for Excel'
asked for scope 'user_impersonation' that doesn't exist
on the resource \<resourceId\>.
```

### Output

Microsoft Power Query for Excel needs permission to access resources in your organization that only an admin can grant.  
Ask an admin to grant permission to this app before you can use it.

If you encounter such an error, ensure the application registration for your OData service has following settings:

- *Application ID* is set to the OData service base URI.
- Scope *user\_impersonation* is defined.
- The application's permissions are appropriately set by the administrator.

## Next steps

You can do all sorts of things with Power BI Desktop. For more information on its capabilities, check out the following resources:

- [What is Power BI Desktop?](#)
- [Query overview with Power BI Desktop](#)
- [Data types in Power BI Desktop](#)
- [Shape and combine data with Power BI Desktop](#)
- [Common query tasks in Power BI Desktop](#)