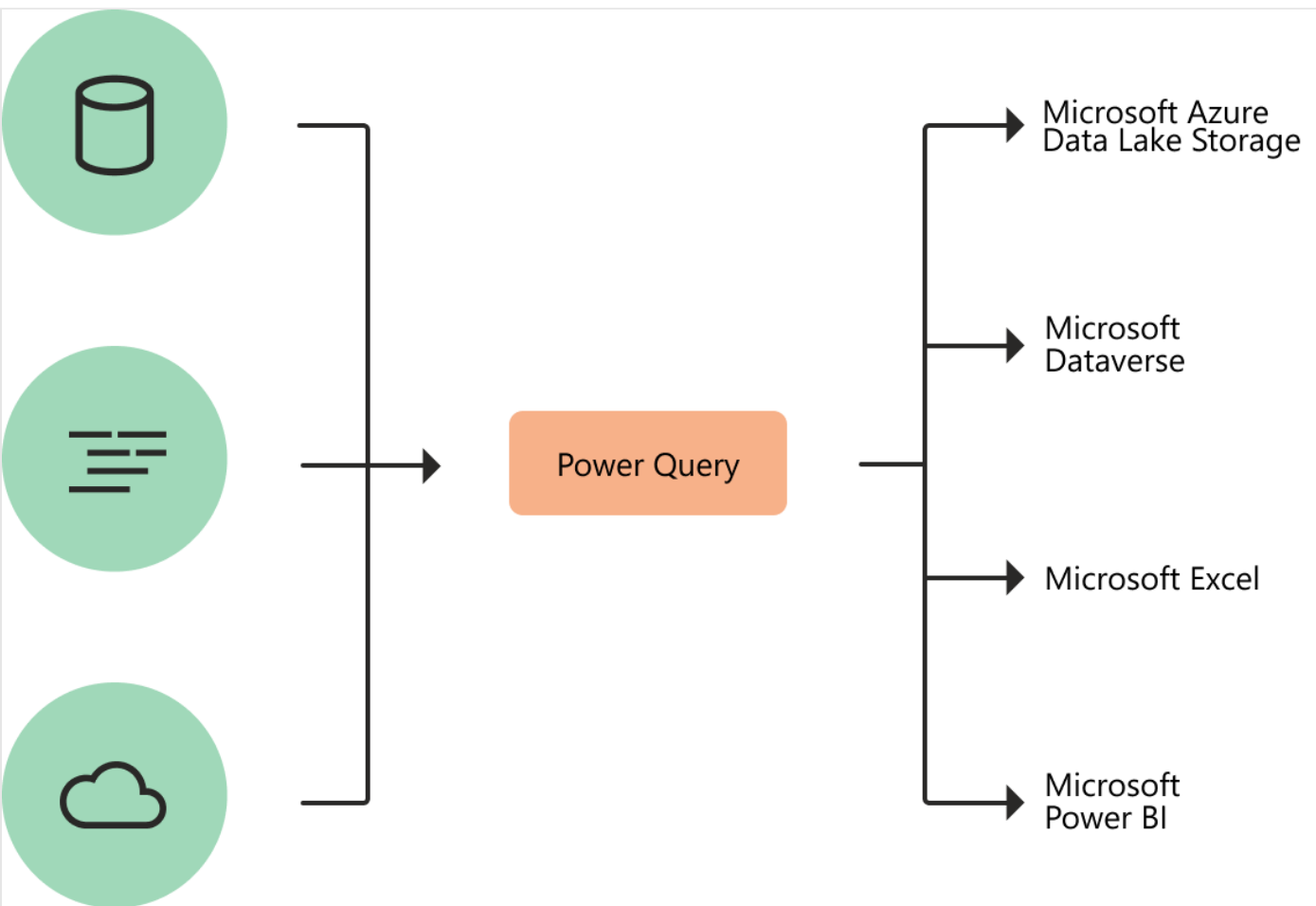


Power Query is a data transformation and data preparation engine. Power Query comes with a graphical interface for getting data from sources and a Power Query Editor for applying transformations. Because the engine is available in many products and services, the destination where the data will be stored depends on where Power Query was used. Using Power Query, you can perform the extract, transform, and load (ETL) processing of data.



## How Power Query helps with data acquisition

Business users spend up to 80 percent of their time on data preparation, which delays the work of analysis and decision-making. Several challenges contribute to this situation, and Power Query helps address many of them.

Existing challenge	How does Power Query help?
Finding and connecting to data is too difficult	Power Query enables connectivity to a wide range of data sources, including data of all sizes and shapes.
Experiences for data connectivity are too fragmented	Consistency of experience, and parity of query capabilities over all data sources.

Existing challenge	How does Power Query help?
Data often needs to be reshaped	Highly interactive and intuitive experience for rapidly and iteratively

before consumption	building queries over any data source, of any size.
Any shaping is one-off and not repeatable	<p>When using Power Query to access and transform data, you define a repeatable process (query) that can be easily refreshed in the future to get up-to-date data.</p> <p>In the event that you need to modify the process or query to account for underlying data or schema changes, you can use the same interactive and intuitive experience you used when you initially defined the query.</p>
Volume (data sizes), velocity (rate of change), and variety (breadth of data sources and data shapes)	<p>Power Query offers the ability to work against a subset of the entire dataset to define the required data transformations, allowing you to easily filter down and transform your data to a manageable size.</p> <p>Power Query queries can be refreshed manually or by taking advantage of scheduled refresh capabilities in specific products (such as Power BI) or even programmatically (by using the Excel object model).</p> <p>Because Power Query provides connectivity to hundreds of data sources and over 350 different types of data transformations for each of these sources, you can work with data from any source and in any shape.</p>

## Power Query experiences

The Power Query user experience is provided through the Power Query Editor user interface. The goal of this interface is to help you apply the transformations you need simply by interacting with a user-friendly set of ribbons, menus, buttons, and other interactive components.

The Power Query Editor is the primary data preparation experience, where you can connect to a wide range of data sources and apply hundreds of different data transformations by previewing data and selecting transformations from the UI. These data transformation capabilities are common across all data sources, whatever the underlying data source limitations.

When you create a new transformation step by interacting with the components of the Power Query interface, Power Query automatically creates the M code required to do the transformation so you don't need to write any code.

Currently, two Power Query experiences are available:

- **Power Query Online**—Found in integrations such as Power BI dataflows, Microsoft Power Platform dataflows, Azure Data Factory wrangling dataflows, and many more that provide the experience through an online webpage.
- **Power Query for Desktop**—Found in integrations such as Power Query for Excel and Power BI Desktop.

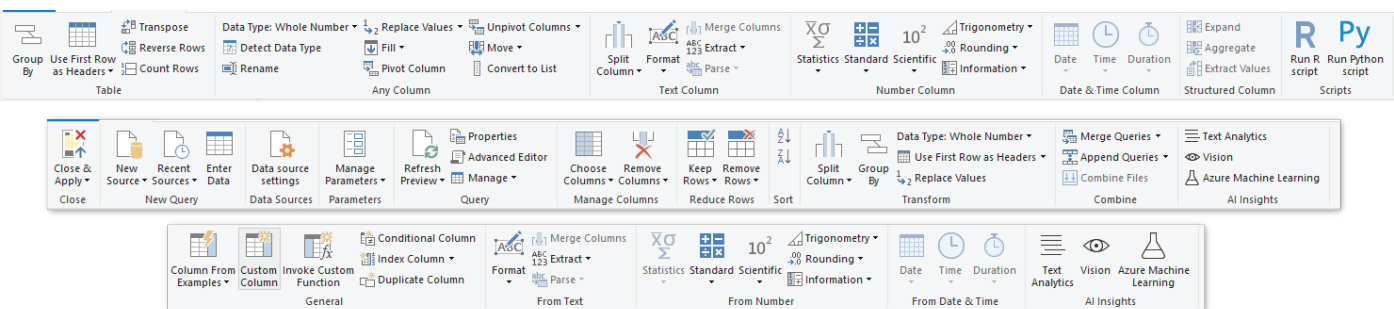
### Note

Although two Power Query experiences exist, they both provide almost the same user experience in every scenario.

# Transformations

The transformation engine in Power Query includes many prebuilt transformation functions that can be used through the graphical interface of the Power Query Editor. These transformations can be as simple as removing a column or filtering rows, or as common as using the first row as a table header. There are also advanced transformation options such as merge, append, group by, pivot, and unpivot.

All these transformations are made possible by choosing the transformation option in the menu, and then applying the options required for that transformation. The following illustration shows a few of the transformations available in Power Query Editor.



More information: [Quickstart: Using Power Query in Power BI](#)

# Dataflows

Power Query can be used in many products, such as Power BI and Excel. However, using Power Query within a product limits its usage to only that specific product. *Dataflows* are a product-agnostic service version of the Power Query experience that runs in the cloud. Using dataflows, you can get data and transform data in the same way, but instead of sending the output to Power BI or Excel, you can store the output in other storage options such as Dataverse or Azure Data Lake Storage. This way, you can use the output of dataflows in other products and services.

More information: [What are dataflows?](#)

# Power Query M formula language

In any data transformation scenario, there are some transformations that can't be done in the best way by using the graphical editor. Some of these transformations might require special configurations and settings that the graphical interface doesn't currently support. The Power Query engine uses a scripting language behind the scenes for all Power Query transformations: the Power Query M formula language, also known as M.

The M language is the data transformation language of Power Query. Anything that happens in the query is ultimately written in M. If you want to do advanced transformations using the Power Query engine, you can use the Advanced Editor to access the script of the query and modify it as

you want. If you find that the user interface functions and transformations won't perform the exact changes you need, use the Advanced Editor and the M language to fine-tune your functions and transformations.

Power Query M
<pre>let     Source = Exchange.Contents("xyz@contoso.com"),     Mail1 = Source{[Name="Mail"]}[Data],     #"Expanded Sender" = Table.ExpandRecordColumn(Mail1, "Sender", {"Name"}, {"Name"}),     #"Filtered Rows" = Table.SelectRows(#"Expanded Sender", each ([HasAttachments] = true)),     #"Filtered Rows1" = Table.SelectRows(#"Filtered Rows", each ([Subject] = "sample files for email PQ test") and ([Folder Path] = "\Inbox\")),     #"Removed Other Columns" = Table.SelectColumns(#"Filtered Rows1",{"Attachments"}),     #"Expanded Attachments" = Table.ExpandTableColumn(#"Removed Other Columns", "Attachments", {"Name", "AttachmentContent"}, {"Name", "AttachmentContent"}),     #"Filtered Hidden Files1" = Table.SelectRows(#"Expanded Attachments", each [Attributes]?[Hidden]? &lt;&gt; true),     #"Invoke Custom Function1" = Table.AddColumn(#"Filtered Hidden Files1", "Transform File from Mail", each #"Transform File from Mail"([AttachmentContent])),     #"Removed Other Columns1" = Table.SelectColumns(#"Invoke Custom Function1", {"Transform File from Mail"}),     #"Expanded Table Column1" = Table.ExpandTableColumn(#"Removed Other Columns1", "Transform File from Mail", Table.ColumnNames(#"Transform File from Mail"("#Sample File"))),     #"Changed Type" = Table.TransformColumnTypes(#"Expanded Table Column1", {"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}, {"Column5", type text}, {"Column6", type text}, {"Column7", type text}, {"Column8", type text}, {"Column9", type text}, {"Column10", type text}) in     #"Changed Type"</pre>

More information: [Power Query M formula language](#)

## Where can you use Power Query?

The following table lists Microsoft products and services where Power Query can be found.

Product	M engine <sup>1</sup>	Power Query Desktop <sup>2</sup>	Power Query Online <sup>3</sup>	Dataflows <sup>4</sup>
Excel for Windows	Yes	Yes	No	No
Excel for Mac	Yes	No	No	No
Power BI	Yes	Yes	Yes	Yes

Product	M engine <sup>1</sup>	Power Query Desktop <sup>2</sup>	Power Query Online <sup>3</sup>	Dataflows <sup>4</sup>

Power Apps	Yes	No	Yes	Yes
Power Automate	Yes	No	Yes	No
Power BI Report Server	Yes	Yes	No	No
Azure Data Factory	Yes	No	Yes	Yes
SQL Server Integration Services	Yes	No	No	No
SQL Server Analysis Services	Yes	Yes	No	No
Dynamics 365 Customer Insights	Yes	No	Yes	Yes

<sup>1</sup> <b>M engine</b>	The underlying query execution engine that runs queries expressed in the Power Query formula language ("M").
<sup>2</sup> <b>Power Query Desktop</b>	The Power Query experience found in desktop applications.
<sup>3</sup> <b>Power Query Online</b>	The Power Query experience found in web browser applications.
<sup>4</sup> <b>Dataflows</b>	Power Query as a service that runs in the cloud and is product-agnostic. The stored result can be used in other applications as services.

## See also

[Data sources in Power Query](#)

[Getting data](#)

[Power Query quickstart](#)

[Shape and combine data using Power Query](#)

[What are dataflows](#)