Microsoft

Power BI and SAP BW

Abstract: This document describes how SAP customers can benefit from connecting Power BI to their existing SAP Business Warehouse (BW) systems.

Target audience: BI analysts, BI-focused IT and SAP professionals

Published: November 2019

Contents

Power BI Overview 4
Power BI Differentiators 4
Power BI Service 5
Power BI Desktop 5
Power BI Platform 6
On-Premises Data Gateway 7
SAP Business Warehouse 8
Introduction to SAP BW 8
Data Objects in SAP BW 8
Reporting in SAP BW 9
Power BI and SAP BW 11
Integration Architecture 11
SAP BW Data Sources 12
SAP BW Connector versions: 13
Installation of SAP BW Connector – Implementation 1.0 14
Prerequisites 14
Installation of SAP BW Connector – Implementation 2.0 14
Prerequisites 14
Power BI Desktop Installation Steps 15
Getting started with the SAP BW Connector 16
Connection Types for SAP Business Warehouse 16
SAP BW Application Server Connection 16
SAP BW Message Server Connection 18
Authentication Methods for SAP BW Connections 20
User Name / Password Authentication 20
Windows Authentication and Single Sign-On 21
Navigating the Query Objects in SAP BW 23
Characteristic Hierarchies 25
Measure Properties 25
Flattening of Multi-Dimensional Data 26
Query Parameters 26
Loading SAP Data into Power BI Desktop 29

Advanced Topics ... 30
Logon Language Code ... 30
MDX Statement ... 31
Query Editor in Power BI Desktop ... 35
Decimal Notation ... 39
Import vs. Direct Query ... 40
Import Connections ... 40
DirectQuery Connections ... 41
General Recommendations ... 43
Data Refresh with the On-Premises Gateway ... 45
Installation of the Gateway ... 45
On-premises Data Gateway App ... 45
Managing your SAP BW Data Source in Power BI ... 46
Using your SAP BW Data Source in Power BI ... 49
Import Connections: Scheduled Refresh ... 49
Import Connections: Manual Refresh ... 51
Direct Query Connections ... 51
Support for SAP BW Features ... 52
Performance Considerations ... 56
Enabling Tracing to debug ... 58
Single Sign-On Configuration ... 58
Requirements ... 58
SAP BW Configuration ... 58
User Mapping ... 61
Secure Login Client ... 61
Configuring SAP GUI for SNC Communications ... 61

Power BI Overview

Power BI is a cloud-based analytics service that provides faster time to insight. It is used for visualizing, exploring and extracting insights from data. Power BI brings together data from diverse sources to deliver rich, comprehensive views of business operations.

What that means is, with Power BI, you can see all of your data through a single pane of glass, and create an analytics environment in minutes to monitor data and share reports. Live dashboards and reports show visualizations and KPIs from data that can reside both on-premises and in the cloud, providing a consolidated view across your business, regardless of where your data lives.

Power BI comes with two companion applications. The first one is Power BI Desktop, a visual data exploration and reporting tool for business analysts. The second one is a native, interactive mobile app for business users that is available for iOS, Android and Windows mobile devices.

In addition, Power BI offers a platform for developers to extend, embed and integrate Power BI with custom solutions. Interactive reports and visualizations can be embedded into applications and web sites and new visualizations can be created for specific industries and use cases. A set of powerful REST APIs is available to connect applications with visualizations and data in Power BI.

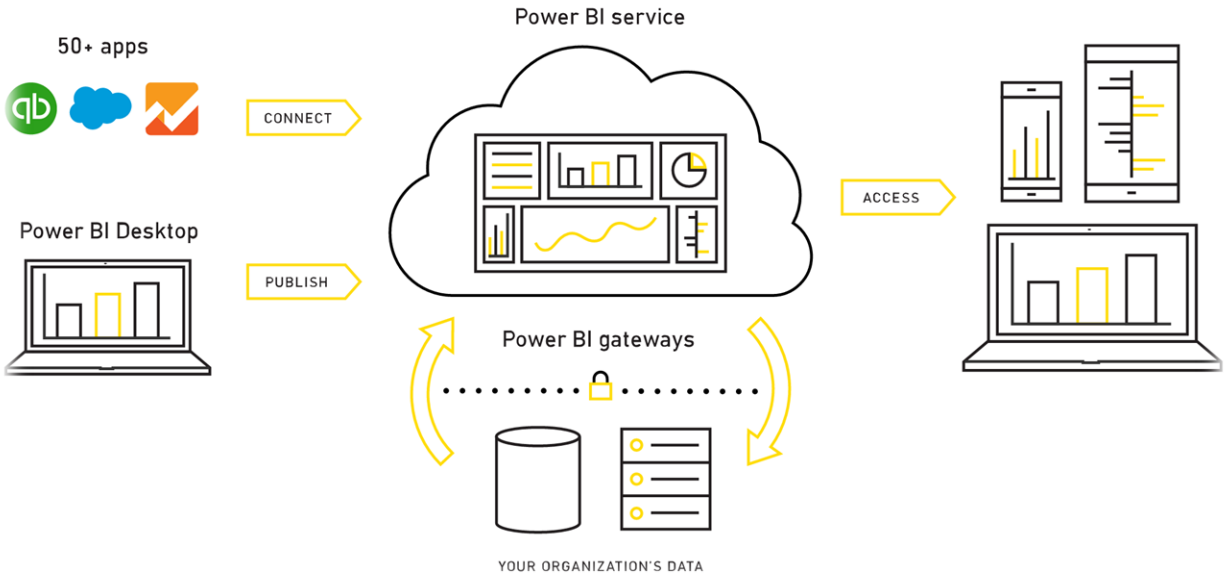
Power BI Differentiators

Power BI provides a unique set of features that empower everyone to benefit from direct access to data and insights. Power BI's distinctive set of features include:

- Pre-built dashboards and reports for popular SaaS solutions
- Live, real-time dashboards
- Secure, live connectivity to your data sources, on-premises and in the cloud
- Intuitive data exploration using natural language query functionality
- Auto insights into your datasets and reports
- Integrated with other Microsoft products and cloud services, including Azure Data Warehouse, Azure Stream Analytics, Azure Machine Learning, Office 365, Cortana

This set of capabilities is designed to help organizations get the most out of their data, no matter where that data lives – on-premise or in the cloud.

The main components of Power BI and their capabilities are briefly introduced below.



Power BI Service

The Power BI service offers a simple, intuitive experience for business users to interact with their data. You can build rich, live dashboards that provide a single consolidated view of your business. A dashboard is a set of visualizations, or charts, from one or more underlying reports, presented in an engaging way that makes it easy to glean insights - no analytics expertise needed. Users can enjoy the benefits of up-to-date data with real-time, automatic and scheduled refreshes. Dashboards can be shared with other people that you work with, inside and outside of your organization.

Users can ask questions of their data in natural language, or use auto-generated insights to quickly analyze a data set or report.

If you are not at the office, take your dashboards and reports with you on the Power BI mobile app for your device.

Power BI connects to a variety of data sources, both in the cloud and on-premises. You can connect to currently more than 60 popular SaaS solutions, as well as your databases, regardless of where they live.

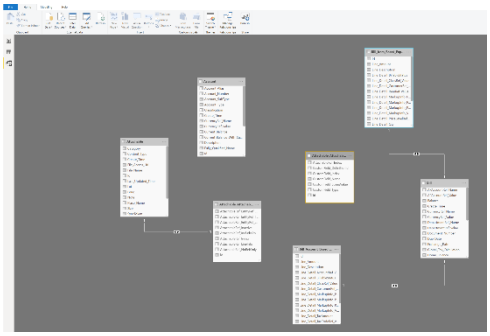
Power BI also works well with files. Excel workbooks can be directly connected to, or may be used with Power BI Desktop (more on Power BI Desktop below). Like Excel files, or CSV files, Power BI Desktop files can be uploaded to the Power BI service.

Power BI Desktop

Power BI Desktop - a desktop companion application to the Power BI service - is a visual data exploration and reporting tool geared toward BI analysts. Power BI Desktop centralizes, simplifies, and streamlines what

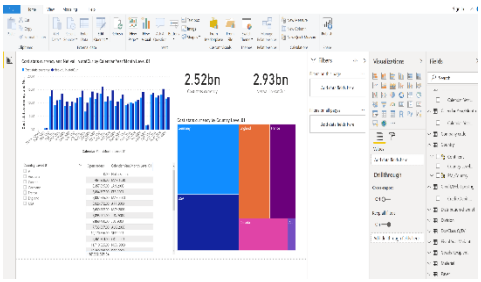
can otherwise be a scattered, disconnected, and arduous process of designing and creating business intelligence repositories and reports.

Among other things, Power BI Desktop enables you to acquire and prepare data using its built-in query capability. Connect to the ever-expanding world of data, including your SAP Business Warehouse or SAP HANA environments. Moreover, you can connect to your data sources in two different ways: you can import a copy of the data into Power BI Desktop, or you can connect directly to the data in its original source repository, which is known as DirectQuery. Both connection modes are available for SAP Business Warehouse and SAP HANA.



Whether on-premise, in the cloud, big data, or file data, once you connect to your data, Power BI Desktop enables you to transform, or shape, the data to meet your needs. With Power BI Desktop, shaping data involves defining and refining queries. Every dataset imported into Power BI Desktop is a query that can be modified.

Power BI Desktop also provides advanced analytics capabilities that help you structure complex data, create relationships, define a variety of calculations, find correlations, highlight exceptions, forecast business outcomes and more.



Explore data using a freeform, drag-and-drop canvas, and author reports with a broad range of data visualizations. When you are ready for others to use your data models and reports, publish directly to Power BI, and share through a unique, curated content approach

Power BI Platform

Power BI is also a platform that allows developers to embed, extend, and integrate Power BI with other applications and environments.

Power BI allows you to embed stunning, fully interactive reports into your customer-facing apps without the time and expense of building your own controls from the ground up. As a developer, you've built applications, and those applications have their own users and distinct set of features. Those apps may also have some built-in data elements like charts and reports that can now be powered by Microsoft Power BI Embedded. Users don't need a Power BI account to use your app. They can continue to sign in to your application just like before, and view and interact with the Power BI reporting experience without requiring any additional licensing.

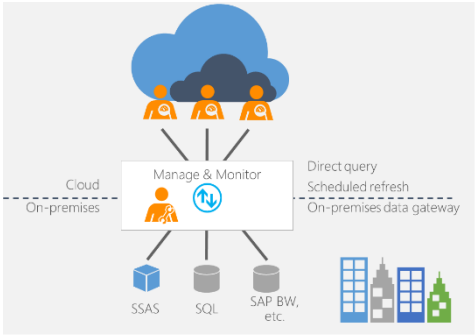
You can create customized visual experiences that meet your users' needs and match your application's design. Microsoft's open-sourced, production-quality visualization code is available on GitHub and

development and test tools are available to help the developer community build high quality custom visuals for Power BI.

The REST APIs make it possible to integrate Power BI with existing application workflows to deliver immediate insights for professional and non-technical users. Applications can send data directly into Power BI, adding data exploration capabilities to an existing system. This allows for dynamic updates to dashboards in real-time when the data changes, ensuring that users have the latest information.

On-Premises Data Gateway

With Power BI, data that needs to reside on-premises can continue to reside on-premises. The on-premises data gateway acts as a bridge, providing quick and secure data transfer between on-premises data, including your SAP Business Warehouse, and the Power BI service. The data gateway also works across several Microsoft services, including Power BI, Microsoft Flow, and PowerApps.



Installing and configuring the data gateway can be done centrally, usually by an IT administrator for a department or organization. IT can monitor and audit the usage of data which is important for BI governance.

With the data gateway, you can keep your dashboards, reports and datasets fresh, on an ad-hoc basis or a scheduled interval. The gateway works with data that has been imported into Power BI, or directly connected to in the source system, which is known as DirectQuery and is further outlined below.

SAP Business Warehouse

Introduction to SAP BW

SAP Business Warehouse (also known as SAP NetWeaver Business Warehouse, or simply SAP BW) is an Enterprise Data Warehouse solution of SAP that provides key BI capabilities for enterprise reporting, query, and analysis, as well as business planning. SAP BW runs on standard relational databases (RDBMS), including Microsoft SQL Server. As of release 7.4, SAP BW is optimized to run on SAP's HANA in-memory database.

SAP BW offers flexible ways of integrating data from various sources, both SAP and non-SAP systems, but in particular from SAP's transactional business systems: SAP ECC (ERP), CRM, SRM etc.

SAP BW is both rich and complex in functionality, this paper we will focus on the capabilities and key concepts that are relevant for the use of Power BI with SAP BW.

Data Objects in SAP BW

SAP BW comes with built-in tools for creating data models based on different data objects. It is helpful to have a rudimentary understanding of how data is represented in SAP BW and the terminology. The main data objects in SAP BW are briefly introduced here:

InfoProvider is the generic term for a BI object into which data is loaded or which provides views of data. InfoProviders can be queried with client tools, such as Business Explorer (or BEx, described below) and also with Power BI Desktop.

InfoProviders can be seen as uniform data providers from the viewpoint of a query definition. Their data can therefore be analyzed in a uniform way.

InfoCube is a type of InfoProvider. An InfoCube describes, from an analysis point of view, a self-contained dataset, for a business-orientated area, for example Purchasing. You can analyze an InfoCube directly as an InfoProvider with analysis and reporting tools, including Power BI.

An InfoCube consists of a set of relational tables that are arranged according to an enhanced star schema. This means there is a (large) fact table that contains the key figures for the InfoCube, as well as several (smaller) dimension tables which surround it.

Key figure is an operational attribute, which indicates a numerical measure such as amount, weight, quantity, etc.

Dimension is a grouping of related **characteristics** under a single generic term. For example, the *Customer* dimension could be made up of the *Customer Number*, the *Customer Group* and the levels of the customer hierarchy.

A *Sales* dimension could contain the characteristics *Sales Person*, *Sales Group* and *Sales Office*.

A *Time* dimension could have the characteristics *Day* (in the form YYYYMMDD), *Week* (in the form YYYY.WW), *Month* (in the form YYYY.MM), *Year* (in the form YYYY) and *Fiscal Period* (in the form YYYY.PPP).

Characteristics refer to master data with their **attributes** and **text descriptions,** and in some cases **hierarchies**. The characteristics of an InfoCube are stored in dimensions.

For example, the *Customer* dimension could have the characteristics *Sold-to-party, Ship-to-party, and Payer*.

The characteristic *Sold-to-party* could have the attributes *Country*, *Region*, *City*, *Street* and *Industry*. The text description of the characteristic would be the *Name* of the *Sold-to-party*.

In MDX query terms, the attributes of characteristics are also referred to as **properties**.

InfoObjects is the generic term for all characteristics and key figures. All InfoObjects are maintained independently of the InfoCube in SAP BW. InfoObjects are the smallest units of BI. Using InfoObjects, information can be stored and mapped in a structured form. This is required for constructing InfoProviders. InfoObjects with attributes or texts can themselves be InfoProviders.

DataStore Object (DSO) serves as a storage location for consolidated and cleansed transaction data or master data on a document (atomic) level. Unlike the multidimensional data in InfoCubes, the data in DataStore objects is stored in transparent, flat database tables. The system does not create separate fact tables or dimension tables for DSOs. Data in DSOs can be evaluated using a BEx query.

MultiProviders are a special type of InfoProvider that combine data from several InfoProviders. They are then available for reporting. MultiProviders themselves do not contain any data, their data comes exclusively from the InfoProviders upon which they are based on. MultiProviders can be based upon any combination of InfoProviders, including InfoCubes, DataStore Objects, InfoObjects, or InfoSets.

InfoSets are a special type of InfoProvider which does not store data physically. InfoSets describe data which is based on joining the tables of other InfoProviders like DataStore Objects, standard InfoCubes, or InfoObjects with master data characteristics. InfoSets can be useful when you have to build a report spanning two or more different data targets in SAP BW.

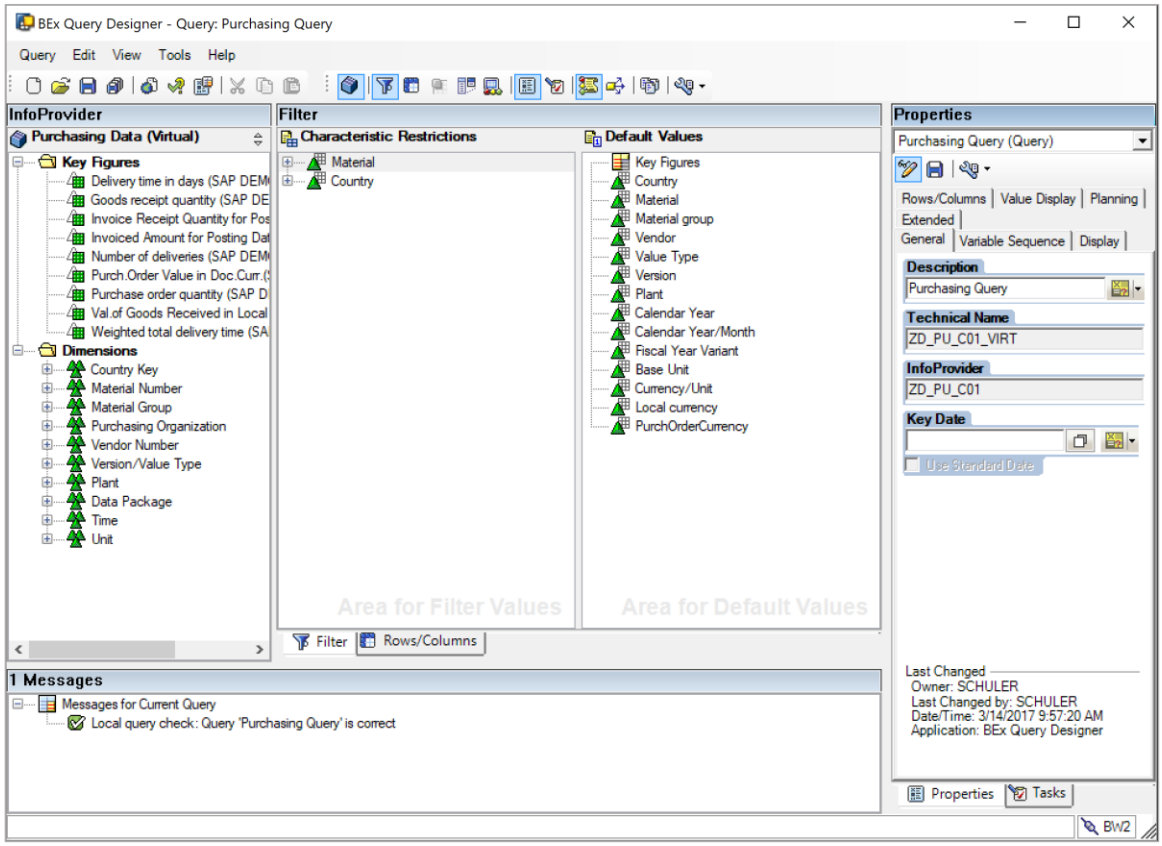
Composite Providers are a new data object in BW systems that run on HANA, e.g. SAP BW 7.5 or BW4/HANA. A composite provider is based on a JOIN or UNION of other InfoProviders or Analytic Indexes. Data in Composite Providers can be evaluated using a BEx query.

Reporting in SAP BW

SAP BW comes with a set of tools – the **Business Explorer**, or simply **BEx** tools - for enterprise reporting, ad- hoc analysis, and dashboarding. With the BEx tools, business users have the ability to design, create, analyze, and distribute data for analyses and decision-making support within their organization. In addition to BEx, SAP offers several BI reporting and analysis tools under the BusinessObjects (BO) brand that offer pixel-perfect reporting, semantic models, browser-based user experiences, and dashboarding. The BusinessObjects tools can work on top of BEx and can also be used with non-SAP data sources.

At the core of the BEx reporting and analysis functionality in SAP BW are **Queries**. Queries provide a flexible platform for data analysis in SAP BW are important to understand for the Power BI user. Queries act as the bridge between SAP BW InfoProviders and the reporting / analysis client tools.

The **BEx Query Designer** is the main tool for designing queries and generating datasets from an InfoProvider for analysis. With the Query Designer, the business user can limit the number of fields that are displayed on a report. It also allows the user to define **filters**, **variables**, **calculations**, and more, that are otherwise not available in the InfoProvider.

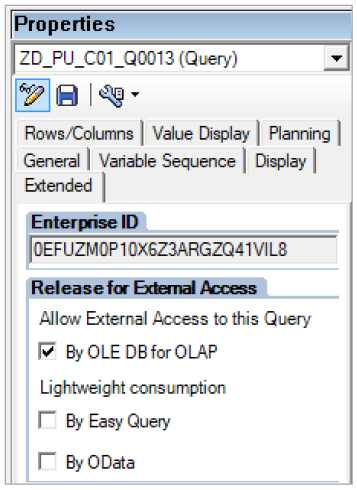


Filters can be used to restrict the data retrieved by a query. Filters can be dynamic or static in nature. Values entered as static filters cannot be overwritten by users at runtime while dynamic filters or **variables** can be overwritten by the user when running a query.

Variables are query parameters that are defined in the Query Designer and are filled with values when a query is executed. Variables are not specific to an InfoProvider, but rather on the InfoObject for which they were created.

Calculations can be specified in the Query Designer on key figures, using formulas with built-in functions, for example percentage functions.

For Power BI Desktop to be able to connect to a BEx query in SAP BW, the property *Allow External Access to this Query* must be set in the BEx Query Designer, as shown below.



Also included in the BEx tools is the **BEx Analyzer**. BEx Analyzer is an Add-On to Microsoft Excel and can be installed as part of the SAP Frontend Tools (SAPGUI) for BI. Data is embedded into Microsoft Excel workbooks providing analyst users with drag-and-drop capabilities to drill down and filter the data using SAP BI OLAP functions. Excel functions and features can be used for additional analysis and capabilities.

Power BI and SAP BW

SAP is a key enterprise BI data source, and Power BI connectivity to SAP systems is a desired capability for enterprise customers who want to unlock insights from their most critical business systems.

Most of SAP customers have deployed SAP BW and they have gone through the effort of integrating their transactional SAP data (from SAP ECC/ERP, CRM, SRM, or other business systems) into SAP BW. Enterprise data in SAP BW has been cleansed, transformed and modeled for use in reporting, analytics and planning applications. It only makes sense that customers want to utilize what they have already invested in and connect Power BI to their data in SAP BW.

Integration Architecture

From a technical point of view, the integration between Power BI Desktop and SAP BW is based on the so-called OLAP BAPIs (for Business Application Programming Interfaces).

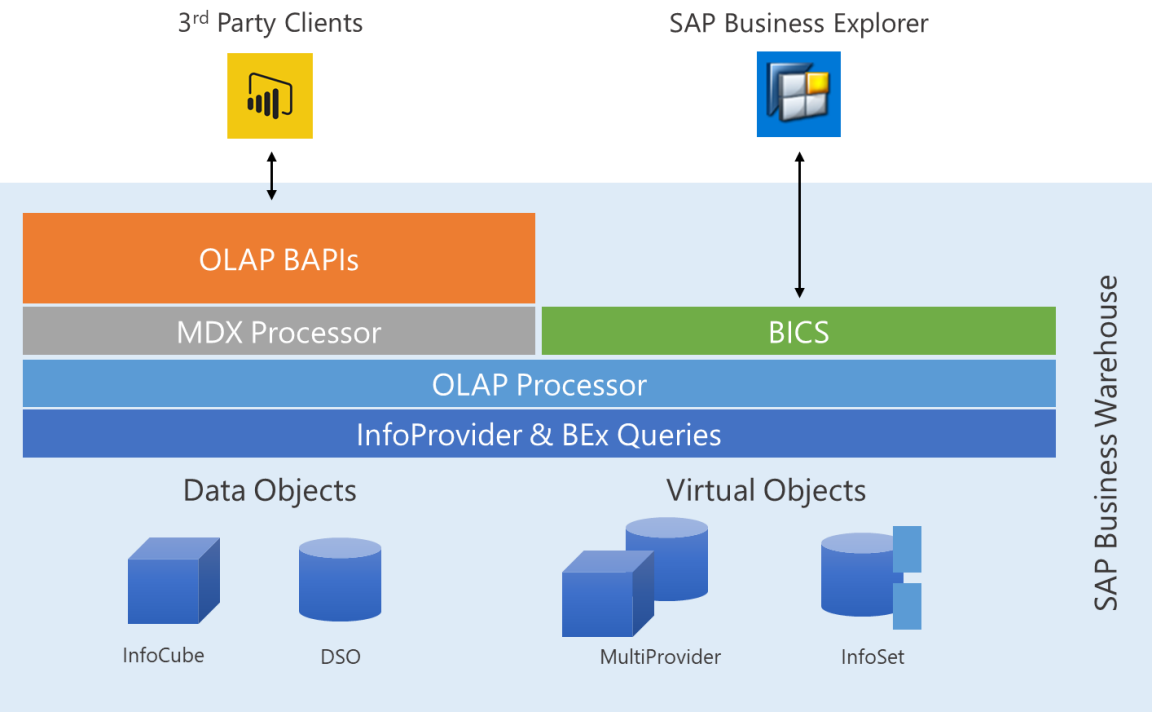
The OLAP BAPIs are delivered with SAP BW and provide 3rd-parties and developers with standardized interfaces that enables them to access the data and metadata of SAP BW with their own front-end tools.

Applications of all types can be connected with a SAP BW server using these methods.

The OLAP BAPIs are implemented in SAP BW as RFC-enabled function modules and are invoked by Power BI Desktop over SAP's RFC protocol. This requires the NetWeaver RFC Library or SAP .NET Connector to be installed on Power BI Desktop machine (also see Prerequisites section below).

The OLAP BAPIs provide methods for browsing metadata and master data, as well as for passing MDX statements for execution to the MDX Processor.

The OLAP Processor is responsible for retrieving, processing and formatting the data from the SAP BW source objects, which are further described below.



In contrast to 3rd party clients like Power BI Desktop, the SAP Business Explorer tools benefit from a more direct and performant interface to the SAP BW OLAP Processor. The Business Intelligence Consumer Services, or BICS interface, is the recommended way for the SAP BI tools to connect to the underlying data sources. BICS is not available for 3rd party tools.

SAP BW Data Sources

The OLAP BAPIs provide Power BI Desktop with access to BW InfoProviders and BEx Queries.

Typically, when a 3rd party tool like Power BI Desktop connects using the OLAP BAPIs, SAP BW first responds with a list of catalogs available in the SAP BW system.

There is one catalog with the technical name **$INFOCUBE** which contains all InfoProviders in the SAP BW system. This catalog is shown as a node in the Navigator of Power BI Desktop. By expanding this node in the Navigator, the Power BI user can select from the available InfoProviders in the SAP BW system.

The other catalogs represent InfoProviders for which at least one Query exists. By expanding one of these nodes in the Navigator, the Power BI user can select from the available queries that are associated with the InfoProvider.

BEx Queries offer some advantages and additional functionality to create customized data sources to meet end-user requirements. For example, you can parameterize queries with variables that can limit the data set to what is important to the end user. Or, you can recalculate key figures using formulas.

Although BEx Queries have advantages as data sources (also see Performance Considerations section below), customers do not need a Query for every report. Customers will need to weigh the cost of developing and maintaining additional Queries against their reporting requirements.

SAP BW Connector versions:

Starting with the June 2018 release of Power BI Desktop (and generally available with the October 2018 release), you can use the SAP BW connector with an implementation that has significant improvements in performance and capabilities. This updated version of the SAP BW connector was developed by Microsoft and is called Implementation 2.0. You can select either version 1 (v1) of the SAP BW Connector, or the Implementation 2.0 SAP Connector.

Implementation 2.0 is based on SAP .NET Connector. This implementation provides all the capabilities of Implementation 1.0 and more:

- Improved Performance
- Ability to retrieve several million rows of data, and fine tuning through the batch size option.
- Support for compressed mode. Especially beneficial for high-latency connections or large datasets.
- Improved detection of Date variables.
- Expose Date (ABAP type DATS) and Time (ABAP type TIMS) dimensions as dates and times respectively, instead of text values. Exposed in the “Key” column.
- Better exception handling: Errors that occur in SAP BAPI calls are now surfaced.
- Column folding in BasXml and BasXmlGzip modes.
- If the generated MDX query retrieves 40 columns but the current selection only needs 10, this request will be passed onto the server to retrieve a smaller dataset.
- Support for currency, units and formatted values.

The following sections describe the installation of each version, in turn. You can choose one or the other connector when connecting to SAP BW from Power BI Desktop.

We suggest you use the Implementation 2.0 SAP Connector whenever possible.

Installation of SAP BW Connector – Implementation 1.0

Prerequisites

In order to use the SAP BW Connector Implementation 1.0, the following requirements must be met:

The **NetWeaver library** must be installed on your local machine. You can get the SAP NetWeaver library from your SAP administrator, or directly from the SAP Software Download Center at https://support.sap.com/swdc. The SAP NetWeaver library is usually included also in the SAP Client Tools installation or SAPGUI installation.

Your SAP administrator can refer to SAP Note #1025361 for the download location of the most recent NetWeaver library version. Ensure that the architecture for the SAP NetWeaver library (32-bit or 64-bit) matches your Power BI Desktop installation, then install all files included in the SAP NetWeaver RFC SDK according to the SAP Note.

In Power BI Desktop, you can select the **File** tab, then **Help -> About** to find out the specific version you have installed.

Installation of SAP BW Connector – Implementation 2.0

Prerequisites

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's website using the following link:

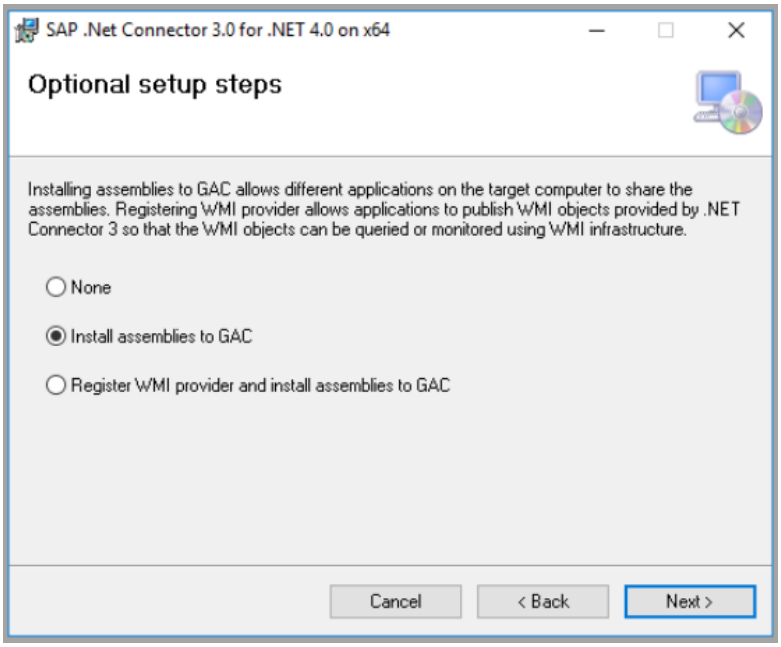
- SAP .NET Connector 3.0

Access to the download requires a valid S-user. Customers are encouraged to contact their SAP Basis team to get the SAP .NET Connector 3.0.

The connector comes in 32-bit and 64-bit versions, and users must choose the version that matches their Power BI Desktop installation. At the time of this writing, the website lists two versions (for .NET 4.0 framework):

- SAP Connector for Microsoft .NET 3.0.20.0 for Windows 32-bit (x86) as zip file (6.896 KB), January 16, 2018
- SAP Connector for Microsoft .NET 3.0.20.0 for Windows 64-bit (x64) as zip file (7.180 KB), January 16, 2018

When installing, in the **Optional setup steps** window, make sure you select the *Install assemblies to GAC* option as shown in the following image.

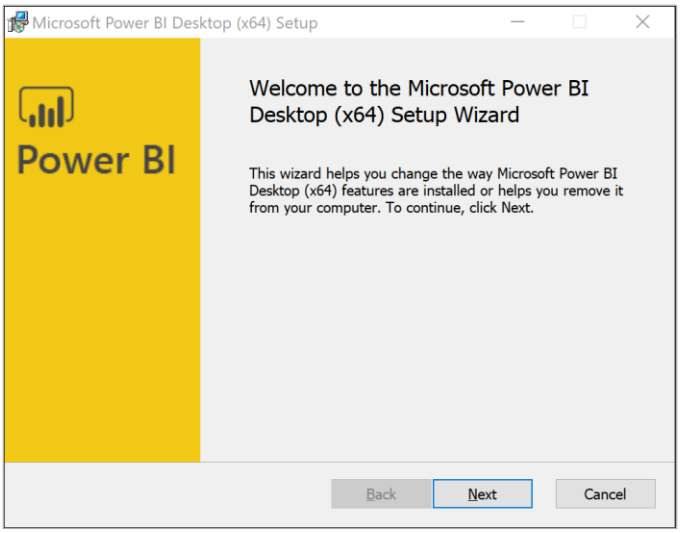


Note: The first version of SAP BW implementation required Netweaver DLLs; if you're using Implementation 2.0 of the SAP Connector, and not using the first version, the Netweaver DLLs are not required.

Power BI Desktop Installation Steps

The SAP BW Connector is included in the installation of Power BI Desktop.

1. Download the latest version of Power BI Desktop from http://powerbi.com and launch the Setup wizard (.msi).
2. Follow the steps in the Setup wizard to complete the installation.



Alternatively, you can install Power BI Desktop from the Windows Store.

Getting started with the SAP BW Connector

Connection Types for SAP Business Warehouse

Power BI Desktop offers two connection types for SAP Business Warehouse – **Application Server** connections and **Message Server** connections. When you create a connection from Power BI Desktop to an SAP BW server, the logon parameters that you need to provide are very similar to when an SAP user is logging on to SAP BW via the *SAPGUI* client tool.

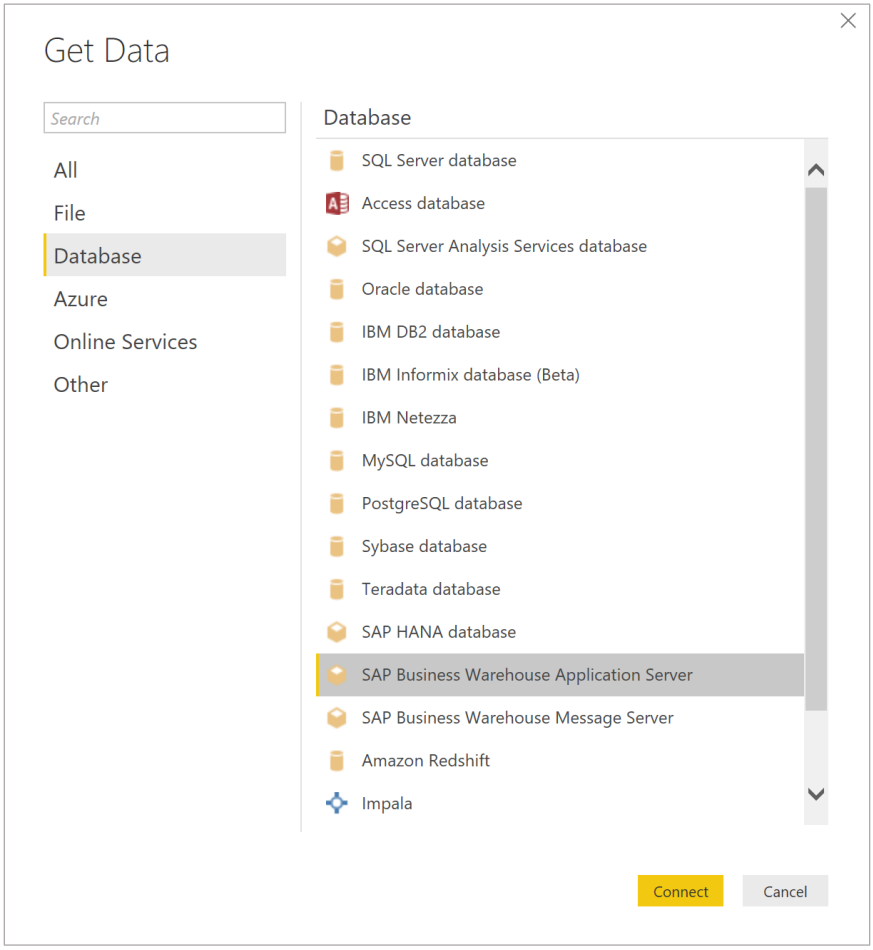
With the connection type *Application Server*, Power BI will always connect to the specified SAP BW server (hostname or IP address) in your SAP landscape. This requires the server to have sufficient resources available at any time to support the connection and process its workload, concurrently to all other connections or work processes that may exist at the same time.

With the connection type *Message Server*, Power BI will query a message server that has been configured in your SAP system landscape. The message server will check which BW application server has the best load statistics or the fewest users at that time, and Power BI will make a connection to the most favorable server based on that information. In addition to the message server, your SAP Administrator has configured a *Logon Group* which comprises two or more application servers that will participate in logon load balancing.

SAP BW Application Server Connection

1. Launch Power BI Desktop and close the splash screen that is shown initially.
2. From the **Home** tab, select **Get Data** -> **More**.

3. In the **Get Data** dialog, filter on **Database** connectors on the left, then select **SAP Business Warehouse Application Server** from the list.

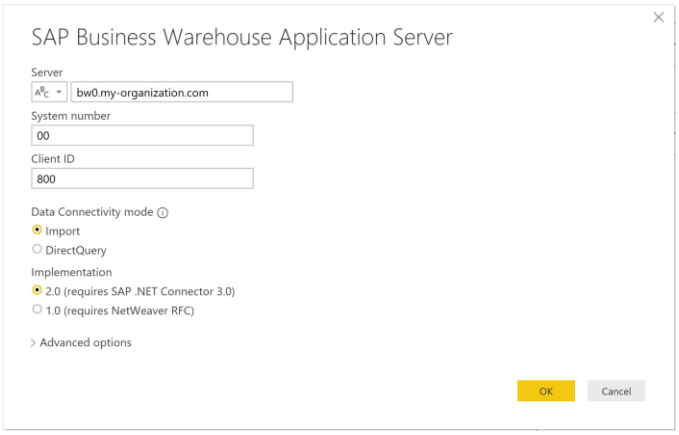


4. Click on **Connect**.
5. In the **SAP Business Warehouse Application Server** connection dialog, enter the following information:

   Server: The application server (hostname or IP address) for the SAP BW instance

   System number: The system number, or instance number of the SAP BW instance. This is typically a two-digit number.

   Client ID: The specific client in your SAP BW system that you want to connect to

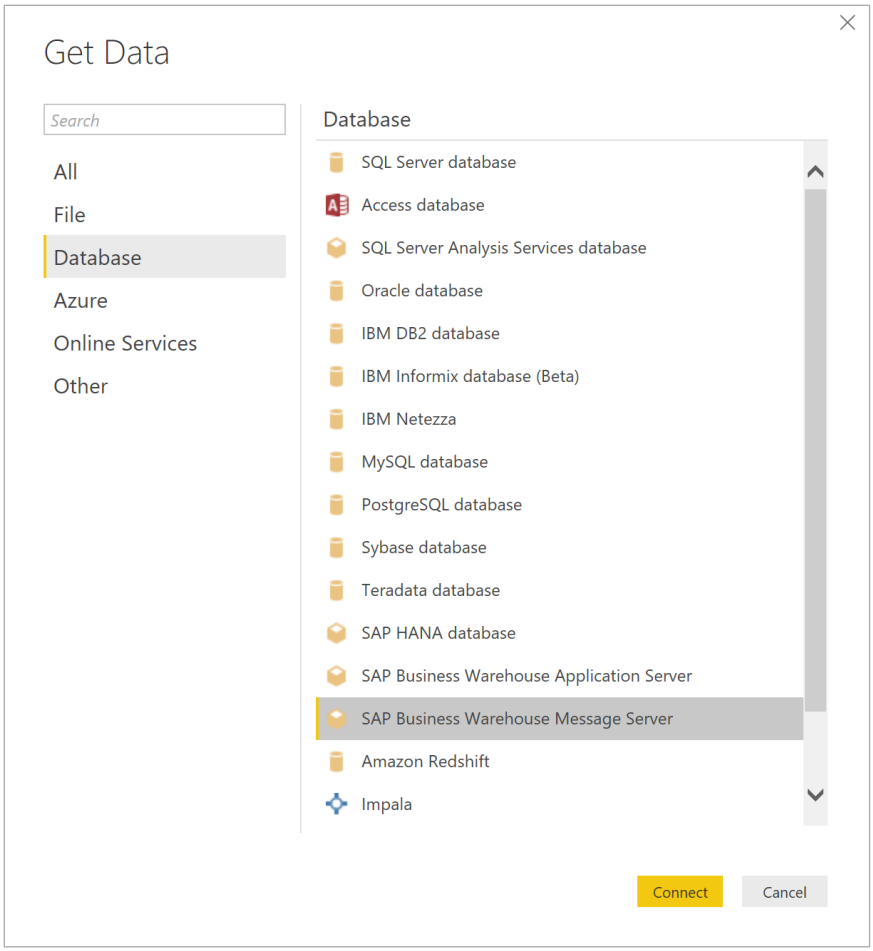


6. Leave the **Data Connectivity mode** on the default **Import** selection. We will examine this setting in more detail later in this document.
7. **2.0 is the default implementation**. Our suggestion is to use Implementation 2.0. Change it if needed.
8. Click **OK**.

SAP BW Message Server Connection

Alternatively, you can create a connection to an SAP BW Message Server to take advantage of logon load balancing in your SAP BW system landscape.

1. Launch Power BI Desktop and close the splash screen that is shown initially.
2. From the **Home** tab, select **Get Data** -> **More**.
3. In the **Get Data** dialog, filter on **Database** connectors on the left, then select **SAP Business Warehouse Message Server** from the list.



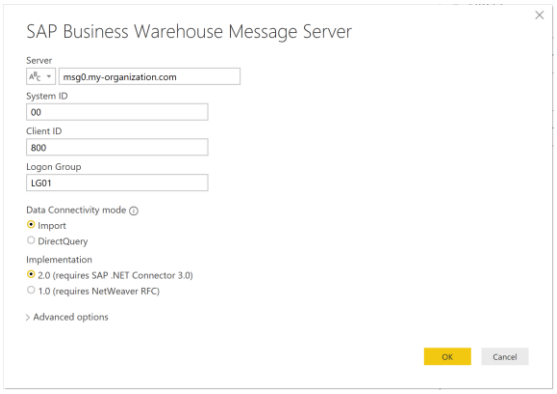
4. Click on **Connect**.
5. In the **SAP Business Warehouse Message Server** connection dialog, enter the following information:

   Server: The message server (hostname or IP address) configured in your SAP system landscape

   System ID: The system ID of the message server. This is different from *System Number* and is typically a three-character ID.

   Client ID: The specific client in the message server that you want to connect to

   Logon Group: The logon group of application servers that was configured by your SAP Administrator



6. Leave the **Data Connectivity mode** on the default **Import** selection. We will examine this setting in more detail later in this document.
7. **2.0 is the default implementation**. Our suggestion is to use Implementation 2.0. Change it if needed.
8. Click **OK**.

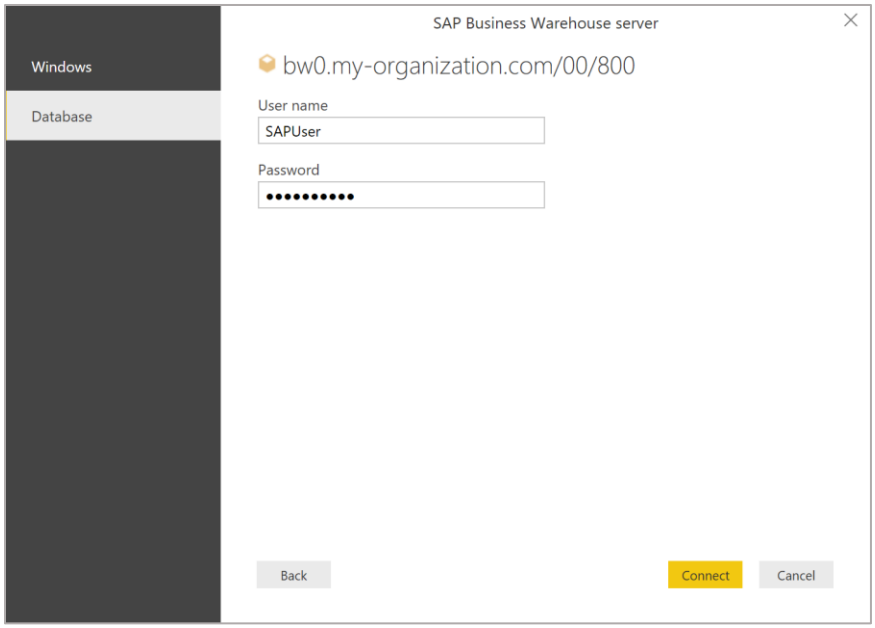
Authentication Methods for SAP BW Connections

After you enter the logon parameters for your SAP BW application or message server, you will be prompted for credentials to authenticate your connection.

Power BI offers two authentication modes for SAP BW connections – *user name / password* authentication, and *Windows* authentication (single sign-on). SAML authentication is not supported at the time of publishing of this document.

User Name / Password Authentication

1. In the **SAP Business Warehouse server** dialog, select the **Database** tab on the left (default).
2. Enter your SAP BW **User name** and **Password**.



3. Click **Connect**.

Windows Authentication and Single Sign-On

For Windows-based authentication and Single Sign-On functionality, your SAP BW server must be configured for logon using SNC (Secure Network Communications). SNC is a mechanism provided by the SAP system that enables application-level security on data exchanged between a client, such as Power BI, and the SAP application server. SNC works with different external security products and offers features that the SAP system does not directly provide, including Single Sign-On.

In addition to your SAP BW server being configured for SNC logon, your SAP user account needs to be configured with an SNC name (transaction SU01 in your SAP system).

For more detailed information please see Secure Network Communication, and the chapter *Single Sign-On Configuration* in this document.

Secure Login is a software solution by SAP that allows customers to benefit from the advantages of SNC without having to set up a public-key infrastructure (PKI). Secure Login allows users to authenticate with Windows Active Directory credentials.

Secure Login requires the installation of the Secure Login Client on your Power BI Desktop machine. The installation package is named SAPSetupSCL.EXE and can be obtained from the SAP Service Marketplace (requires SAP customer credentials).

For further information please see Secure Login.

1. In the **SAP Business Warehouse server** dialog, select the **Windows** tab on the left.
2. Select to either use your current Windows credentials or specify alternate Windows credentials.
3. Enter the **SNC Partner Name**. This is the configured SNC name in the SAP BW application server's security token. You can retrieve the SNC name with transaction **RZ11** (Profile Parameter Maintenance) in SAPGUI and parameter name **snc/identity/as**.

   For X.509 certificate security tokens, the format is:

   p:<X.509 Distinguished Name>

   Example (values are case sensitive!): **p:CN=BW0, OU=BI, O=MyOrg, C=US**

   For Kerberos security tokens, the format is:

   p:CN=<service_User_Principal_Name>

   Example (values are case sensitive!): **p:CN=SAPServiceBW0@BWSERVER.MYORG.COM**

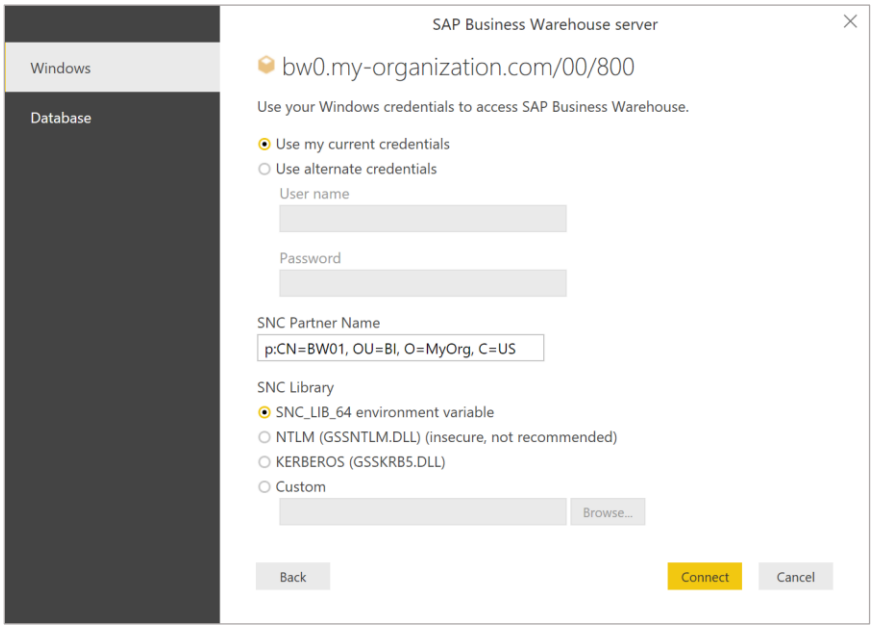
4. Select the **SNC Library** that **your** SAP BW environment has been configured for.

   The option **SNC_LIB** or **SNC_LIB_64** will check the corresponding environment variable on your machine and use the DLL that is specified there.

   The options **NTLM** and **KERBEROS** will expect the corresponding DLL to be in a folder that has been specified in the PATH variable on your local machine. The libraries for 32bit systems are GSSNTLM.DLL (for NTLM) and GSSKRB5.DLL (for Kerberos). The libraries for x64 systems are GX64NTLM.DLL (for NTLM) and GX64KRB5.DLL (for Kerberos).

   The option **Custom** allows for the use of a custom developed library.

   Please validate the settings with your SAP Administrator.



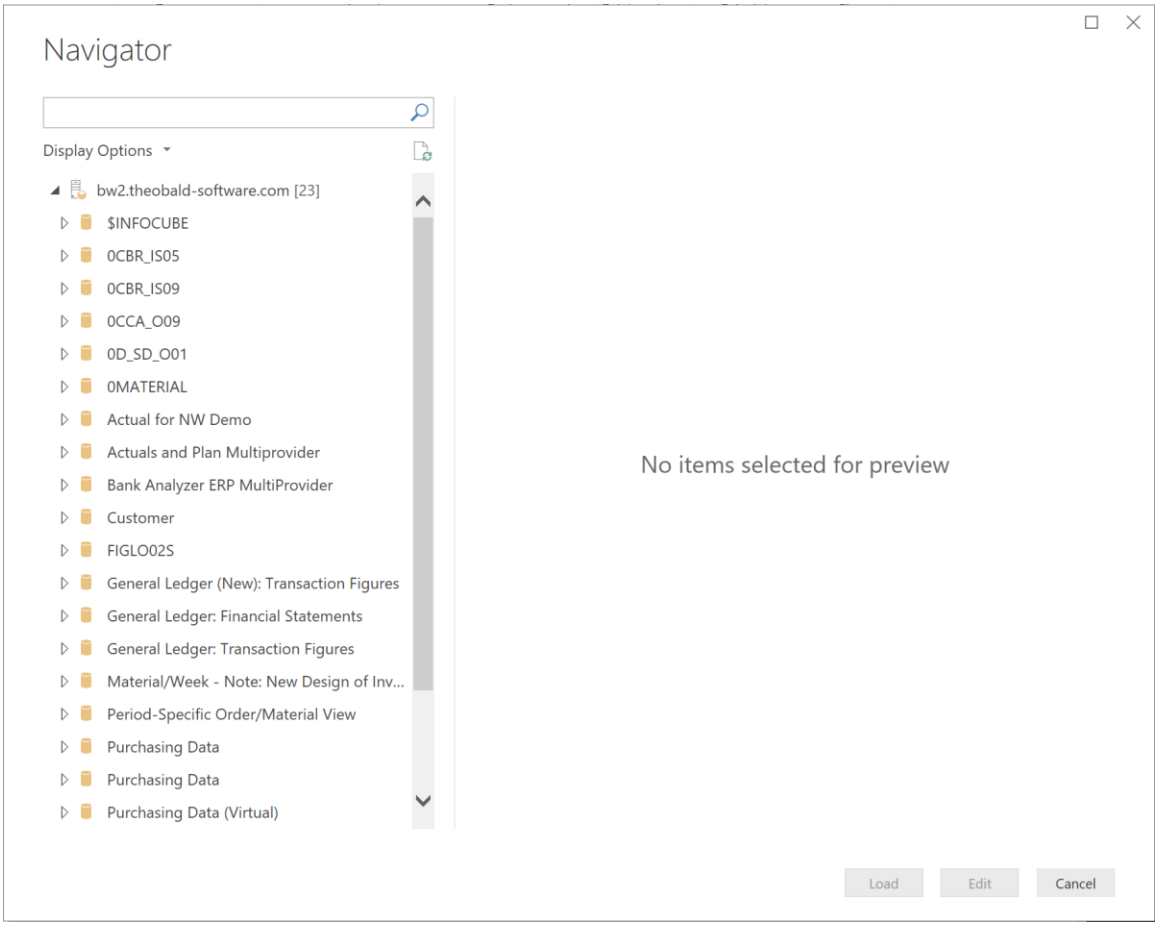
5. Click **Connect**.

Navigating the Query Objects in SAP BW

After you connect to your SAP BW instance, the **Navigator** dialog will be shown, with a list of available catalogs in your SAP BW instance.

You will see one catalog folder with the name **$INFOCUBE**. This folder contains all InfoProviders in the SAP BW system.

The other catalog folders represent InfoProviders in SAP BW for which at least one query exists.



The Navigator will display a hierarchical tree of data objects from the connected SAP BW system. The types of objects are listed here:

Your SAP BW application server

Catalog - either **$INFOCUBE** or an InfoProvider

InfoCube or a BEx Query

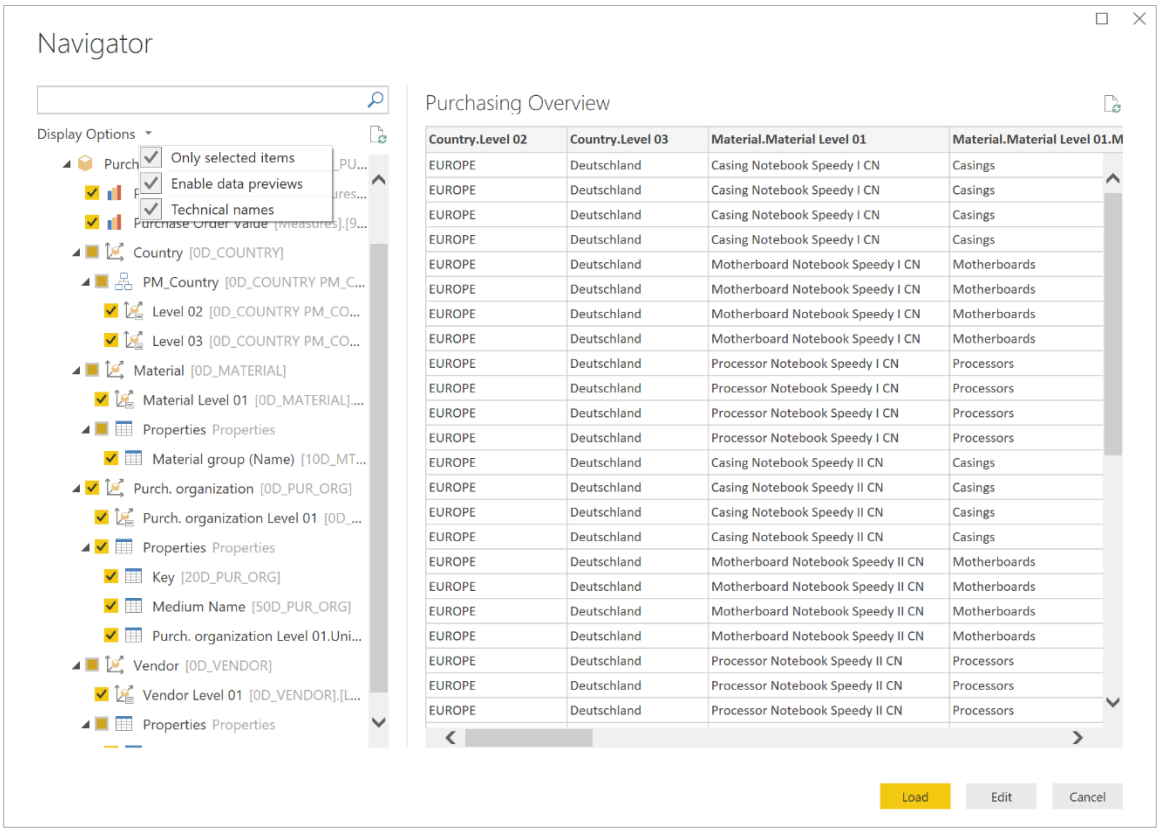
Key figure

Characteristic

Characteristic Level

Property (Attribute)

Hierarchy



Country.Level 02	Country.Level 03	Material.Material Level 01	Material.Material Level 01.M
EUROPE	Deutschland	Casing Notebook Speedy I CN	Casings
EUROPE	Deutschland	Casing Notebook Speedy I CN	Casings
EUROPE	Deutschland	Casing Notebook Speedy I CN	Casings
EUROPE	Deutschland	Casing Notebook Speedy I CN	Casings
EUROPE	Deutschland	Motherboard Notebook Speedy I CN	Motherboards
EUROPE	Deutschland	Motherboard Notebook Speedy I CN	Motherboards
EUROPE	Deutschland	Motherboard Notebook Speedy I CN	Motherboards
EUROPE	Deutschland	Motherboard Notebook Speedy I CN	Motherboards
EUROPE	Deutschland	Processor Notebook Speedy I CN	Processors
EUROPE	Deutschland	Processor Notebook Speedy I CN	Processors
EUROPE	Deutschland	Processor Notebook Speedy I CN	Processors
EUROPE	Deutschland	Processor Notebook Speedy I CN	Processors
EUROPE	Deutschland	Casing Notebook Speedy II CN	Casings
EUROPE	Deutschland	Casing Notebook Speedy II CN	Casings
EUROPE	Deutschland	Casing Notebook Speedy II CN	Casings
EUROPE	Deutschland	Casing Notebook Speedy II CN	Casings
EUROPE	Deutschland	Motherboard Notebook Speedy II CN	Motherboards
EUROPE	Deutschland	Motherboard Notebook Speedy II CN	Motherboards
EUROPE	Deutschland	Motherboard Notebook Speedy II CN	Motherboards
EUROPE	Deutschland	Motherboard Notebook Speedy II CN	Motherboards
EUROPE	Deutschland	Processor Notebook Speedy II CN	Processors
EUROPE	Deutschland	Processor Notebook Speedy II CN	Processors
EUROPE	Deutschland	Processor Notebook Speedy II CN	Processors

Objects below an InfoCube or BEx Query node, such as the key figures, characteristics and properties, are only shown in **Import** connectivity mode, not in **DirectQuery** mode (further described below).

You can select from different **Display Options** to view the available query objects in SAP BW:

Only selected items: This option limits the objects shown in the list to just the selected items. By default, all query objects are displayed. This option is useful for a review of the objects that you included in your query.

Enable data previews (default behavior): This option allows you to control whether a preview of the data should be displayed on the right-hand side in the **Navigator**. Disabling data previews reduces the amount of server interaction and response time. Data preview is only available in **Import** connectivity mode.

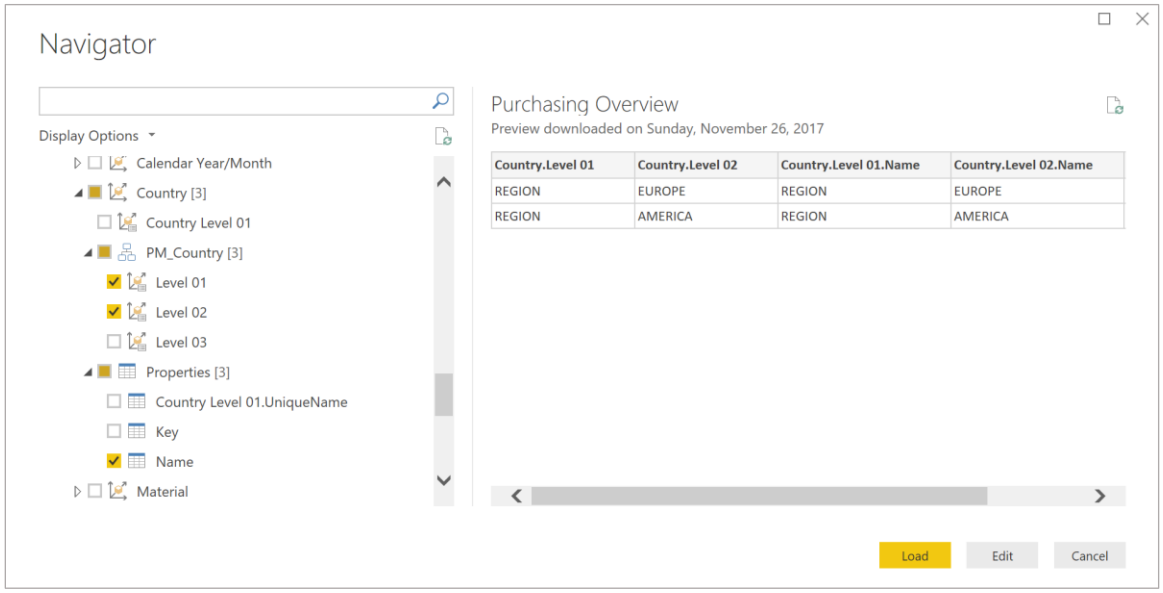
Technical names: SAP BW supports the notion of technical names for query objects, as opposed to the descriptive names that are shown by default. Technical names uniquely identify an object within SAP BW. With the option selected, the technical names will appear next to the descriptive name of the object.

Characteristic Hierarchies

A characteristic will always have at least one characteristic level (Level 01), even when no hierarchy is defined on the characteristic. The Characteristic Level 01 object contains all members for the characteristic as a flat list of values.

Characteristics in SAP BW can have more than one hierarchy defined. For those characteristics, the user can only select one hierarchy or the Level 01 object.

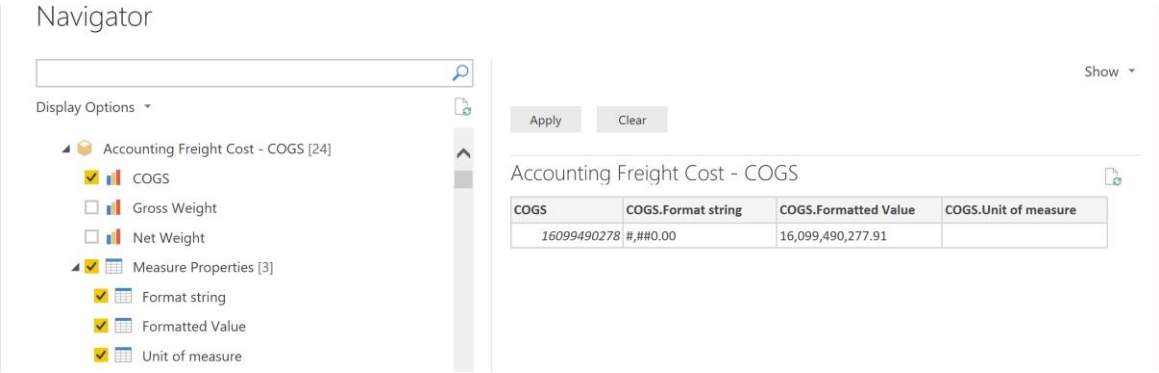
For characteristics with hierarchies, the properties selected for that characteristic will be included for each selected level of the hierarchy.



Measure Properties

When you pick a Measure in **Import mode**, you have an option to select the **Units/Currency, Formatted value, Format string**. In the screenshot below, it is useful to get the Formatted value for COGS. This helps us follows the same formatting standard across all the reports.

Note: Measure properties is not available in Direct Query mode.



Flattening of Multi-Dimensional Data

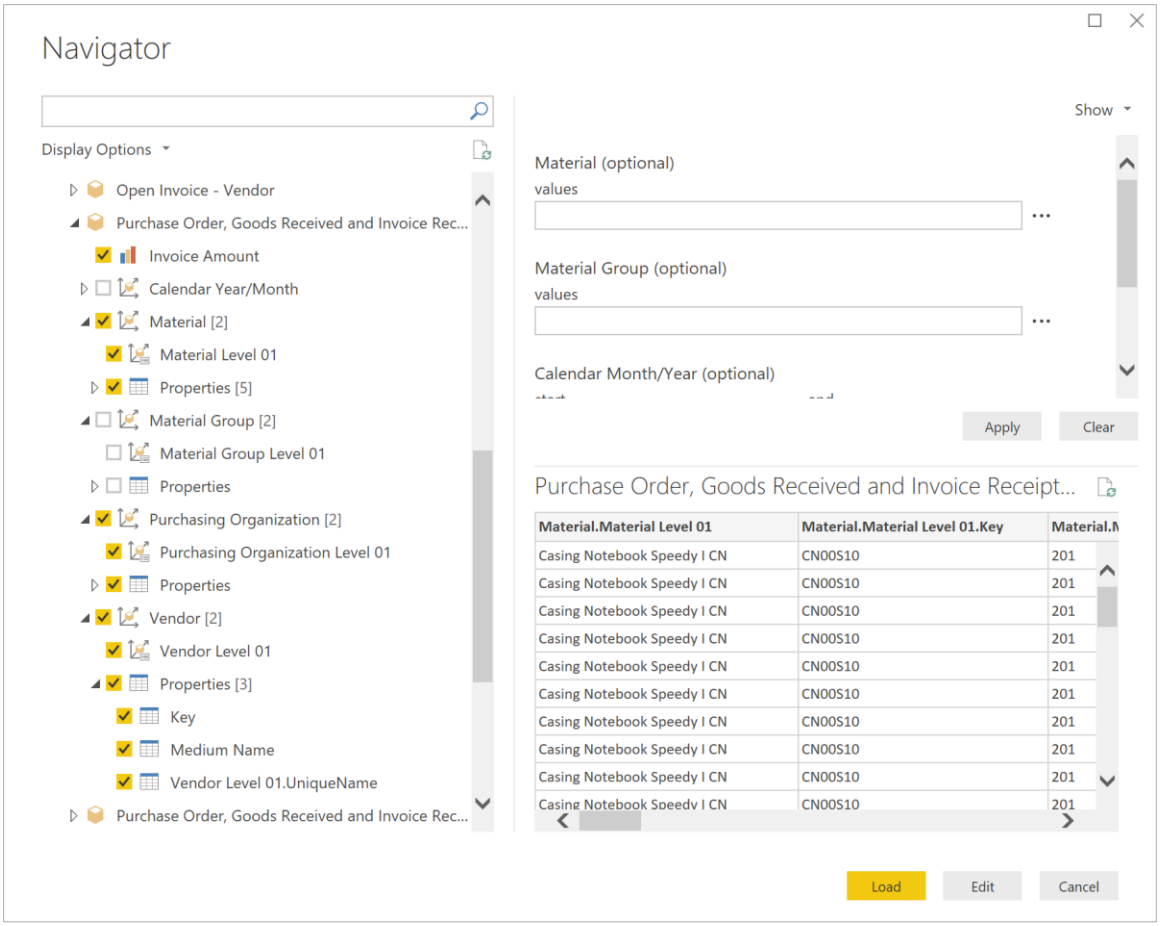
Based on the selected objects and properties in the Navigator screen, Power BI Desktop constructs an MDX statement that is sent for execution to SAP BW. The MDX statement returns a flattened data set that can be loaded into Power BI or further manipulated using the Query Editor.
Power BI uses a newer interface that is available in SAP BW version 7.01 or higher. The interface reduces memory consumption and the result set is not restricted by the number of cells.
The flattened data set is aggregated in SAP BW at the level of the selected characteristics and properties.
Even with these improvements, the resulting dataset can become very large and time-consuming to process.

Performance recommendation: Only include the characteristics and properties that you ultimately need in your Power BI report or dashboard. Aim for higher levels of aggregation, e.g. do you need Material-level details in your report, or is MaterialGroup-level enough? What hierarchy levels are required in Power BI? Try to create smaller datasets, with higher levels of aggregation, or multiple smaller datasets, that can be subsequently joined together in Power BI.

Query Parameters

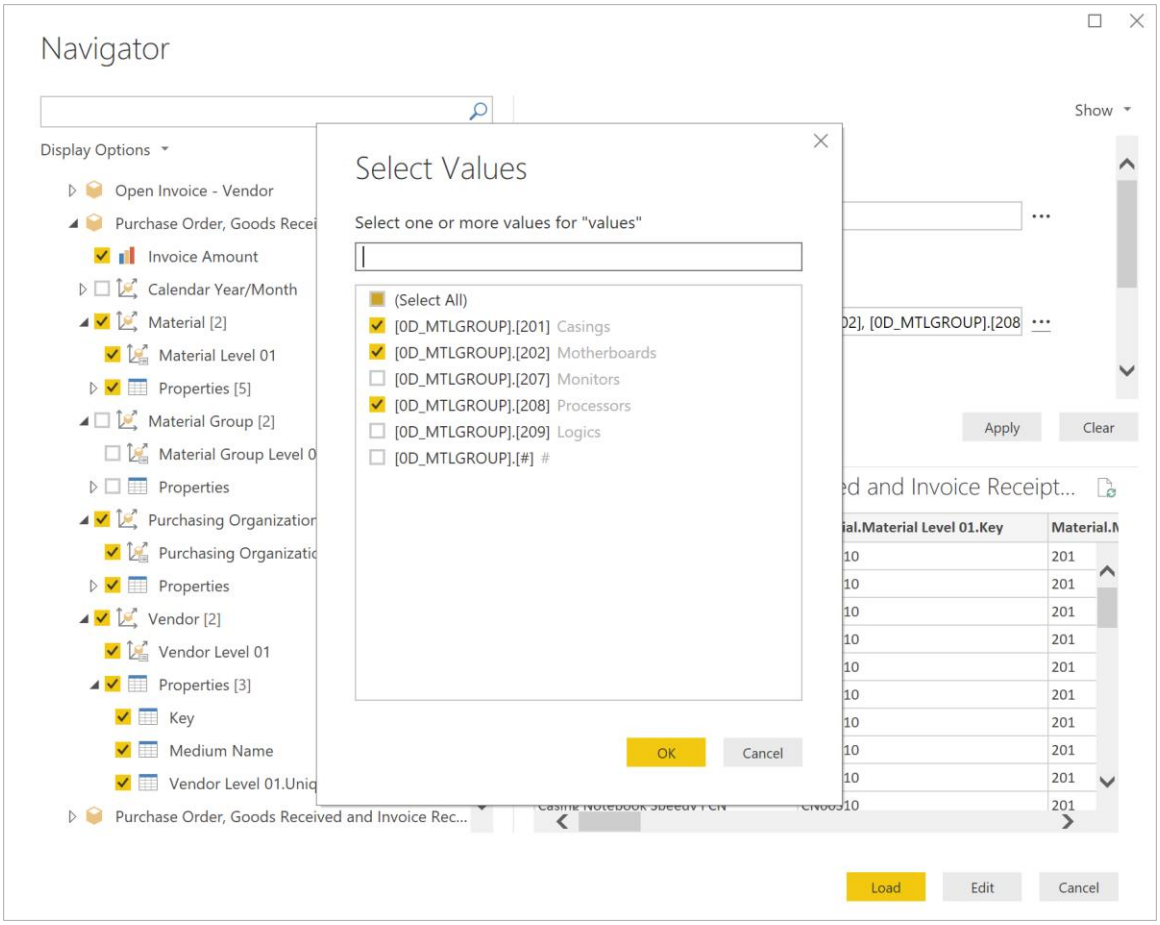
Queries in BW can have dynamic filters defined which allow the end user to restrict the data set that is returned by the query. In the BEx Query Designer, this type of dynamic filter can be defined with what is called a *Characteristic Restriction* and assigning a *Variable* to that restriction. Variables on a query can be required or optional and they are available to the user in the Power BI Desktop Navigator.

When you select a BW query with characteristic restrictions in the Navigator, you will see the variables displayed as **Parameters** above the data preview area.



Using the **Show** selector, you can display all parameters that are defined on the query, or just the required ones.

The query shown here has several optional parameters, including one for Material Group.
You can select one or more Material Groups to only return purchasing information for the selected values, e.g. Casings, Motherboards and Processors. You can also type the values directly into the **values** field. For variables with multiple entries, comma separated values are expected, in this example it would look like *[0D_MTLGROUP].[201], [0D_MTLGROUP].[202], [0D_MTLGROUP].[208]*.
The value **#** means unassigned, in the example any data record without a Material Group value assigned.



When parameter values are applied, the data shown in the Preview is filtered based on the parameter selection.

Performance recommendation: Filters based on parameter values get processed in the SAP BW data source, not in Power BI. This can have performance advantages when loading or refreshing SAP BW data into Power BI, for larger datasets. The time it takes to load data from SAP BW into Power BI increases with the size of the dataset, i.e. the number of columns and rows in the flattened result set. To reduce the number of columns, only select the key figures, characteristics and properties in the Navigator that you eventually want to see in your report or dashboard.
Similarly, to reduce the number of rows, use the available parameters on the query to narrow the dataset, or to split up a larger dataset into multiple, smaller datasets that can be joined together in the Power BI Desktop data model.
In many cases it may also be possible to work with the author of the BEx Query in SAP BW to clone and modify an existing query and optimize it for performance by adding additional characteristic restrictions or removing unnecessary characteristics.

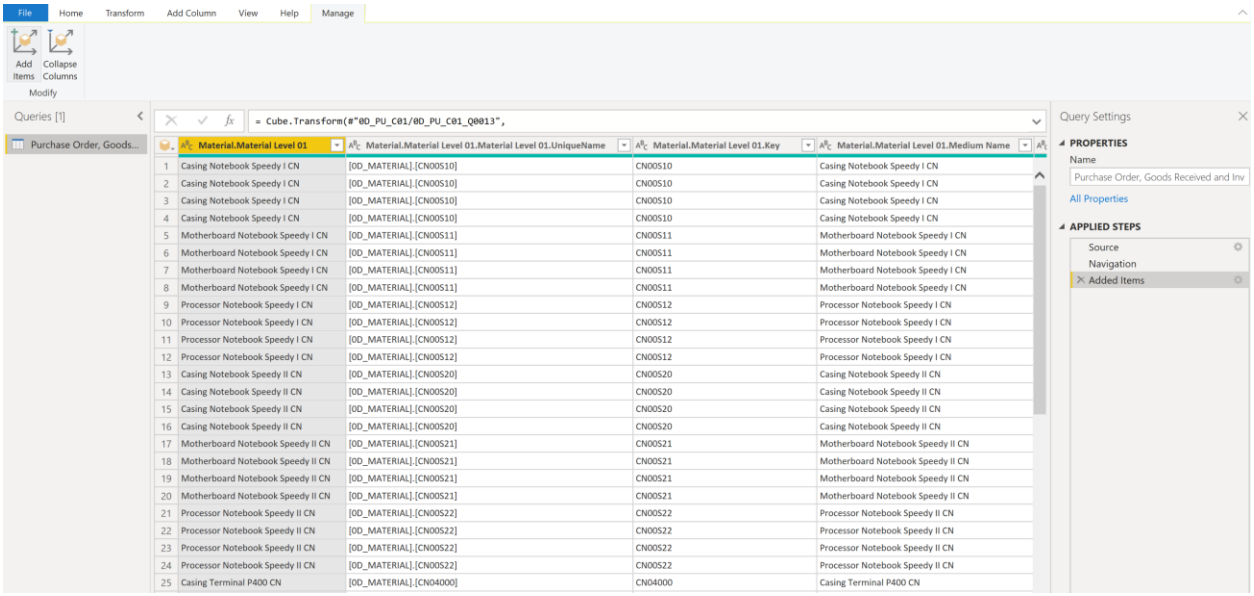
Loading SAP Data into Power BI Desktop

Once you have selected the desired SAP data set in the Navigator, you can import the data into Power BI Desktop in two ways:

Select **Load** to bring the entire data set from SAP BW into the Power BI Desktop data model. Power BI Desktop will take you to the **Report** view where you can begin visualizing the data or making further modifications using the **Data** or **Relationships** views.

Select **Edit** to launch the **Query Editor**, where you can perform additional data transformation and filtering steps before the entire data set from SAP BW is brought into the Power BI Desktop data model.

In the example above, a parameter was used to only bring back records with a Material Group of *Casings*, *Motherboards* and *Processors*.



In addition to importing data from SAP BW, you can also import data from a wide range of other data sources into Power BI Desktop, and combine them into a single report. For performance reasons, as outlined above, you can also split a larger SAP BW data set into smaller datasets and join them together in Power BI Desktop. Or, using the built-in connector for SAP HANA, you can combine bring together data from different SAP sources. This opens many interesting scenarios for reporting and analytics on top of your SAP data.

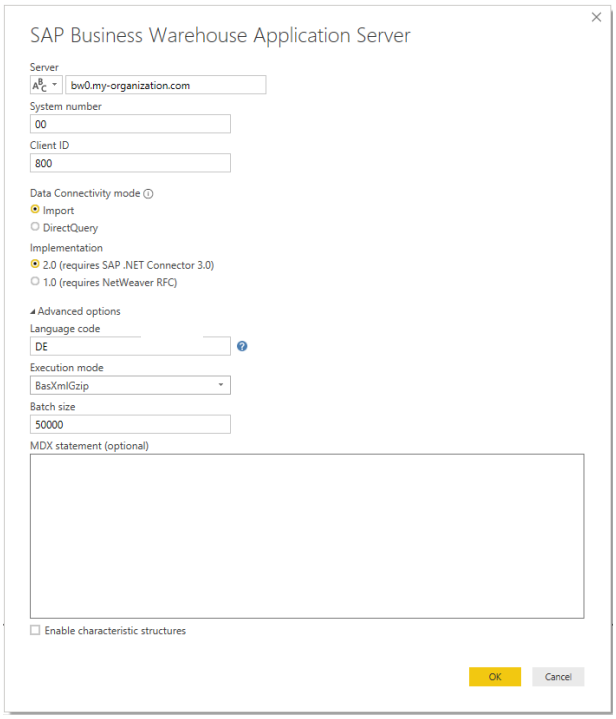
Additional topics for working with SAP BW data in the Power BI Desktop Query Editor are described in the following chapter.

Advanced Topics

When you create a connection to an SAP Business Warehouse Application Server or an SAP Business Warehouse Message Server you can optionally specify a *Language Code, Execution mode, Batch size* or an *MDX Statement*.

Logon Language Code

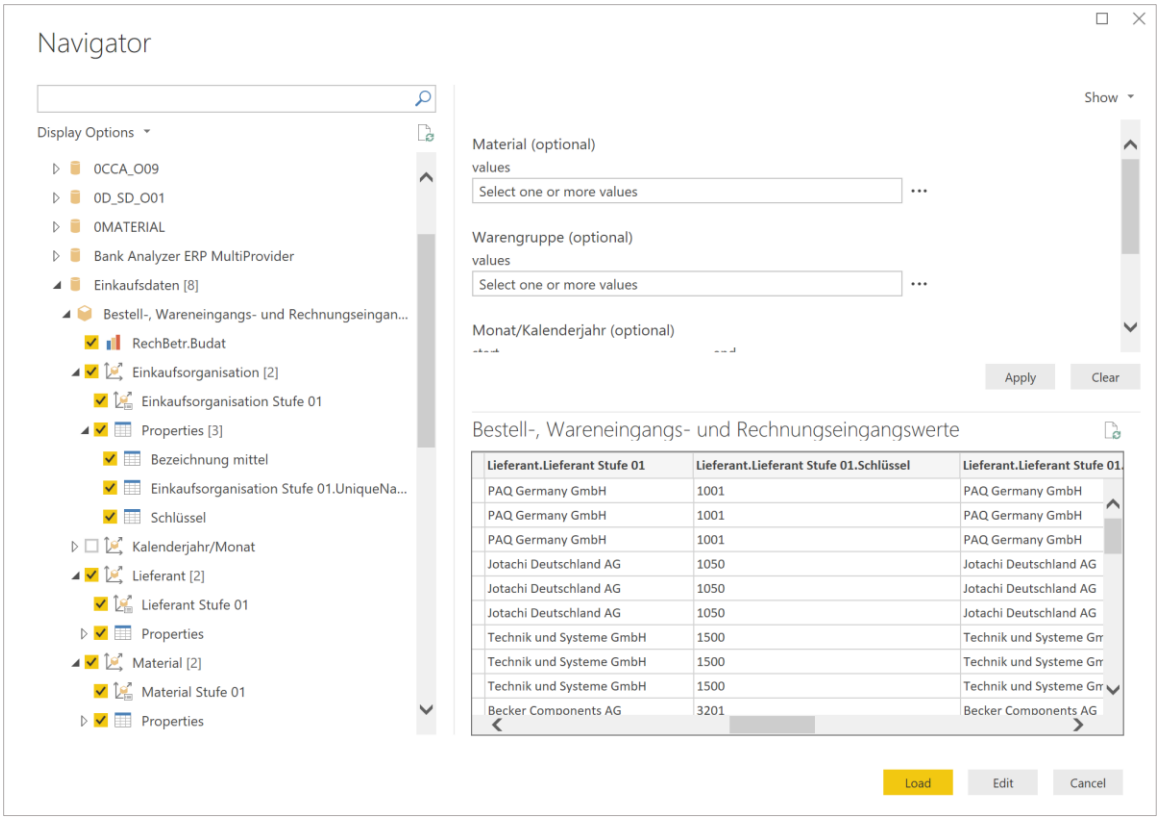
Users can optionally specify a language code when establishing a connection to the SAP BW server.



The expected value is a two-letter language code as defined in the SAP system. For a list of valid values, click the Help icon (question mark) next to the *Language Code* field.

As a result, Power BI Desktop will display the descriptive names of the data objects in SAP BW in the specified language, including the field names for the selected objects.

Please note that not all languages listed may be configured in your SAP BW system or object descriptions may not be translated in all languages.

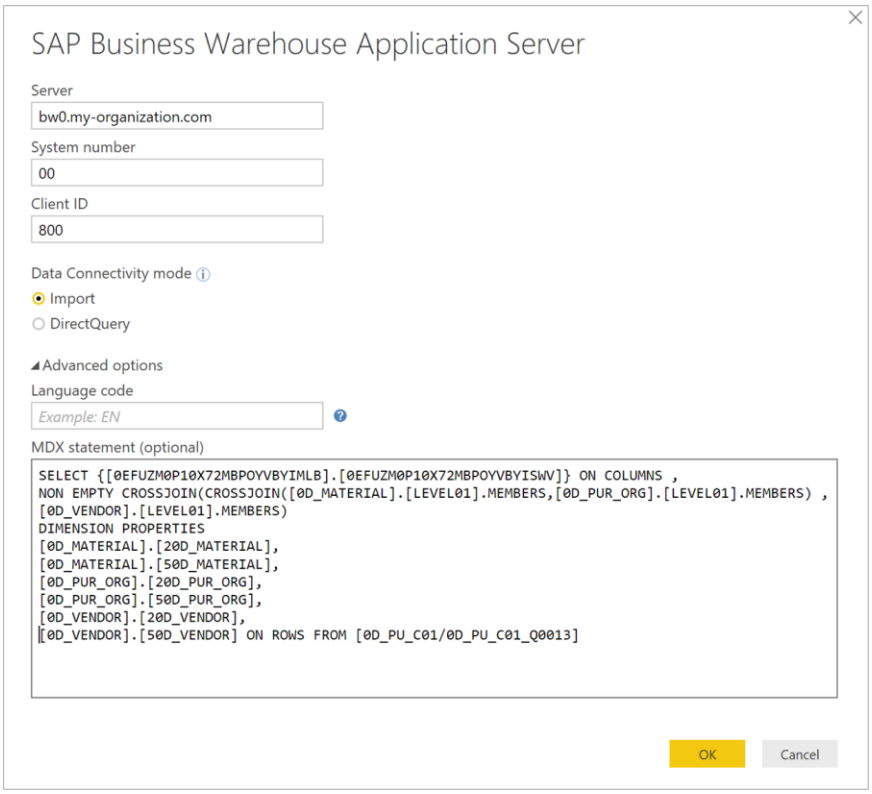


If no language code is specified, the System Locale of the current user will be used and mapped to a valid SAP language code.

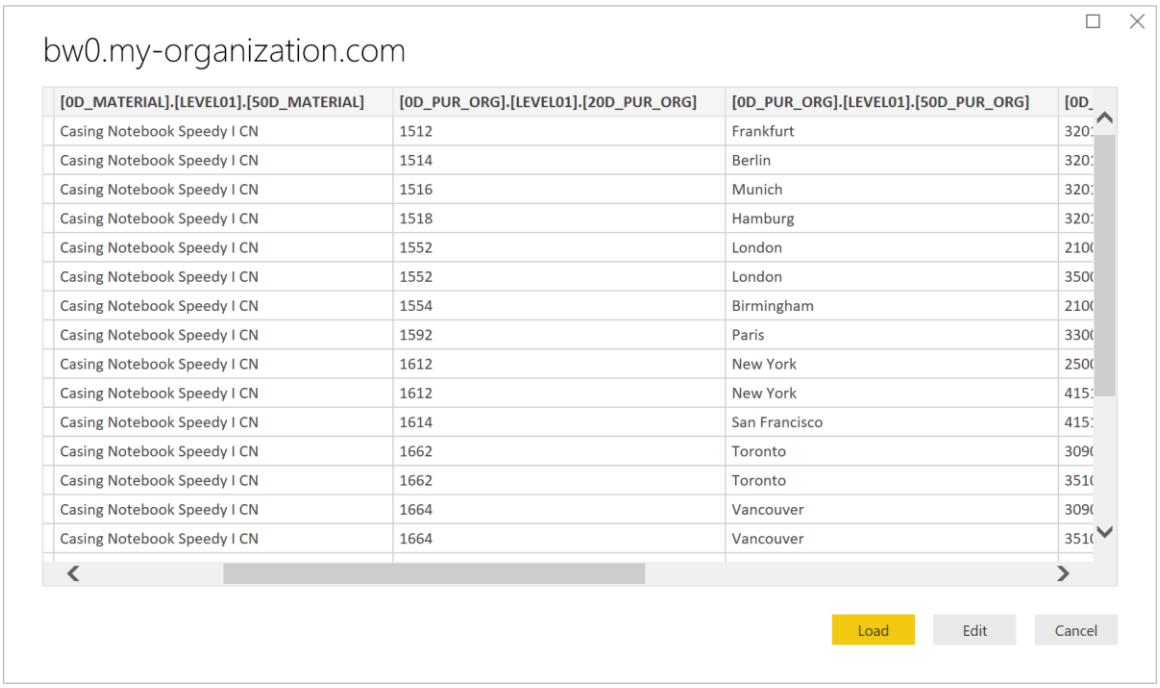
MDX Statement

Instead of using the Navigator to browse through and select from available data objects in SAP BW, the user who is familiar with the MDX query language can specify an MDX statement for direct execution in SAP BW.

The statement for the example used throughout this document would look as shown below, based on the technical names of the objects and properties in SAP BW.

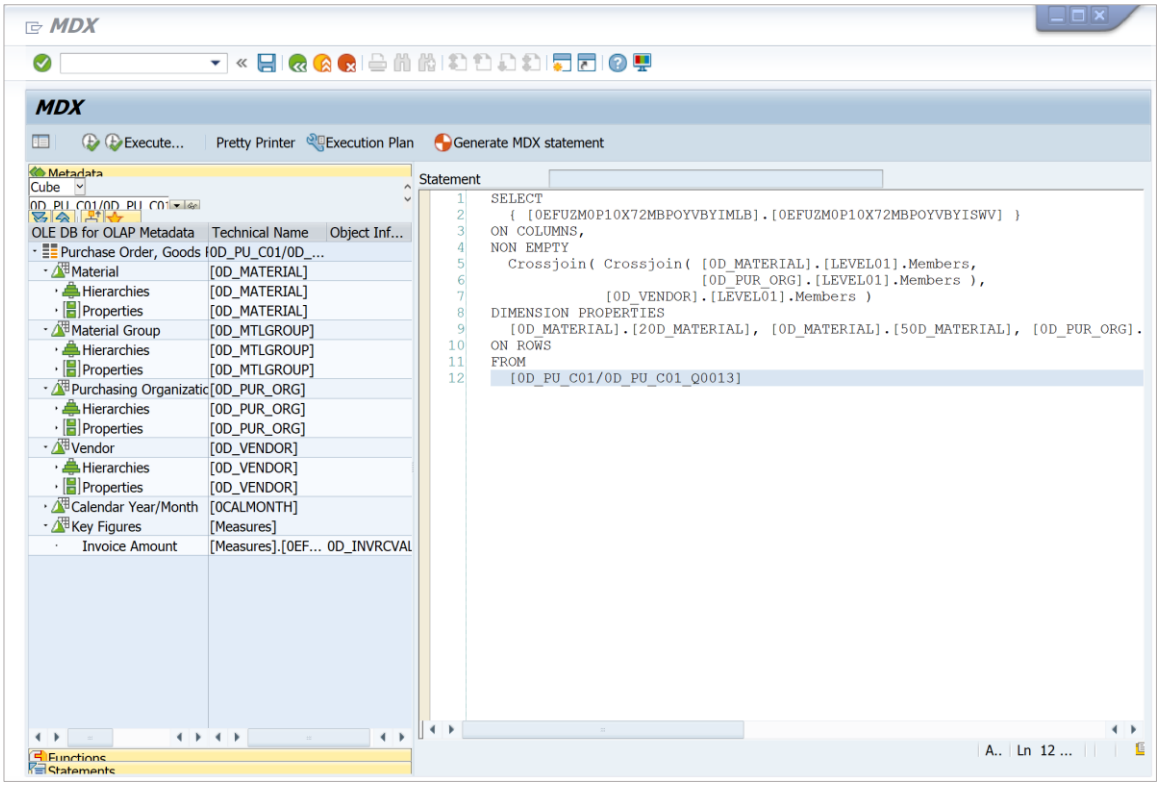


Power BI Desktop will display a preview of the data that is returned by the MDX statement and the user can select to load the data or further manipulate the data set in the Query Editor.



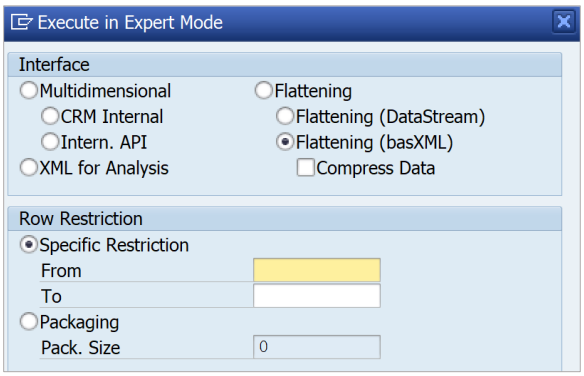
To validate and troubleshoot an MDX statement, SAP BW provides the transaction *MDXTEST* for SAPGUI users. Further, the MDXTEST transaction can be a useful tool for analyzing server errors or performance concerns as a result of processing that occurs within the SAP BW system.

For more detailed information on this transaction, please see MDX Test Environment.

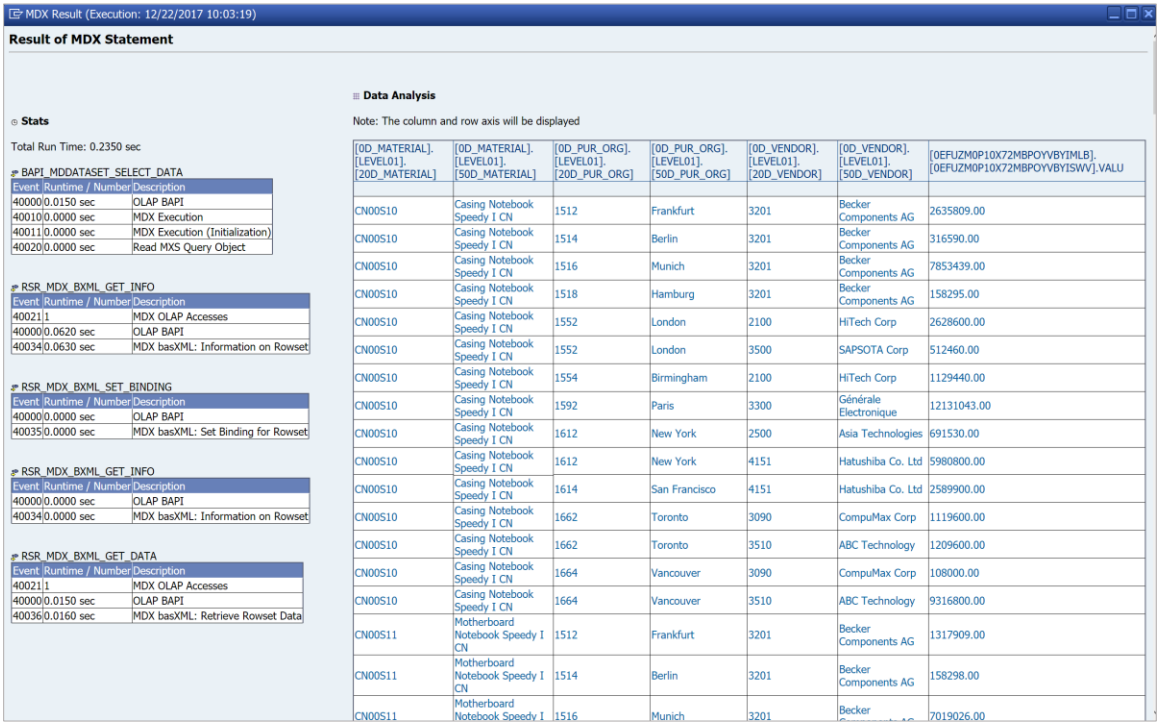


MDXTEST can also be used to construct an MDX statement. The transaction screen includes panels on the left that assist the user in browsing to a query object in SAP BW and generating an MDX statement.

The transaction offers different execution modes/interfaces for the MDX statement. Select *Flattening (basXML)* to mimick how Power BI would execute the query in SAP BW. As briefly described earlier in this document, this interface in SAP BW creates the row set dynamically using the selections of the MDX statement. The resulting dynamic table that is returned to Power BI Desktop has a very compact form that reduces memory consumption.



The transaction will display the result set of the MDX statement as well as useful runtime metrics.

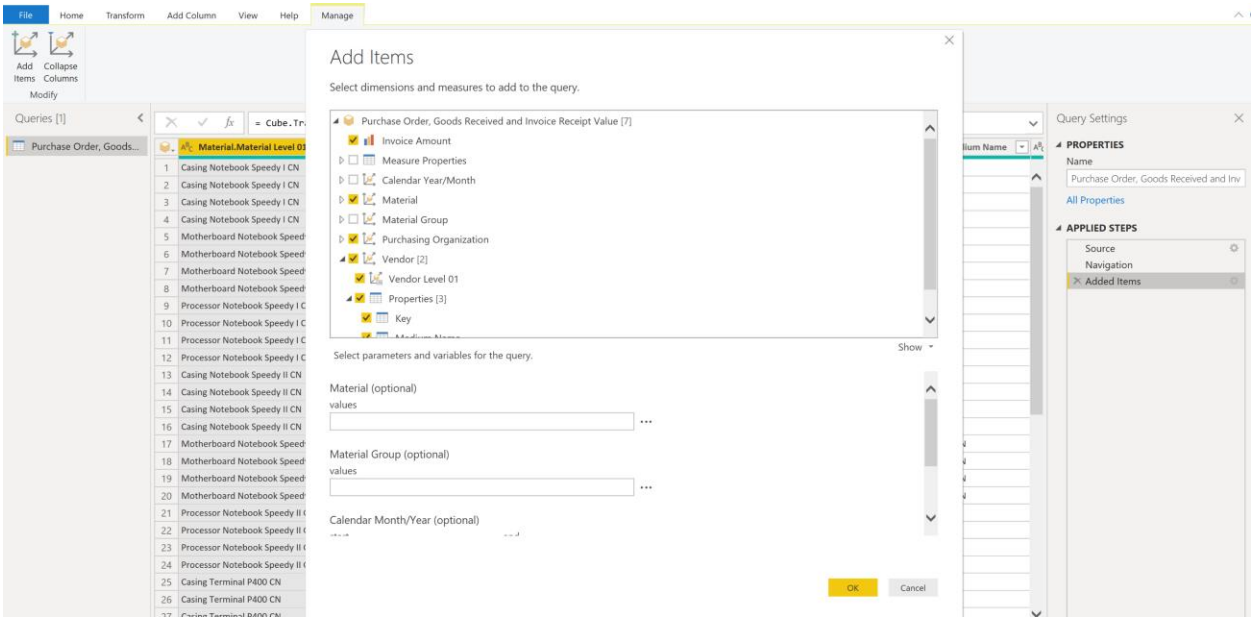


Query Editor in Power BI Desktop

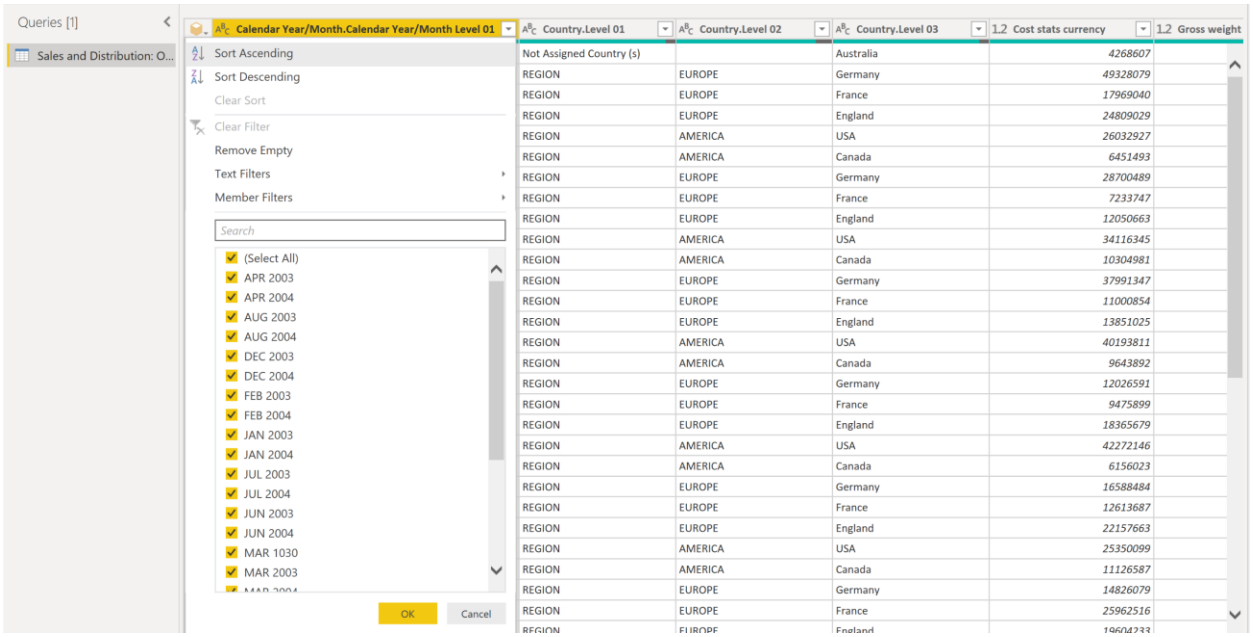
With the Query Editor in Power BI Desktop you can apply additional data transformations and filtering steps before you bring the dataset from SAP BW into the Power BI Desktop data model.

In the *Query Settings* panel on the right, the *Applied Steps* for query will be shown. To modify or review a step, click the gear icon next to a step.

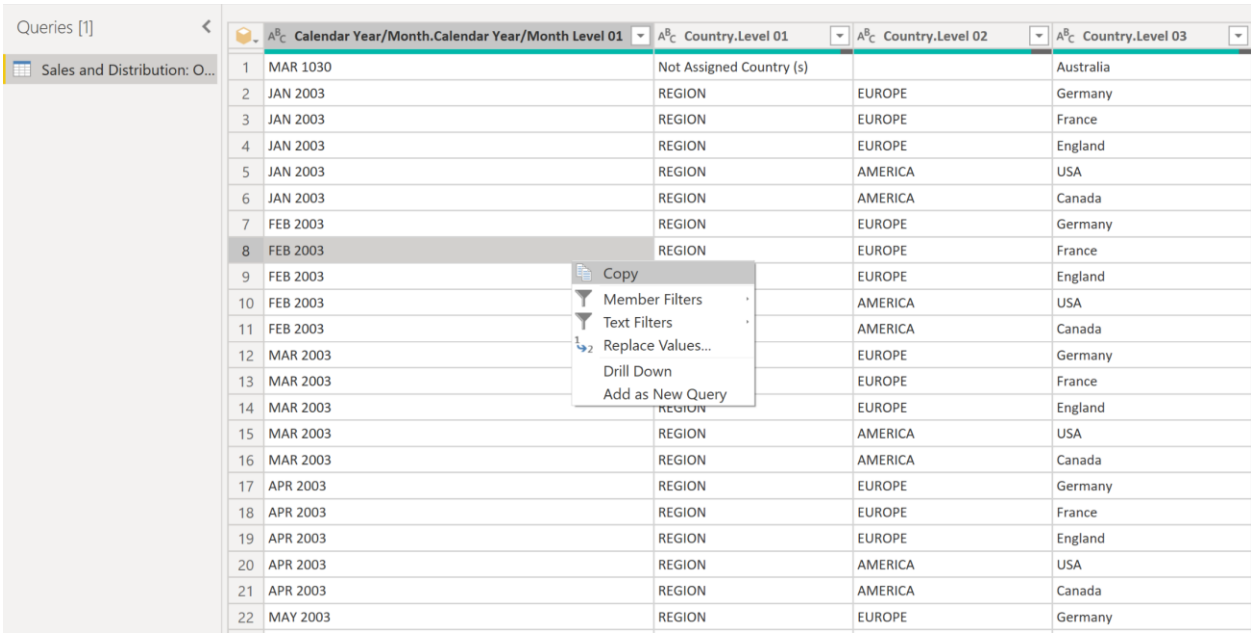
If you click on the gear icon next *Added Items*, you can review the selected data objects in SAP BW, or modify the specified query parameters, for example. As described earlier, it is possible this way to filter a dataset using a characteristic that is not included in the result set.



You can apply additional filters on the dataset, by selecting the drop-down menu for one of the columns.



Another easy way to set a filter is to right-click on one of the values in the table, then select **Member Filters** or **Text Filters**.

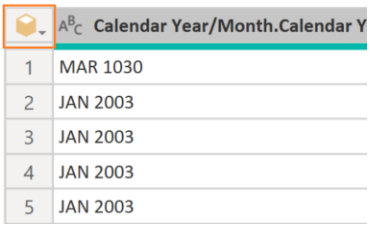


	Calendar Year/Month.Calendar Year/Month Level 01	Country.Level 01	Country.Level 02	Country.Level 03
1	MAR 1030	Not Assigned Country (s)		Australia
2	JAN 2003	REGION	EUROPE	Germany
3	JAN 2003	REGION	EUROPE	France
4	JAN 2003	REGION	EUROPE	England
5	JAN 2003	REGION	AMERICA	USA
6	JAN 2003	REGION	AMERICA	Canada
7	FEB 2003	REGION	EUROPE	Germany
8	FEB 2003	REGION	EUROPE	France
9	FEB 2003		EUROPE	England
10	FEB 2003		AMERICA	USA
11	FEB 2003		AMERICA	Canada
12	MAR 2003		EUROPE	Germany
13	MAR 2003		EUROPE	France
14	MAR 2003		EUROPE	England
15	MAR 2003	REGION	AMERICA	USA
16	MAR 2003	REGION	AMERICA	Canada
17	APR 2003	REGION	EUROPE	Germany
18	APR 2003	REGION	EUROPE	France
19	APR 2003	REGION	EUROPE	England
20	APR 2003	REGION	AMERICA	USA
21	APR 2003	REGION	AMERICA	Canada
22	MAY 2003	REGION	EUROPE	Germany

For example, you could filter the dataset to only include records for Calendar Year/Month FEB 2003, or apply a text filter to only include records where Calendar Year/Month contains 2003.

Please note that not every filter will get folded into the query against SAP BW. You can determine if a filter is folded into the query by examining the icon in the top left corner of the data table, directly above the number *1* of the first data record.

If the icon is a cube, then the filter is applied in the query against the SAP BW system.



	Calendar Year/Month.Calendar Y
1	MAR 1030
2	JAN 2003
3	JAN 2003
4	JAN 2003
5	JAN 2003

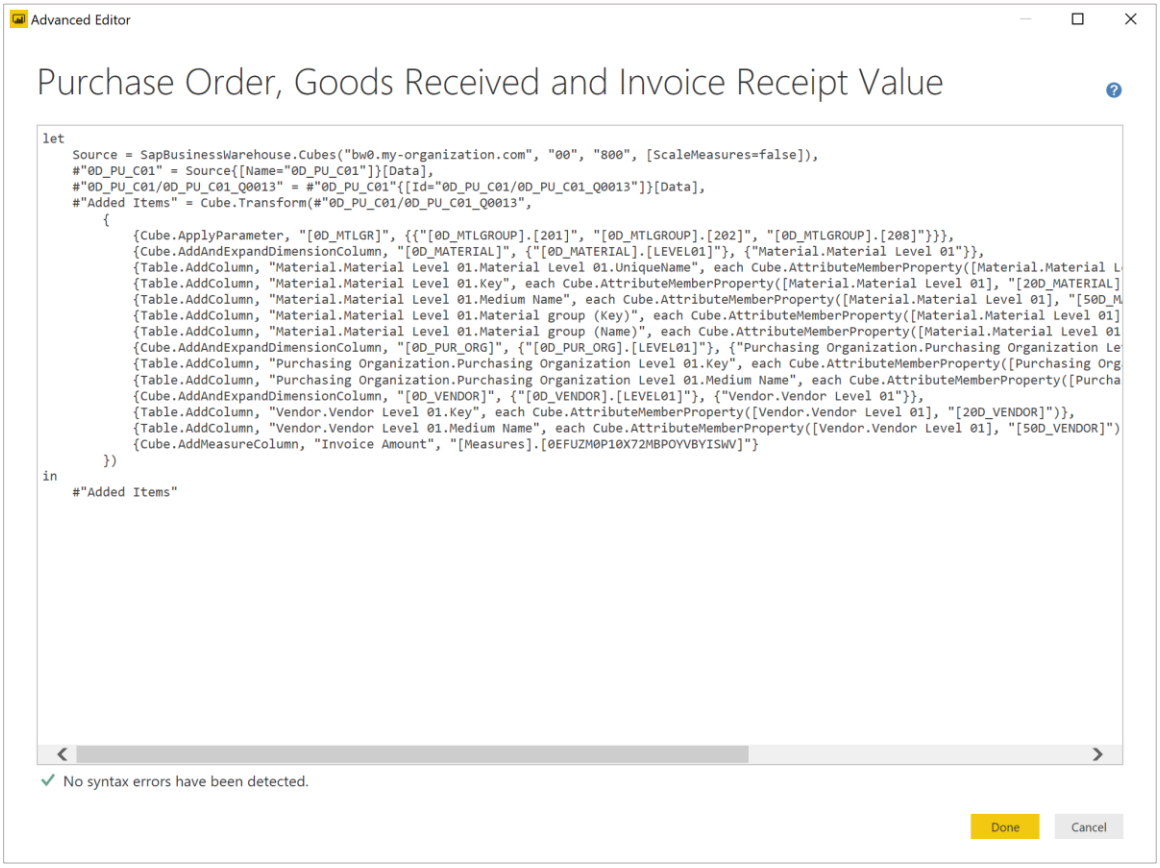
If the icon is a table, then the filter is not part of the query and only applied to the table.

	Calendar Year/Month.Calendar Y
8	APR 2003
9	APR 2003
10	APR 2003
11	APR 2003
12	MAY 2003
13	MAY 2003

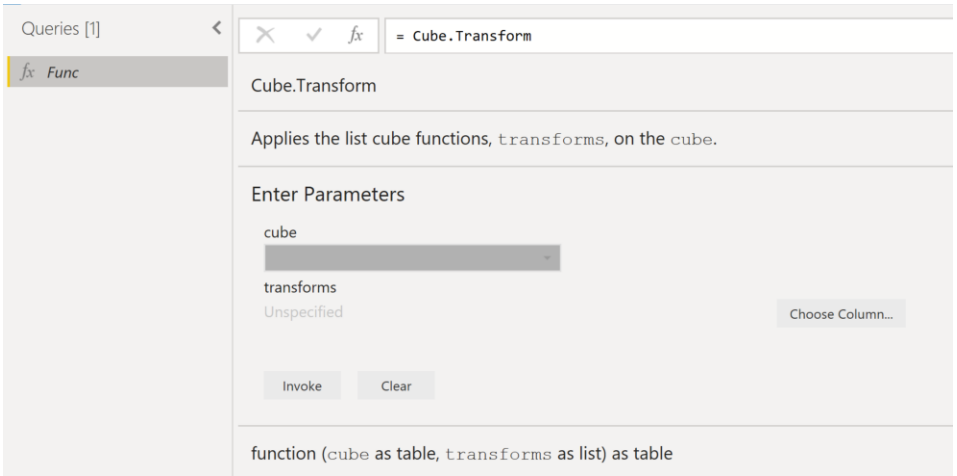
Behind the UI of the Query Editor, Power BI generates code, based on the M formula language for data mashup queries.

For additional information please see the Power Query M Reference.

You can view the generated M code with the *Advanced Editor* option in the *View* menu.



To see a description for each function, or to test it, you can right-click on the existing SAP BW query in the *Queries* panel and select *Create Function*. In the Formula Bar at the top, type = <*function name*>

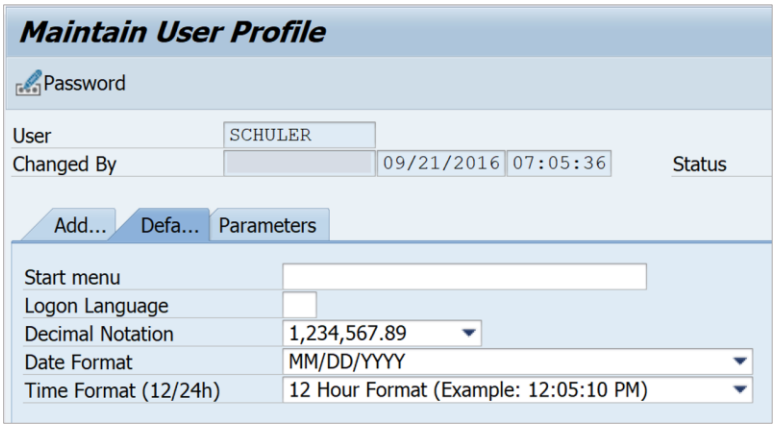


Cube.Transform	Applies the list of functions or transforms on the cube.
Cube.ApplyParameter	Applies the specified values for a parameter.
Cube.DisplayFolders	Returns a nested tree of objects representing the display folder hierarchy of the cube
Cube.Parameters	Returns a table with the set of parameters that can be applied to the cube
Cube.Dimensions	Returns a table with the set of dimensions for the cube
Cube.Measures	Returns a table with the set of measures for the cube

Decimal Notation

User accounts in SAP BW have default settings for how decimal or date/time values are formatted when displayed to the user in SAP GUI.

The default settings are maintained in the SAP system in the **User Profile** for an account, and the user can view or change these settings in SAP GUI with the menu path **System -> User Profile -> Own Data**.



Power BI Desktop queries the SAP system for the decimal notation of the connected user and it uses that notation to format decimal values in the data from SAP BW.

To determine the decimal notation for the connected user, Power BI calls the function module BAPI_USER_GET_DETAIL in SAP BW. This function module returns a structure named DEFAULTS, with a field named DCPFM that stores the decimal format notation.

It is important that the user account in SAP BW has the necessary permissions to execute the function module BAPI_USER_GET_DETAIL, otherwise the call from Power BI Desktop will fail, typically with the error message "You are not authorized to display users in group …"

Import vs. Direct Query

With Power BI, you can connect to a wide variety of data sources, including online services, databases, different file formats, and others. You can connect to these data sources in two different ways: you can either **import** data into Power BI, or you can connect directly to data in the source repository, which is known as **DirectQuery**. When you connect to an SAP BW system, you can also choose between the two connectivity modes. For a complete list of sources that support DirectQuery, refer to the article Data Sources supported by DirectQuery.

The main differences between the two connectivity modes are outlined here, as well as guidelines and limitations, as they relate to SAP BW connections. For additional information please also take a look at Using DirectQuery in Power BI.

Import Connections

When you connect to a data source with Power BI Desktop, the Navigator will allow you to select a set of tables (for relational sources) or a set of source objects (for multidimensional sources).

For SAP BW connections, you can select the objects you want to include in your query from the tree displayed. You can select an InfoProvider or BEx query for an InfoProvider, expand its key figures and dimensions and select specific key figures, characteristics, attributes (properties) or hierarchies to be included in your query.

The selection defines a query that will return a flattened data set consisting of columns and rows. The selected characteristics levels, properties and key figures will be represented in the data set as columns. The key figures are aggregated according to the selected characteristics and their levels. A preview of the data is displayed in the Navigator. You can edit these queries in Power BI Desktop prior to loading the data, for example to apply filters, or aggregate the data, or join different tables.

When the data defined by the queries is loaded, it will be imported into the Power BI in-memory cache.

As you start creating your visuals in Power BI Desktop, the imported data in the cache will be queried. The querying of cached data is very fast and changes to the visuals will be reflected immediately.

However, the user should take care when building visuals that further aggregate the data, when dealing with non-additive measures. For example, if the query imported each *Sales Office*, and the *Growth %* for each one, then if the user built a visual that will *Sum* the *Growth %* values across all *Sales Offices*, that aggregation will be performed locally, over the cached data. The result would not be the same as requesting

the overall *Growth %* from SAP BW, and is probably not what is intended. To avoid such accidental aggregations, it is useful to set the **Default Summarization** for such columns to **Do not summarize**.

If the data in the underlying source changes, it will not be reflected in your visuals. It will be necessary to do a **Refresh**, which will re-import the data from the underlying source into the Power BI cache.

When you publish a report (.pbix file) to the Power BI service, a dataset is created and uploaded to the Power BI server. The imported data in the cache is included with that dataset. While you work with a report in the Power BI service, the uploaded data is queried, providing a fast response time and interactivity. You can set up a scheduled refresh of the dataset, or re-import the data manually. For on-premise SAP BW data sources, it is necessary to configure an on-premises data gateway (further described in the following chapter).

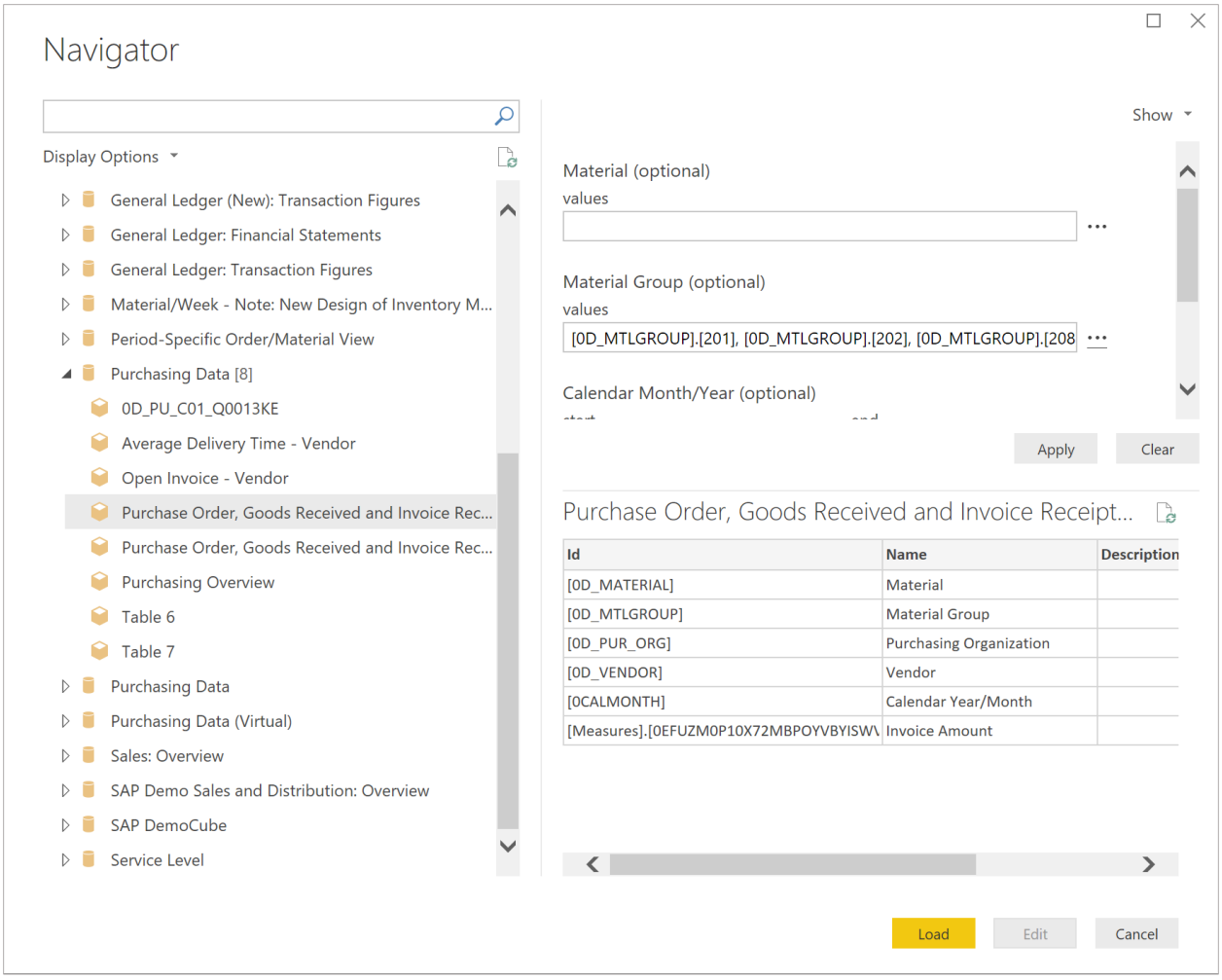
DirectQuery Connections

The navigation experience is slightly different when connecting to an SAP BW source in DirectQuery mode. The Navigator will still display a list of available InfoProviders and BEx queries in SAP BW, however no Power BI query is defined in the process. You will simply select the source object itself, i.e. the InfoProvider or BEx query, and see the field list with the characteristics and key figures once you connect.

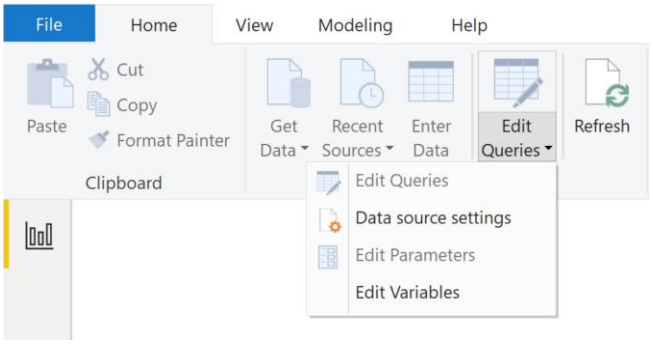
For BW queries with variables, you will be able to enter or select values as parameters of the query. Select the **Apply** button, to include the specified parameters in the query.

Instead of a data preview, the metadata of the selected InfoCube or BEx Query is displayed.

Once you click on the **Load** button in the Navigator, no data will be imported.



You can make changes to the values for the BW query variables with the **Edit Queries** command on the Power BI Desktop ribbon.



As you start creating your visuals in Power BI Desktop, the underlying data source in SAP BW will be queried to retrieve the required data. The time it takes to update a visual will depend on the performance of the underlying SAP BW system.

Any changes in the underlying data will not be immediately reflected in your visuals. It will still be necessary to do a **Refresh**, which will re-run the queries for each visual against the underlying data source.

When you publish a report to the Power BI service, it will again result in the creation of a dataset in the Power BI service, just as for an import connection. However, no data is included with that dataset.

While you work with a report in the Power BI service, the underlying data source is queried again to retrieve the necessary data. For DirectQuery connections to your SAP BW and SAP HANA systems, it is required to have an on-premises data gateway installed and the data source registered with the gateway (further described in the following chapter).

For BW queries with variables, end users will have the ability to edit parameters of the query.
Note: at the time of writing this document this feature is in preview mode. For the end user to edit parameters, the dataset needs to be published to a premium workspace, in Direct query mode and SSO needs to be enabled.

General Recommendations

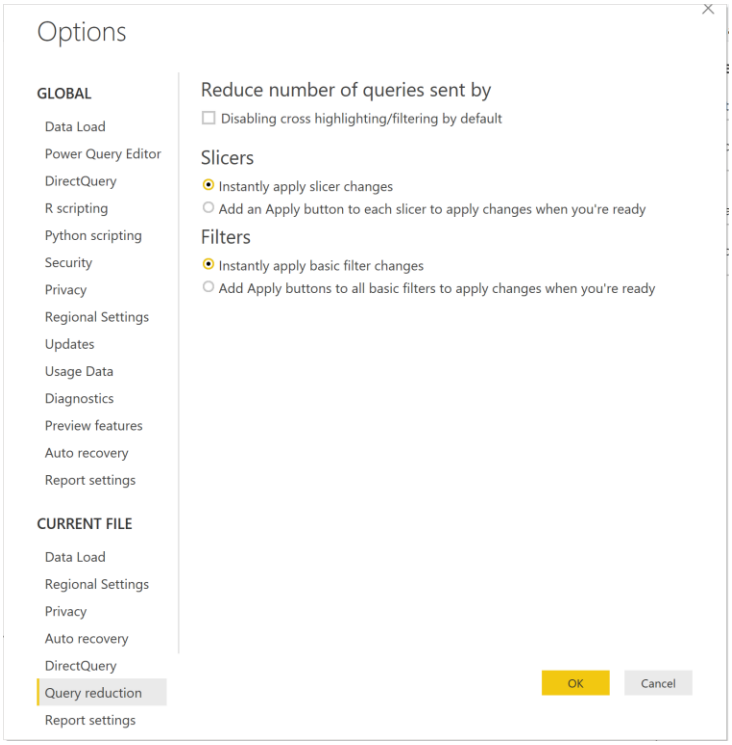
You should **import** data to Power BI whenever possible. This takes advantage of the high-performance query engine of Power BI and provides a highly interactive and fully featured experience over your data.

However, DirectQuery provides the following advantages when connecting to SAP BW:

- Provides the ability to access SAP BW data using SSO, to ensure that security defined in the underlying SAP BW source is always applied. When accessing SAP BW using SSO, the user's data access permissions in SAP will apply, which may produce different results for different users. Data that a user is not authorized to view will be trimmed by SAP BW.
- Ensures that the latest data can easily be seen, even if it is changing frequently in the underlying SAP BW source.
- Ensures that complex measures can easily be handled, where the source SAP BW is always queried for the aggregate data, with no risk of unintended and misleading aggregates over imported caches of the data.
- Avoids caches of data being extracted and published, that might violate data sovereignty or security policies that apply.

Using DirectQuery is generally only feasible when the underlying data source can provide interactive queries for the typical aggregate query within seconds and is able to handle the query load that will be generated. Additionally, the list of limitations that accompany use of DirectQuery should be considered, to ensure your goals can still be met.

If you are working with either very large datasets or encounter slow SAP BW query response time in DirectQuery mode, Power BI provides options in the report to send fewer queries, which makes it easier to interact with the report. To access these options in Power BI Desktop, go to **File > Options and settings > Options**, and select **Query reduction**.



You can disable cross-highlighting throughout your entire report which will reduce the number of queries sent to SAP BW. You can also add an **Apply** button to slicers and filter selections. You can make as many slicer and filter selections as you want, but no queries will be sent to SAP BW until you select the Apply button. Your selections will then be used to filter all your data.

These changes will apply to your report while you interact with it in Power BI Desktop as well as when your users consume the report in the Power BI service.

In the Power BI service, the query cache for Direct Query connections is updated on a periodic basis by querying the data source. By default, this happens every hour, but it can be configured to a different interval in dataset settings. This is further described in the On-Premises Data Gateway chapter of this document.

In addition, many of the general best practices described in Using DirectQuery in Power BI apply equally when using DirectQuery over SAP BW. Additional details specific to SAP BW are described in https://docs.microsoft.com/en-us/power-bi/desktop-directquery-sap-bw

Data Refresh with the On-Premises Gateway

Once you have your visuals and reports published to the Power BI service, you can keep your data fresh by connecting to your BW data sources with the on-premises data gateway. This is an important consideration for SAP customers with on-premise installations of BW.

The on-premises data gateway from Microsoft acts as a bridge, providing quick and secure data transfer between the on-premises SAP BW system(s) and Power BI. The on-premises data gateway also works with Microsoft Flow, Logic Apps, and PowerApps services.

The data gateway supports both **DirectQuery** connections to your SAP BW systems, as well as **Import** connections, which can be refreshed manually or scheduled.

For general information about the on-premises data gateway, please see https://docs.microsoft.com/en-us/power-bi/service-gateway-onprem

Installation of the Gateway

The on-premises data gateway supports two installation modes:

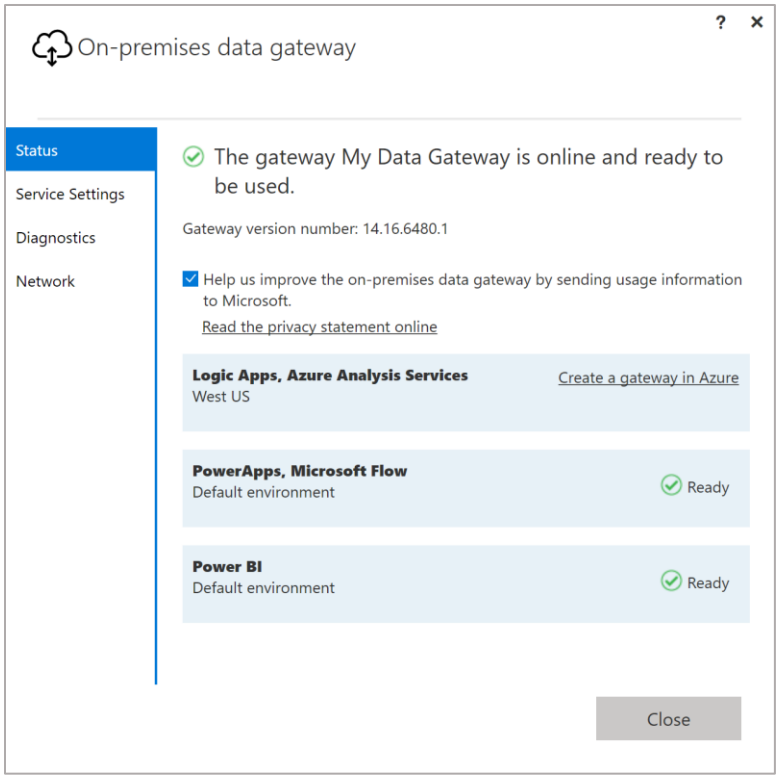
On-premises data gateway – allows multiple users to connect to multiple on-premises data sources, and can be used by Power BI, PowerApps, Flow and Azure Logic apps, all with a single gateway installation. Both DirectQuery and scheduled refresh are supported.
This is the recommended mode for enterprise application sources such as SAP BW.

On-premises data gateway (personal mode) – allows one user to connect to sources, and can't be shared with others. Can only be used with Power BI. Only scheduled refresh is supported.

For detailed requirements and installation steps for the on-premises data gateway, please go see the download page.

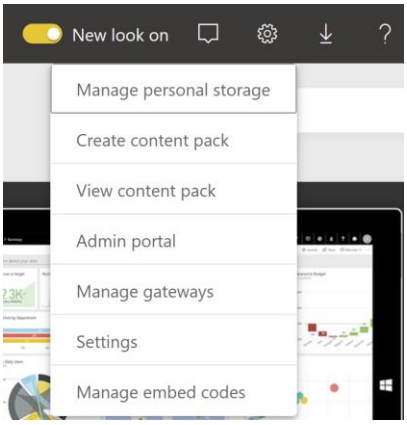
On-premises Data Gateway App

The on-premises data gateway will install an app on your on-premise machine which you can launch to find status information, restart the gateway, access and set the level for diagnostics information and configure secure communication.

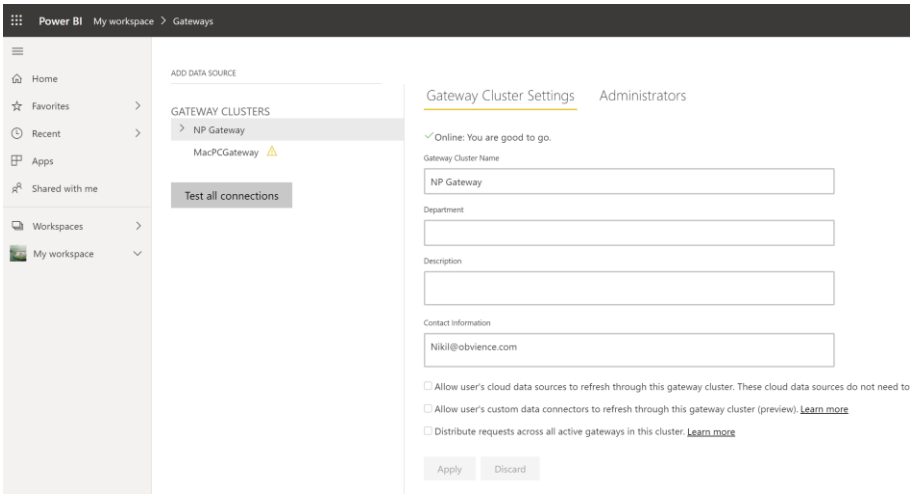


Managing your SAP BW Data Source in Power BI

Once you have installed the on-premises data gateway, you can manage it in the Power BI service from the **Settings -> Manage gateways** menu.



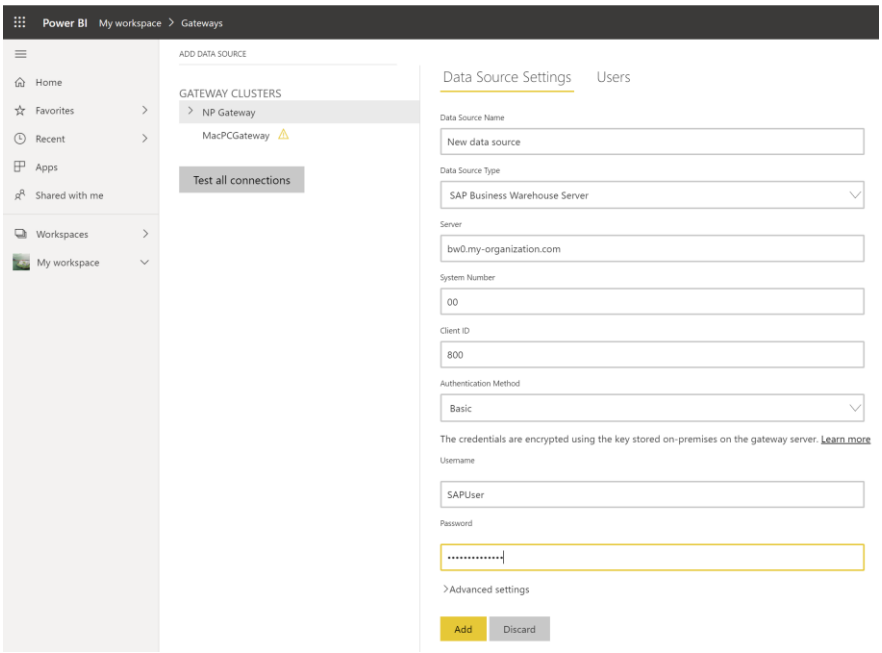
In the Gateway Settings, you can provide basic information about your gateway, including **Name**, **Description** or **Contact Information**.



On the **Administrators** tab for the gateway, you can add, or remove, users or security groups that can administer the gateway.

To add your SAP BW data source(s) that you want to access with the gateway, select **Add Data Source**.

Specify a **Source Name**, select the **Source Type** as **SAP Business Warehouse Server** and enter the connection parameters for your server.

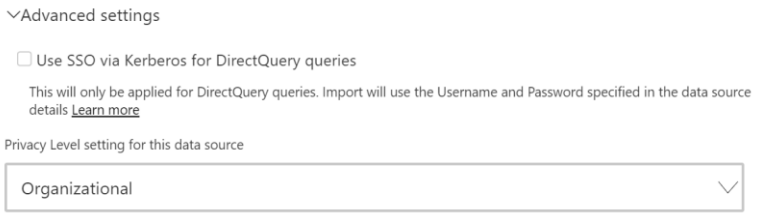


Note that the **Username** and **Password** represent an SAP logon account which has been configured with certain privileges in SAP BW. The data set being retrieved from SAP BW will depend on the privileges for this account in SAP BW.

Under **Advanced settings**, you can configure the privacy level for your data source. This controls how data can be mashed up. This setting is only used for scheduled refresh, it does not apply to DirectQuery.

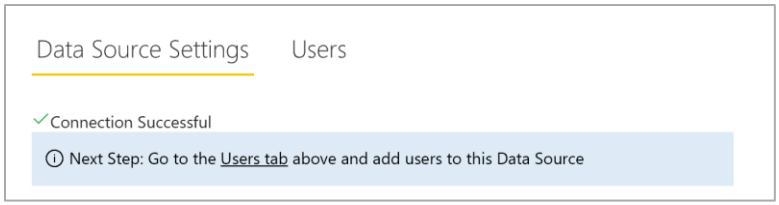
You can also enable SSO via Kerberos for Direct Queries under Advanced settings.

For additional information on setting privacy levels in Power BI Desktop, please see https://docs.microsoft.com/en-us/power-bi/desktop-privacy-levels



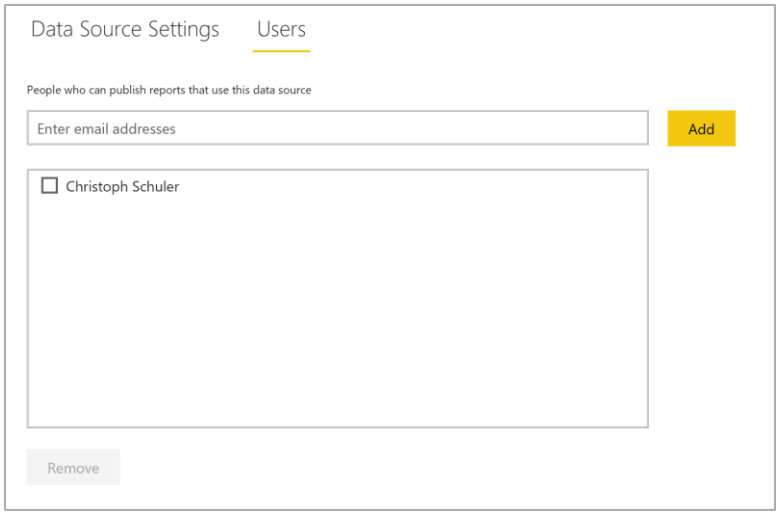
After you **Add** the data source settings, you can use this data source for scheduled refresh or DirectQuery against your on-premise SAP BW system.

You will see a confirmation message if your connection was successful.



On the **Users** tab for your SAP BW data source, you can add, or remove, users, or security groups, that can use this data source.

Note that the users list only controls who is allowed to publish reports that use the BW data source. The report owners can create dashboards, or content packs, and share those with other users.



Using your SAP BW Data Source in Power BI

After you have created the SAP BW data source, it will be available to use with both DirectQuery and Import (manual or scheduled refresh) connections.

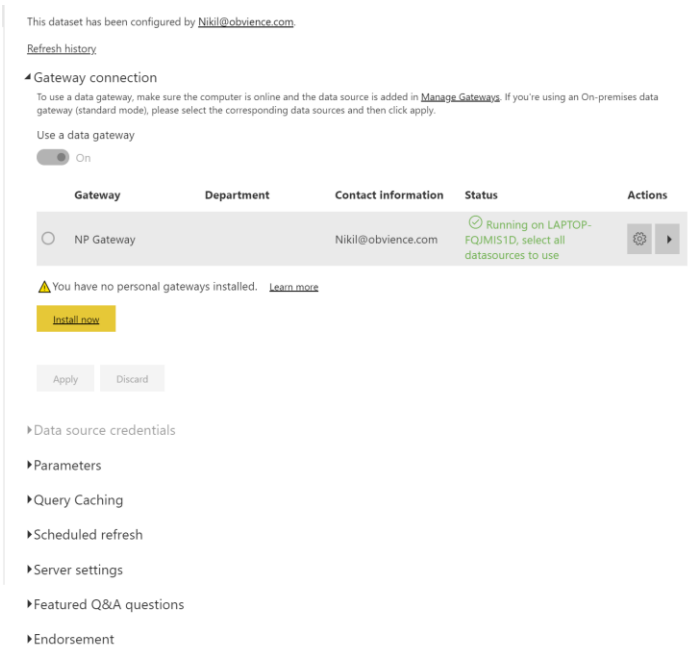
Note that the SAP BW **server**, **instance** number, and **client** number have to match between the connection in Power BI Desktop and the data source connection in the on-premises data gateway.

The link between your dataset and the data source in the gateway is based on these three attributes of the SAP BW connection.

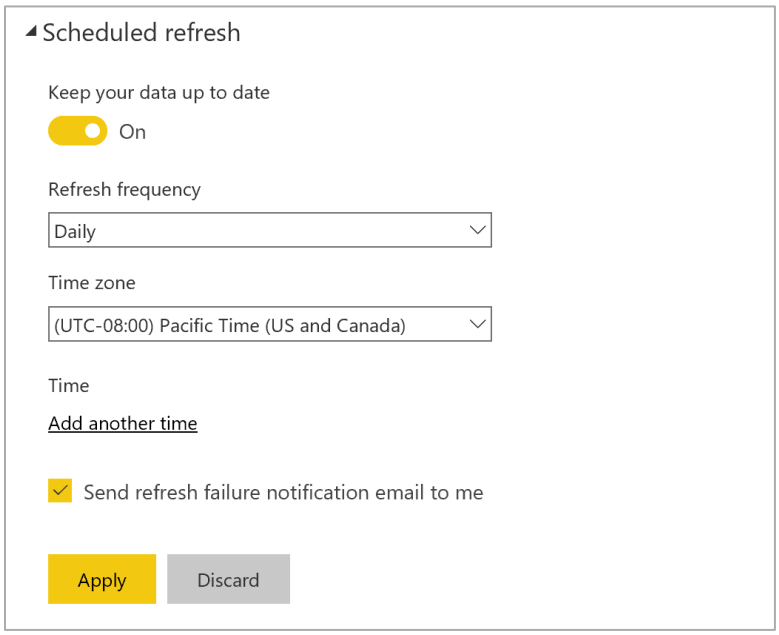
Import Connections: Scheduled Refresh

If you are listed in the **Users** tab of the data source configured in the gateway, and the SAP BW server name, instance and client number match, you will see the **Gateway connection** and **Scheduled refresh** as an option for your dataset.

Go to your Workspace **Settings** and switch to the **Datasets** tab. Select the dataset that you want to schedule a refresh for.



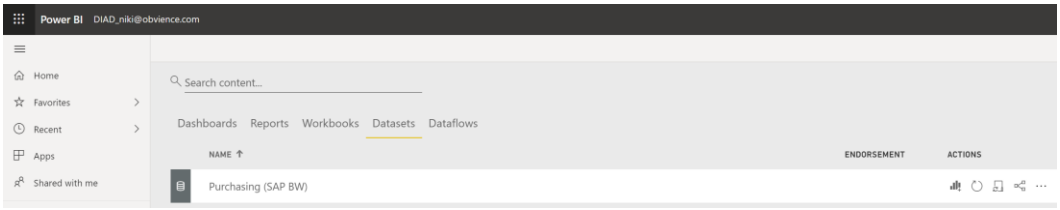
You can select the data gateway to be used to connect to your on-premise SAP BW system, and specify a refresh frequency for the dataset.



Additional information about configuring scheduled refresh is available at https://docs.microsoft.com/en-us/power-bi/refresh-scheduled-refresh

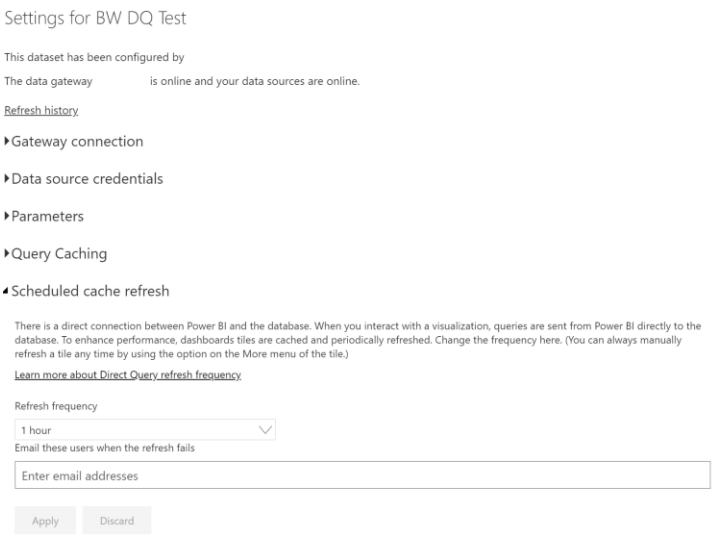
Import Connections: Manual Refresh

If you are listed in the **Users** tab of the data source configured in the gateway, and the SAP BW server name, instance and client number match, you will be able to manually refresh the BW dataset. Go to your Power BI workspace, select the **Datasets** tab and choose **Refresh** from the **Actions** menu.



Direct Query Connections

When you interact with a visualization or report with a Direct Query connection to SAP BW, the underlying source will be queried directly. To enhance performance, Power BI caches dashboard tiles and refreshes them periodically. You can change the default frequency for the cache refresh in the settings for your Direct Query dataset.



Please also see https://powerbi.microsoft.com/en-us/blog/announcing-custom-cache-refresh-schedules-in-the-power-bi-service/ for more information on custom cache refresh schedules in the Power BI service.

Support for SAP BW Features

The following table lists all SAP BW features that are not fully supported or will behave differently when using Power BI.

Feature	Description
Local calculations	Local calculations defined in a BEX Query will change the numbers as displayed through tools like Bex Analyzer. However, they are not reflected in the numbers returned from SAP, through the public MDX interface. **As such, the numbers seen in a Power BI visual will not necessarily match those for a corresponding visual in an SAP tool.** For example, when connecting to a query cube from a BEx query that sets the aggregation to be Cumulated (i.e. 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 is not done.
Aggregations	In some cases (particularly when dealing with multiple currencies), the aggregate numbers returned by the SAP public interface do not match those shown by SAP tools. **As such, the numbers seen in a Power BI visual will not necessarily match those for a corresponding visual in an SAP tool.** 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.
Currency formatting	Any currency formatting (for example, $2,300 or 4000 AUD) is not reflected in Power BI.
Units of measure	Units of measure (for example, 230 KG) are not reflected in Power BI.

Feature	Description
Key versus text (short, medium, long)	For an SAP BW characteristic like CostCenter, the Navigator will show a single item Cost Center Level 01. Selecting this item will include the default text for Cost Center in the field list. In addition, the Key value, Short Name, Medium Name, and Long Name values are available for selection in the Properties node for the characteristic (if maintained in SAP BW). Note that this only applies to Import connectivity mode. For DirectQuery mode, only the default text will be included in the data set.
Attributes	The attributes of a characteristic will be available for selection in the Properties for the characteristic. This only applies to Import connectivity mode. For DirectQuery mode, attributes will not be available.
Multiple hierarchies of a characteristic	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 will result in empty data being returned by SAP.

Treatment of ragged hierarchies	SAP BW supports ragged hierarchies, where levels can be missed, e.g.
	
	Continent
	  Americas
	  Canada
	  USA
	Not Assigned
	  Australia
	
	In Power BI, this appears with (Blank) at the missing level:
	
	Continent
	  Americas
	  Canada
	  USA
	Not Assigned
	  (Blank)

Feature	Description
	Australia
Scaling factor/reverse sign	In SAP, a key figure can have a scaling factor (for example, 1000) defined as a formatting option, meaning that all display will be 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) will result in the unscaled number being used (and the sign is not reversed). The underlying scaling factor is not available. In Power BI visuals, the scale units shown on the axis (K,M,B) can be controlled as part of the visual formatting.
Hierarchies where levels appear/disappear dynamically	Initially when connecting to SAP BW, the information on the levels of a hierarchy will be retrieved, resulting in a set of fields in the field list. This is cached, and if the set of levels changes, then the set of fields do not change until Refresh is invoked. This is only possible in Power BI Desktop. Such Refresh to reflect changes to the levels cannot be invoked in the Power BI service after Publish.
Default filter	A BEX query can include Default Filters, which will be applied automatically by SAP Bex Analyzer. These are not exposed, and hence the equivalent usage in Power BI will not apply the same filters by default.
Hidden Key figures	A BEX query can control visibility of Key Figures, and those that are hidden will not appear in SAP BEx Analyzer. This is not reflected through the public API, and hence such hidden key figures will still appear in the field list. However, they can then be hidden within Power BI.
Numeric formatting	Any numeric formatting (number of decimal positions, decimal point etc.) will not automatically be reflected in Power BI. However, it is possible to then control such formatting within Power BI.
Hierarchy versioning	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 will be available in Power BI, as information on versions is not exposed by the public API.

Feature	Description
Time dependent hierarchies	When using Power BI, time dependent hierarchies are evaluated at the current date.
Currency conversion	SAP BW supports currency conversion, based on rates held in the cube. Such capabilities are not exposed by the public API, and are therefore not available in Power BI.
Sort Order	The sort order (by Text, or by Key) for a characteristic can be defined in SAP. This sort order is not reflected in Power BI. For example, months might appear as “April”, “Aug”, and so on. It is not possible to change this sort order in Power BI.
Technical names	In the Navigator, the characteristic/measure names (descriptions) and technical names can both be displayed using the Display Options selector. The field list contains the characteristic/measure names (descriptions).
End user language setting	The locale used to connect to SAP BW is set as part of the connection details, and does not reflect the locale of the final report consumer.
Text Variables	SAP BW allows field names to contain placeholders for variables (for example, "$YEAR$ Actuals") that would then get replaced by the selected value. For example, the field appears as "2016 Actuals" in BEx tools, if the year 2016 were selected for the variable. The column name in Power BI will not be changed depending on the variable value, and therefore would appear as "$YEAR$ Actuals". However, the column name can then be changed in Power BI.
Customer Exit Variables	Customer Exit variables are not exposed by the public API, and are therefore not supported by Power BI.

Performance Considerations

The following table provides a summery list of suggestions to improve performance for data load and refresh from SAP BW.

Suggestion	Description
Limit characteristics and properties (attribute) selection	The time it takes to load data from SAP BW into Power BI increases with the size of the dataset, i.e. the number of columns and rows in the flattened result set. To reduce the number of columns, only select the characteristics and properties in the Navigator that you eventually want to see in your report or dashboard.
Make use of parameters	Using filters / parameters contributes to reducing the size of the result set, thereby significantly improving query runtimes. Parameters are especially valuable when used with large dimensions, where there is many members such as customers, materials or document numbers.
Limit number of key figures	Selecting many key figures from a BEx query/BW model can have a significant performance impact during query execution, due to the time being spent on loading metadata for units. Only include the key figures that you need in Power BI.
Split up very large queries into multiple, smaller queries	For very large queries against InfoCubes or BEx queries it may be beneficial to split up the query. For example, one query might be getting the key figures, while another query (or several other queries) is getting the characteristics data. You can join the individual query results in the Power BI Desktop model.
Avoid Virtual Providers (MultiProviders or InfoSets)	VirtualProviders are similar to structures without persistent storage. They are useful in many scenarios, but can show slower query performance because they represent an additional layer on top of actual data.

Suggestion	Description
Avoid use of navigation attributes in BEx query	A query with a navigation attribute has to run an additional join, compared with a query with the same object as a characteristic in order to arrive at the values.
Use RSRT to monitor and troubleshoot slow running queries	You SAP Admin can use the Query Monitor in SAP BW (transaction RSRT) to analyze performance issues with SAP BW queries. Please review SAP note 1591837 for more information.
Avoid Restricted Key Figures and Calculated Key Figures	Both are computed during query execution and can slow down query performance.
Consider using incremental refresh to improve performance	Power BI refreshes the complete dataset with each refresh. If you are working with large volume of data, refreshing full dataset on each refresh may not be optimal. In this scenario, you can use incremental refresh, so you are refreshing only a subset of data. Following link has details – https://docs.microsoft.com/en-us/power-bi/service-premium-incremental-refresh

Enabling Tracing to debug

While working with SAP BW connector, you might get MDX errors. This is the error that is bubbled from SAP BW and sometimes it is hard to make sense of these errors. In this scenario, you might want to enable trace and work through the error. Here is how to do this.

- Create an environment variable **PBI_EnableSapBwTracing** and set it to **true**.
- Open Power BI Desktop file and navigate to **File -> Options and Settings -> Options -> Diagnostics**.
- Select **Open crash dump/traces** folder.
- This opens the trace folder.
- Now refresh the report or create a new query.
- Notice trace files are created with the MDX query. Open this file and copy the MDX statement.
- Open SAP GUI use tcode MDXTEST.
- Paste the MDX statement and execute it. You will get more details and work through the issue.

Single Sign-On Configuration

Single Sign-On can be enabled by configuring your SAP BW system with SNC (Secure Network Communication). The primary purpose of SNC is to secure connections between the NetWeaver ABAP application server and external applications, including SAP GUI. SNC provides an interface for external security products which can used to enable Single Sign-On.

For more detailed information please see Secure Network Communication.

The requirements and steps for configuring Single Sign-On (SSO) between an SAP BW system and Power BI are described here.

Requirements

The following requirements must be met to configure SSO between your SAP BW system and Power BI:

- Active Directory is required in your landscape

SAP BW Configuration

Please refer to this SAP blog and follow the steps in the video: https://blogs.sap.com/2017/07/27/sap-single-sign-on-authenticate-with-kerberosspnego/

1. **Run the SNCWIZARD / SPNEGO configuration wizards**
   As of NetWeaver ABAP version 7.31 or higher, use the configuration wizards (transactions **SNCWIZARD** and **SPNEGO**) in your SAP system to configure SSO.

For earlier versions of NetWeaver ABAP, or if you don't have access to the configuration wizards, you can configure SSO manually:

2. **Create a new service user in AD, representing the NetWeaver ABAP system**

3. **Use SETSPN to set the Service Principal Name (SPN)**

   Make sure that your SPN matches your User Principal Name (UPN).
   You should use YourServiceUser@DOMAIN for the UPN, and SAP/YourServiceUser as the SPN.
   Note the case sensitivity. For additional information see:
   https://wiki.scn.sap.com/wiki/display/Security/SNC+Client+Encryption

4. **Install CommonCryptoLib on your SAP BW server**

   If you are on NetWeaver ABAP kernel 7.20 PL88 or higher, it is highly recommended that you use CommonCryptoLib 8.5.

   Please read the following note for lower kernel versions:
   https://launchpad.support.sap.com/#/notes/0001848999

   You can download CommonCryptoLib from the SAP Service Marketplace (search for "CommonCryptoLib" in the *Downloads* section).
   Rename the file to SAPCRYPTOLIB.SAR and unzip the .SAR file in a convenient location using the following SAPCAR command (make sure you are in the same directory as the SAPCAR tool):

   sapcar -xvf SAPCRYPTOLIB.SAR

5. **Specify your SECUDIR directory**
   Your SECUDIR directory is the directory where the CommonCryptoLib license ticket file lives, as well as where all your generated PSE files will go.

   To set a directory as your SECUDIR directory, create a new environment variable called **SECUDIR** and point it to a directory for example *\usr\sap\<SID>\DVEBMGS00\sec*.

6. **Set the profile parameters in your SAP BW instance**

   Connect to your SAP BW instance via SAP GUI and execute transaction **RZ10**.
   Create new profile parameters as follows:
   snc/gssapi_lib = *<path to directory where you extracted CommonCryptoLib>\sapcrypto.dll*
   snc/identity/as = *p:CN=YourServiceUser@DOMAIN*

   Example:
   p:CN=KerberosABC@MYDOMAIN (this is case sensitive!)

7. **Restart the SAP BW instance**

   Use the **SAP Management Console** to restart the BW instance and enable the profile parameter settings.
   More information on profile parameter settings can be found here:
   https://help.sap.com/saphelp_nw70ehp1/helpdata/en/e6/56f466e99a11d1a5b00000e835363f/frameset.htm

8. **Create the Kerberos Keytab**

   Change directory to where you extracted the .SAR file for CommonCryptoLib earlier and run the following command:
   sapgenpse keytab -p SAPSNCSKERB.pse -x <PSE_password> -X <service_user_password> -a <service_user_UPN>

Example:
sapgenpse keytab -p SAPSNCSKERB.pse -x 12345 -X KerberosABCPassword -a KerberosABC@MYDOMAIN

Be sure to use the service user's password for the -X parameter, otherwise you will not be able to authenticate later.

Your new PSE file should show up in the SECUDIR directory as **SAPSNCSKERB.pse**.
More information about generating the Kerberos keytab can be found here:
https://help.sap.com/viewer/bce2e80c873a4003a6d446e1f370696d/3.0/en-US/e49be955aea4461484cebe0be1c8cb01.html

9. **Create the SAP Cryptolib PSE file**

   After running the previous command, it should produce a warning stating that the SAP Cryptolib PSE file is missing.

   To create the file, sign in to your BW instance via SAPGUI and go to transaction **STRUST**. Right-click **SNC SAPCryptolib**, then select **Create**. Use the default settings.

   Your new PSE file should show up in the **SECUDIR** directory as **SAPSNCS.pse**.

   More information about creating the PSE can be found here:
   https://help.sap.com/saphelp_nw70ehp1/helpdata/en/e6/56f466e99a11d1a5b00000e835363f/frameset

10. **Create the SSO credentials**

    Run the following comand:

    sapgenpse seclogin -p <path>\SAPSNCSKERB.pse -x <PIN> -O <system_user>

    Example:

    sapgenpse seclogin -p /usr/sap/abc/dvebmgS00/sec/SAPSNCSKERB.pse -x 12345 -O SAPServiceABC

    The system user is the user specified in Services.msc for your SAP instance. Usually, this is *SAPService<SID>* for Windows and *<SID>adm* for UNIX.

    The credentials created will be stored in a file called **cred_v2**.

    More information on the system user can be found here:
    https://help.sap.com/saphelp_nw70ehp1/helpdata/en/81/be593bd0714b58e10000000a11402f/content.htm

    More information about creating the SSO credentials can be found here:
    https://help.sap.com/viewer/bce2e80c873a4003a6d446e1f370696d/3.0/en-US/e49be955aea4461484cebe0be1c8cb01

11. **Set the profile parameters in your SAP BW instance**

    Connect to your SAP BW instance via SAP GUI and execute transaction **RZ10**.
    Create new profile parameters as follows:
    snc/enable = **1**

12. **Restart the SAP BW instance**

    Use the **SAP Management Console** to restart the BW instance and enable the profile parameter settings.

User Mapping

1. Sign in to your BW instance again via SAPGUI and go to transaction **SU01**.
2. Enter your SAP user (or the user that you want to map for SSO) in the name field and select **Edit**.
3. Select the **SNC** tab and specify the user's SNC name. It should follow this format:

    p:CN=USERPRINCIPALNAME@DOMAIN

    Note: This is case sensitive!

Secure Login Client

SNC communication will not work unless the external security product is installed on both the client (Power BI Desktop) and server. Clients that install Secure Login Client will have CommonCryptoLib installed for them and all of the environment variables will be preconfigured as well.

The installation package is named SAPSetupSCL.EXE and can be obtained from the SAP Service Marketplace (requires SAP customer credentials).

For further information please see Secure Login.

Configuring SAP GUI for SNC Communications

You can verify the successful configuration of SSO with SAPGUI.

1. Open the **SAP Logon** tool on your desktlop and right-click the connection to your BW instance.
2. Select **Properties** and switch to the **Network** tab.
3. Click the checkbox for **Activate Secure Network Communication** and specify the **SNC Name**.

    This should be in the format:

    p:CN=ServicePrincipalName@domain

    Example:

    p:CN=SAP/KerberosABC@mydomain
4. Click **OK** and try connecting. If everything was configured correctly, you should be signed in without being prompted to enter a password.