

Senturus Analytics Connector

User Guide

Cognos to Power BI

Contents

Overview.....	3
Data Connector vs SQL Server Database Connection.....	4
Connecting to Analytics Connector Server from Power BI Desktop	6
Conecting via SQL Server database connector	6
Connect via Data Connector	11
Creating Relationships	15
Cognos Parameterized Tables and Reports.....	18
Working with Parameterized Tables	18
Working with Reports	22
Working with Power BI Parameters	25
Function Syntax.....	27
Publish to powerbi.com.....	29
CONTACT US.....	33

Overview

This guide describes how the Senturus Analytics Connector is used from Power BI after it has been configured.

Please refer to the *Senturus Analytics Connector Installation and Configuration Guide* for instructions on installing and configuring the Analytics Connector.

The Analytics Connector is designed for Power BI and has been tested against different versions, including both Power BI Desktop and Power BI Gateway on Windows starting with the August 2018 release. For optimal stability and performance, it is recommended that you keep your Power BI software fully updated.

Data Connector vs SQL Server Database Connection

Starting from version 5.0, you can connect from Power BI Desktop and Power BI Gateway to Analytics Connector via SQL Server database connection.

If you migrated from version 4.x, you may still use our custom Data Connector to connect, all your old reports and data sources will continue working. And you can still use our custom Data Connector in new reports if you wish.

Comparing to custom Data Connector, using SQL Server database connection gives you following advantages:

1. No need to install Analytics Connector ODBC client.
2. No need to copy custom Data Connector (.mez files).
3. Use Windows authentication from Power BI Desktop to Analytics Connector server to Cognos.¹
4. More Power BI functions are supported, e.g. [query cancellation](#).

But there is a known issue when using SQL Server database connection to connect to Analytics Connector, you cannot filter decimal columns. This issue will be fixed in later releases.

Our custom Data Connector wraps Analytics Connector ODBC client and enables DirectQuery capabilities.

To use custom Data Connector, follow steps below:

1. Install Senturus Analytics Connector ODBC client. The latest ODBC client can be found at [Senturus Analytics Connector - Customer Resources](#).²
2. Copy the Custom Data Connector “SenturusAnalyticsConnector.mez” from Analytics Connector server (under “<install_root>\Power BI” folder) to your Power BI Desktop computer (under “{User}\Documents\Power BI Desktop\Custom Connectors” folder).

If your Senturus Analytics Connector server administrators made some changes using Senturus Data Source Configuration tool or Service Configuration tool, for example mapped a new database to a Cognos package, or moved Analytics Connector server to a new host, a new “SenturusAnalyticsConnector.mez” file will be generated and you need to copy it again.

¹ Both Power BI Desktop and Cognos server must be secured by the same Active Directory service.

² ODBC client is automatically installed when you are installing Senturus Analytics Connector server. So, you don't have to install ODBC client separately on the same computer.

3. Start Power BI Desktop, go to Files -> Options and Settings -> Options, select Security on left panel, and under "Data Extensions" option, check "(Not Recommended) Allow any extensions to load without validation or warning." You need to restart Power BI Desktop.

Connecting to Analytics Connector Server from Power BI Desktop

You don't connect to Cognos directly from Power BI Desktop or Power BI Gateway. Instead you connect to Analytics Connector server and Analytics Connector server connects to Cognos dispatcher on your behalf. For more information, please refer to "*Senturus Analytics Connector Installation Guide*", section Architecture Overview.

As mentioned before, you can connect to Analytics Connector Server using either Power BI built-in SQL Server database connector, or Senturus custom Data Connector.

Connecting via SQL Server database connector

To begin, open Power BI Desktop and select **Get Data -> SQL Server** from the ribbon. "SQL Server database" dialog pops up.

In **Server** input box, Type in hostname of IP of Analytics Connector server.

In **Database (optional)** input box, type in database name given by your Analytics Connector administrator when he/she mapped a Cognos package or data module to it. Please note, Database is mandatory in this case.

Check **DirectQuery** and (optionally) check "**Navigate using full hierarchy**".

Then click on **OK** button.

SQL Server database

Server [ⓘ]
win03.example.com

Database (optional)
GO Sales (query)

Data Connectivity mode [ⓘ]
 Import
 DirectQuery

Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

Include relationship columns
 Navigate using full hierarchy
 Enable SQL Server Failover support

OK Cancel

You can use either **Windows** authentication or **Database** authentication (by providing your Cognos login user name and password).

SQL Server database

Windows

Database

Microsoft account

win03.example.com;GO Sales (query)

User name
cognos

Password

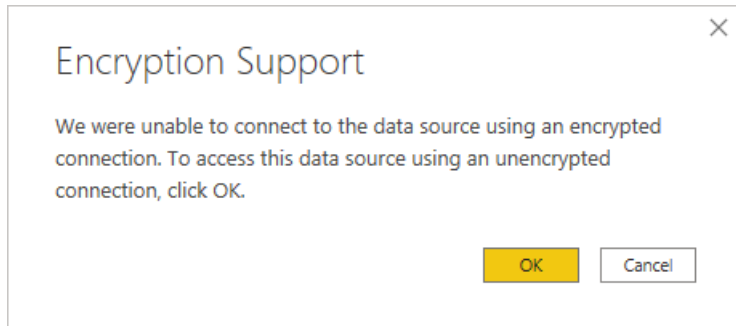
Select which level to apply these settings to
 win03.example.com;GO Sales (query)

Back Connect Cancel

NOTE: Power BI desktop will securely store these credentials and skip this dialog if you have entered before.

To manage or delete the stored authentication information, go to **File → Options and settings → Data source settings**.

You may see a warning popup if SSL is not enabled on Analytics Connector server. Click **OK** to dismiss this message.



If your database is mapped from Cognos package, you will see following objects in Navifator panel:

1. Each namespace is mapped to a schema. Plus some system schemas as you will see when connecting to a real SQL server database.
2. Each query subject is mapped to a table.
3. Each query item will be mapped to a column. Plus a dummy column AAA_LINK in each mapped table. See [Creating Relationships](#) regarding more informaton.
4. If you have standalone calculations under a namespace, each calculation is mapped to a column under a special table **AAA_CALCULATIONS** (table name is configurable in Data Source Configuration tool). You can use this special table same as other query subject mapped tables.
5. If you configured schema to report folder mappings, each report will be mapped to a table under a schema. However, you cannot use report mapped tables as regular tables, please refer to [Working with Reports](#) for more information.

If your database is mapped from Cognos data module, you will see following objects in Navifator panel:

1. Only one schema with the same name as your data module. Plus some system schemas as you will see when connecting to a real SQL server database.
2. Each table in data module is mapped to a SQL Server database table.
3. If you configured schema to report folder mappings, each report will be mapped to a table under a schema. However, you cannot use report mapped tables as regular tables, please refer to [Working with Reports](#) for more information.

Select only tables you need to pull columns from and click on **Load** button.

Hint, you don't have to wait table preview to be loaded before clicking on Load button, and you can uncheck **Display Options -> Enable data previews** option to skip previews.

Navigator

Display Options

- Return
- root
- Sales (query) [15]
 - Branch
 - CALCULATIONS
 - Order
 - Order method
 - Parameterized Products
 - Parameterized Products 2
 - Products
 - Retailer type
 - Retailers
 - Sales
 - Sales staff
 - Time
 - Time (close date)
 - Time (ship date)
 - User
- Sales target (query)
- sys

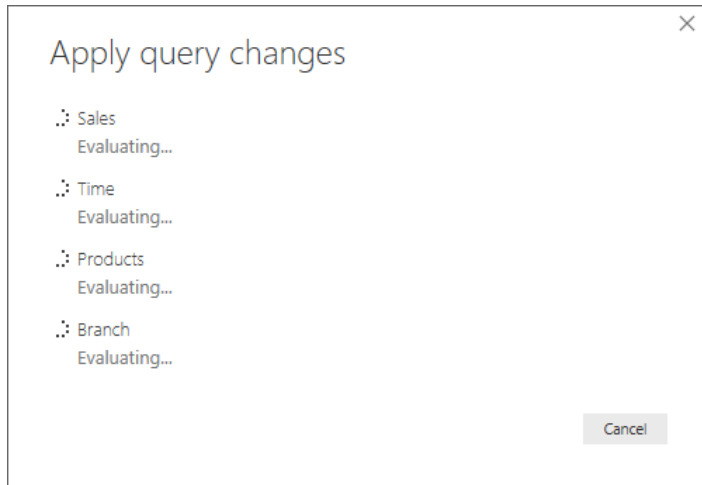
Products

AAA_LINK	Product line code	Product line	Product type code
1	994	Outdoor Protection	
2	991	Camping Equipment	
3	993	Personal Accessories	
4	993	Personal Accessories	
5	993	Personal Accessories	
6	993	Personal Accessories	
7	993	Personal Accessories	
8	994	Outdoor Protection	
9	991	Camping Equipment	
10	991	Camping Equipment	
11	994	Outdoor Protection	
12	994	Outdoor Protection	
13	994	Outdoor Protection	
14	994	Outdoor Protection	
15	994	Outdoor Protection	
16	991	Camping Equipment	
17	991	Camping Equipment	
18	991	Camping Equipment	
19	991	Camping Equipment	
20	995	Golf Equipment	
21	994	Outdoor Protection	
22	995	Golf Equipment	
23	993	Personal Accessories	

Select Related Tables

Load Transform Data Cancel

Power BI will then inspect the selected tables and load them into report.



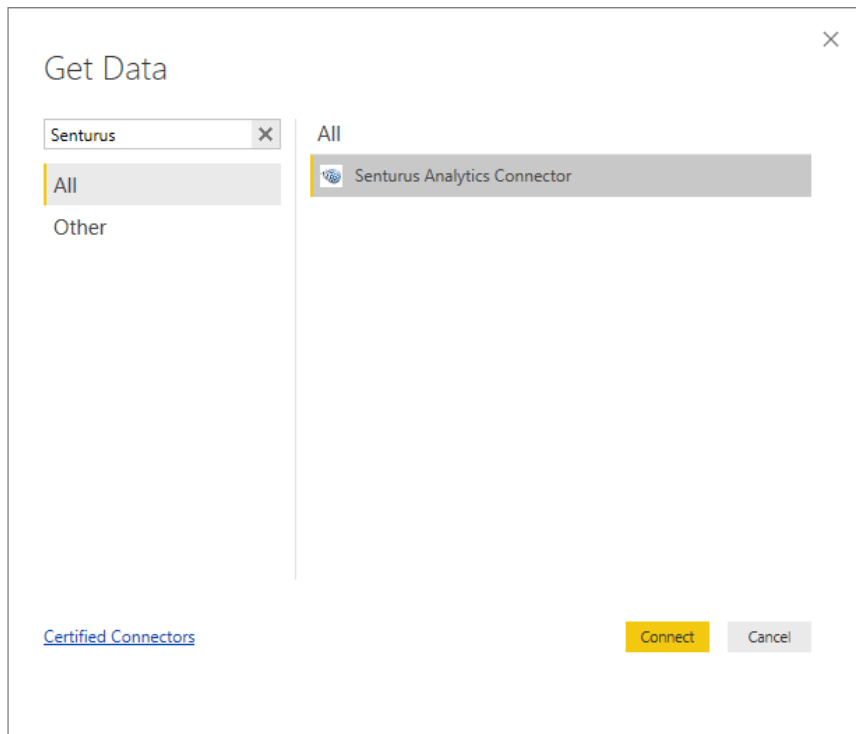
After loading tables into Power BI, you need to create relationships among tables before building report pages. Please refer to *Creating Relationships* for more information.

Connect via Data Connector

To begin, open Power BI Desktop and select **Get Data**.

Find the Senturus Analytics Connector in the list.

Click **Connect**.



Power BI will display a dialog where you can select the Server, Port, Database, and Data Connectivity modes.

The Server, Port and Database lists are pulled from the configuration information in the DataSourceConfig.xml file under the server's Analytics Connector installation folder.

For the **Data Connectivity** mode, select *DirectQuery*.

Click **OK**.

Senturus Analytics Connector

Server
▼

Port
▼

Database
▼

Data Connectivity mode ⓘ
 Import
 DirectQuery

OK Cancel

An authentication dialog will appear. Type in your Cognos credentials, then click **Connect**.

Senturus Analytics Connector

Basic

Senturus Analytics Connector

User name
□

Password
□

Back Connect Cancel

NOTE: Power BI desktop will securely store these credentials and skip this dialog if you select the same Server and Database in a future connection.

To manage or delete the stored authentication information, go to **File → Options and settings → Data source settings**.

Once connected to the Senturus Analytics Connector data source, the **Navigator** window will display.

Select the tables you wish to include.

Click **Load**.

Navigator

Display Options ▾

- Cognos_Senturus: GO Sales (query) [1]
 - GO Sales (query) [11]
 - Connector Reports
 - Filters and calculations
 - Inventory (query)
 - ns1
 - ns2
 - Product forecast (query)
 - Returned items (query)
 - root
 - Sales (query) [14]
 - AAA_CALCULATIONS
 - Branch
 - Order
 - Order method
 - Parameterized Products
 - Parameterized Products 2
 - Products
 - Retailer type
 - Retailers

Branch

Preview downloaded on Thursday, October 4, 2018

_LINK_COLUMN_	Branch code	Address 1	Address 1 (multis
1	6	75, rue du Faubourg St-Honoré	75, rue du Fau
2	7	Piazza Duomo, 1	Piazza Duomo
3	9	Singelgravenplein 4	Singelgravenp
4	13	Schwabentor 35	Schwabentor :
5	14	Leopoldstraße 36	Leopoldstraße
6	15	Isafjordsgatan 30 C	Isafjordsgatan
7	17	7800, 756 - 6th Avenue. S.W.	7800, 756 - 6th
8	18	789 Yonge Street	789 Yonge Str
9	19	1288 Dorchester Avenue	1288 Dorchesi
10	20	299 Yale Avenue	299 Yale Aven
11	21	1288 South Barrington Ave.	1288 South Ba
12	22	10032 NW 186th	10032 NW 186
13	23	6c, rue de l'Église	6c, rue de l'Ég
14	24	Prol. Paseo de la Reforma No. 51	Prol. Paseo de
15	25	202-2-3 Hyakunincho	202-2-3 百人
16	26	543-225 Asahi	543-225 旭
17	28	2315 Queen's Ave	2315 Queen's
18	29	Plaza de la Constitución, s/n	Plaza de la Co
19	30	Avenida Paulista, 333	Avenida Paulis
20	31	Kauppakatu 33	Kauppakatu 3:
21	32	234-12, Kongdeok-Dong	공덕동 234-1:
22	33	10 Claymore Hill	克莱莫山 10 :

Load Edit Cancel

Power BI will then inspect the selected tables for column information.

Apply query changes

- Sales
 - Evaluating...
 - Time
 - Evaluating...
 - Products
 - Evaluating...
 - Branch
 - Evaluating...

Cancel

When complete, navigate to the Relationships view and create relationships among tables.

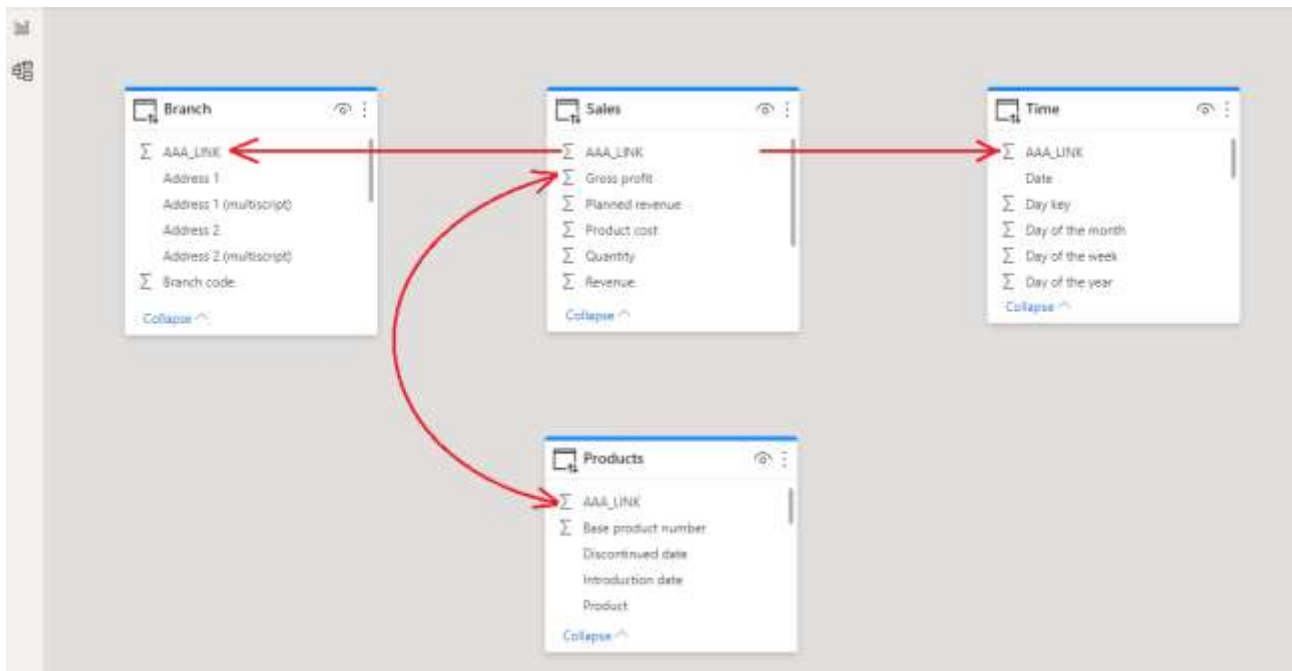
Creating Relationships

Power BI requires relationships between the tables, but Cognos typically does not present primary key and foreign keys at the business user view.

The Analytics Connector injects AAA_LINK columns into the tables so you can build relationships between the tables. The Analytics Connector ignores these columns when passing queries to Cognos to execute.

NOTE: It is very important that you create the relationships as described so that Power BI sends the appropriate queries to Cognos!

Start by dragging the *AAA_LINK* column in the fact table to the *AAA_LINK* table in the dimension table.



In the **Create Relationship** dialog, make sure the cardinality is set *Many to one (*:1)*.

Change the Cross filter direction to *Both*.

Check the box next to *Assume referential integrity*.

Click **OK**.

Hint: You don't have to wait Power BI to populate preview contents before clicking OK button. And if Power BI pops up an error dialog because it does not finish probing two tables yet, you can simply dismiss and ignore that error, as Power BI will create relationships as you specified anyway.

Create relationship

Select tables and columns that are related.

Sales

AAA_LINK	Quantity	Unit cost	Unit price	Unit sale price	Revenue	Product cost	Gross profit	Profit
1	146	40.45	85	85	12410	5905.7	6504.3	
2	8	42.73	89.3	89.3	714.4	341.84	372.56	
3	23	41.36	73	73	1679	951.28	727.72	

Branch

AAA_LINK	Branch code	Address 1	Address 1 (multiscript)	Address 2	Address 2 (multiscript)	City
1	40	55 Rue Rothschild	55 Rue Rothschild	null	null	Genève
2	39	Jedleser Straße 7	Jedleser Straße 7	null	null	Wien
3	38	Interleuvenlaan 2	Interleuvenlaan 2	null	null	Heverlee

Cardinality: Many to one (*:1)

Cross filter direction: Both

Make this relationship active

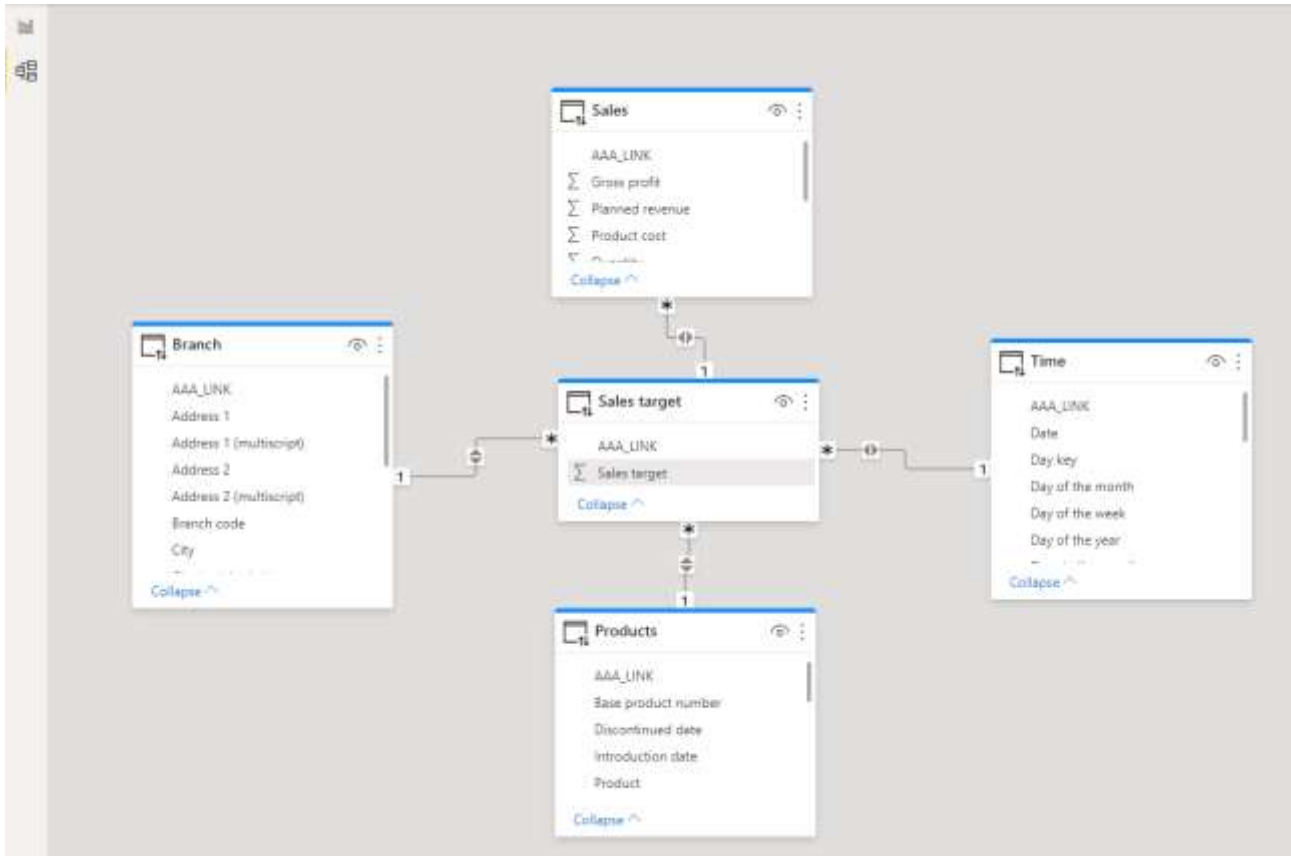
Assume referential integrity [Learn more](#)

Apply security filter in both directions

OK Cancel

Repeat these steps until the fact table has relationships to all dimension tables.

If you have multiple fact tables, chain these fact tables based on granularity and link the highest level fact table to dimension tables, as shown in the diagram below.



After creating relationships, change to the **Report** view.

Use the columns under the **Fields** tab to create your report.

The screenshot shows a report view with a table of sales data. The table has the following data:

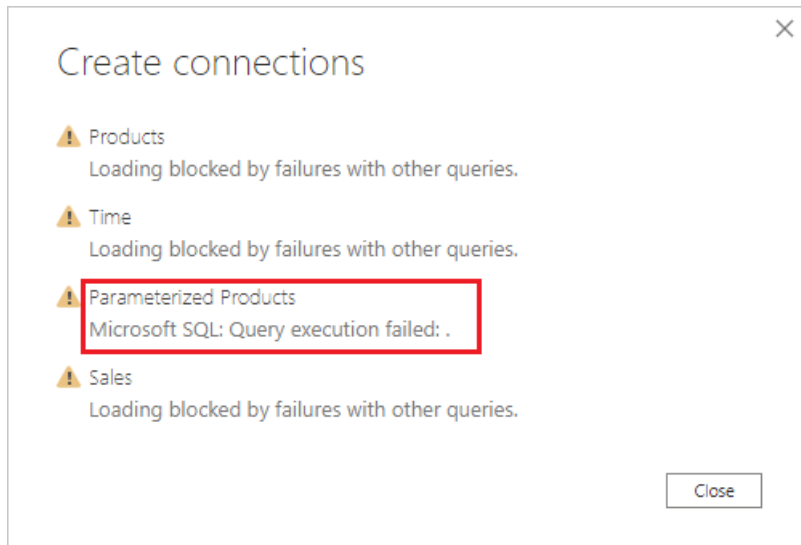
Product line	2010	2011	2012	2013
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	35
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	17
Mountaineering Equipment		107,099,659.94	161,039,823.26	14
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	
Personal Accessories	391,647,093.61	456,323,355.90	594,009,408.42	44
Total	914,352,803.72	1,159,195,590.16	1,495,891,100.90	1,117

The Fields pane on the right shows the following data sources and fields:

- Branch
- Products
- Sales
 - AAA_LINK
 - Gross profit
 - Planned revenue
 - Product cost

Cognos Parameterized Tables and Reports

You cannot use parameterized tables (mapped from Cognos parameterized query subjects) or report mapped tables as regular tables. Power BI will not be able to import them if you select any parameterized tables or report tables:



When using parameterized query subjects in a Cognos report, or executing a pre-defined Cognos report with prompts, users will be prompted to enter parameter values. But you cannot provide parameter values in Power BI for regular database tables.

The answer is to write custom SQL and use **OpenTable** and **RunReport** functions to feed parameter values to Cognos.

Working with Parameterized Tables

The Analytics Connector supports working with parameterized tables in Power BI by using the **OpenTable** function, which lets you pass parameter values to Cognos.

Example:

```
select p."Product line", p."Product type", sum(s.Revenue) as Revenue,
sum(s.Quantity) as Qantity
from
  OpenTable (
    "Sales (query)"."Parameterized Products",
    "p_product line code"="[991, 992, 993, 994, 995]",
    "p_Product Line"="'Golf Equipment'",
```

```
"p_Date"="{d '2010-01-01'}",  
"p_Datetime"="'2010-01-01 19:00:00'"  
) as p  
join "Sales (query)". "Sales" as s on (p.AAA_LINK = s.AAA_LINK)  
group by p."Product line", p."Product type"
```

Instead of selecting from table directly, you use **OpenTable** function (a tabular function) to invoke the parameterized table and join it to other tables. The first parameter "schema name"."table name" is the Cognos table name. This is followed by zero or more Cognos parameter name-value pairs separated by commas. For more information, please refer to *Function Syntax*.

To begin, open Power BI Desktop and select **Get Data -> SQL Server** from the ribbon. "SQL Server database" dialog pops up.

In **Server** input box, Type in hostname of IP of Analytics Connector server.

In **Database (optional)** input box, type in database name given by your Analytics Connector administrator when he/she mapped a Cognos package or data module to it. Please note, Database is mandatory in this case.

Check **DirectQuery** and (optionally) check "**Navigate using full hierarchy**".

Expand "**Advanced options**", put custom SQL in "**SQL Statement (optional, requires database)**" box.

Click **OK**.

SQL Server database

Server [ⓘ]
localhost

Database (optional)
GO Sales (query)

Data Connectivity mode [ⓘ]
 Import
 DirectQuery

Advanced options
Command timeout in minutes (optional)

SQL statement (optional, requires database)
select p."Product line", p."Product type", sum(s.Revenue) as Revenue, sum(s.Quantity) as Qant:
from
OpenTable (
"Sales (query)"."Parameterized Products",
"p_product line code"="[991, 992, 993, 994, 995]",

Include relationship columns
 Navigate using full hierarchy
 Enable SQL Server Failover support

OK Cancel

A data preview window will display. Click **Load**.

localhost: GO Sales (query)

Product line	Product type	Revenue	Quantity
Golf Equipment	Golf Accessories	51514343.88	3119747
Golf Equipment	Irons	254814338	391445
Golf Equipment	Putters	106184271.4	1284570
Golf Equipment	Woods	313898414.7	317939

Load Transform Data Cancel

You can now use the columns from the **Fields** tab to build the report.

The screenshot shows the Power BI Desktop interface. The main area displays a data table with the following content:

Product line	Quantity	Revenue
Golf Equipment	5113701	726,411,367.89
Golf Accessories	3119747	51,514,343.88
Irons	391445	254,814,337.99
Putters	1284570	106,184,271.37
Woods	317939	313,898,414.65
Total	5113701	726,411,367.89

The Fields pane on the right shows a search bar and a list of fields for 'Query1':

- Product line
- Product type
- Σ Quantity
- Σ Revenue

At the bottom of the window, it says 'Page 1 of 1' and 'Storage Mode: DirectQuery (click to change)'.

Working with Reports

The Analytics Connector supports working with Cognos reports in Power BI by using the `RunReport` function, which lets you pass parameter values to a Cognos report.

Analytics Connector only support list style Cognos reports, not crosstabs or charts.

Example:

```
Select *
From
    RunReport (
        "Connector Reports"."Parameter Report",
        "p_product line code"="[991,992,993,994,995]",
        "p_Product Line"="'Golf Equipment'",
        "p_Date"="'2010-01-01'",
        "p_Datetime"="2010-02-01 19:00:00"
    ) as "Report"
```

The first parameter `"schema name"."report name"` is the Cognos report. This is followed by zero or more Cognos parameter name-value pairs separated by commas. For more information of about `RunReport` function, please refer to *Function Syntax*.

To begin, open Power BI Desktop and select **Get Data -> SQL Server** from the ribbon. "SQL Server database" dialog pops up.

In **Server** input box, Type in hostname of IP of Analytics Connector server.

In **Database (optional)** input box, type in database name given by your Analytics Connector administrator when he/she mapped a Cognos package or data module to it. Please note, Database is mendentary in this case.

Check **DirectQuery** and (optionally) check "**Navigate using full hierarchy**".

Expend "**Advanced options**", put custom SQL in "**SQL Statement (optional, requires database)**" box.

Click **OK**.

SQL Server database

Server ⓘ
localhost

Database (optional)
GO Sales (query)

Data Connectivity mode ⓘ
 Import
 DirectQuery

Advanced options
Command timeout in minutes (optional)
[]

SQL statement (optional, requires database)
Select *
From
RunReport (
"Connector Reports"."Parameter Report",
"p_product line code"="[991,992,993,994,995]",
"p_Product Line"="'Golf Equipment'",
...)

Include relationship columns
 Navigate using full hierarchy
 Enable SQL Server Failover support

OK Cancel

A data preview window will display. Click **Load**.

localhost: GO Sales (query)

Product line code	Product line	Product type code	Product type	Year	Revenue
995	Golf Equipment	968	Irons	2010	4654281.1
995	Golf Equipment	969	Woods	2010	5138304.03
995	Golf Equipment	970	Putters	2010	2459044
995	Golf Equipment	971	Golf Accessories	2010	864227.83

Load Transform Data Cancel

You can now use the columns from the **Fields** tab to build the report.

The screenshot shows the Power BI Desktop interface. The ribbon includes File, Home, Insert, Modeling, View, Help, Format, and Data / Drill. The main area displays a PivotTable with the following data:

Product line	2010	Total
Golf Equipment	13,115,856.96	13,115,856.96
Golf Accessories	864,227.83	864,227.83
Irons	4,654,281.10	4,654,281.10
Putters	2,459,044.00	2,459,044.00
Woods	5,138,304.03	5,138,304.03
Total	13,115,856.96	13,115,856.96

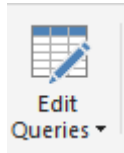
The Fields pane on the right shows a search bar and a list of fields under 'Query1':

- Product line
- Σ Product line code
- Product type
- Σ Product type code
- Σ Revenue
- Σ Year

At the bottom, it says 'Page 1 of 1' and 'Storage Mode: DirectQuery (click to change)'.

Working with Power BI Parameters

In your Power BI ODBC queries, you can replace the static values with Power BI parameters to create dynamic values that can be easily changed.



After configuring a [Parameterized Table](#) or [Report](#) query, click **Edit Queries** to bring up the Power Query Editor.

Click **Manage Parameters** to display the **Parameters** dialog.

Create a new parameter.

The image shows the "Parameters" dialog box in Power BI. On the left, there is a list of parameters with "Start_Date" selected. On the right, the configuration for "Start_Date" is shown: Name: Start_Date, Description: Start date to pass to the parameterized table, Required: checked, Type: Date, Suggested Values: Any value, Current Value: 1/1/2010. At the bottom right are "OK" and "Cancel" buttons.

Parameters

New

Start_Date

Name
Start_Date

Description
Start date to pass to the parameterized table

Required

Type
Date

Suggested Values
Any value

Current Value
1/1/2010

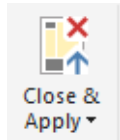
OK Cancel

Next, edit the query to include the parameter in the SQL statement. You may need to format non-text data types using M language functions in order to append them to the statement.

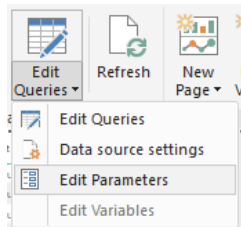
The below example formats a date parameter to text and appends it to the SQL statement.

```
= Odbc.Query("dsn=Senturus Colo .45", "Select * From (#(lf)OpenTable(#(lf) ""Sales (query)""."Parameterized Products"", #(lf) ""p_product line code""="[991, 992, 993, 994, 995]"" , #(lf) ""p_Product Line""=""Golf Equipment'"" , #(lf) ""p_Date""="" & Date.ToText(Start_Date, ""yyyy-MM-dd"" ) & """" , #(lf) ""p_Datetime""=""2010-01-01 19:00:00'""#(lf))#(lf)) as ""Parameterized Table""")]
```

NOTE: Double quotes must be escaped by using two sets of double quotes.



Click **Close & Apply**.



To change the Parameter value(s), select *Edit Parameters* under **Edit Queries**.

The **Enter Parameters** dialog will display with a list of configured parameters and their values.

Fill in new values and click **OK**.

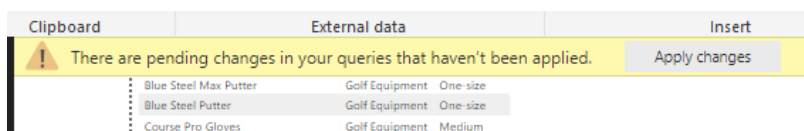
Enter Parameters

Start_Date ⓘ

12/31/2009

OK Cancel

Click **Apply Changes** if the data does not refresh.



Function Syntax

Both OpenTable and RunReport functions adopt same syntax.

Syntax

Both OpenTable and RunReport function adopt following syntax:

```
OpenTable ( "Schema Name"."Table/Report Name"(, "Parameter"="value", ... )
```

The first parameter is a table identifier, using standard SQL syntax. All identifiers are case insensitive and the schema name can be omitted if the table name is unique cross schemas.

Any additional parameters are name-value pairs, separated by commas. Name and values should be surrounded by double quotation marks.

Parameters

Use parameter name in place of "Parameter" for the Cognos connector.

Values

All values should follow SQL/ODBC standard.

Only plain string representation is allowed for numeric values. Formatted representations such as **1,200.00** with thousand separators, **(15)** for -15, or **50%** for .5 are not supported.

For string values, single quotation marks are used around the value. For example, 'Golf Equipment' or 'Sam's Club'.

For date/time/timestamp values, use the ODBC standard format. For example {d '2010-01-01'} or ISO 8601 standard format, 'YYYY-MM-DD' for date, 'hh:mm:ss' for time and 'YYYY-MM-DD hh:mm:ss' for timestamp.

An array of values is passed using brackets. For example:

- "product line code"="[991,992,993,994,995]"

The syntax for a Range parameters is represented by a 2 element array. For example:

- "YearRange"="[,2019]" – All years less than or equal to 2019
- "YearRange"="[2015,]" – All years greater than or equal to 2015
- "YearRange"="[2015,2019]" – Years between 2015 and 2019

Character escaping

Certain characters must be escaped.

For single quotes within a string value, use two single quotes to escape a single quote inside the string.

Double quotes in parameter name and values need to be escaped with two double quotes.

Passing multiple values to a parameter

A Cognos parameter may accept multiple values, for example [Product line code] in (?p_Product line code?). To specify multiple values in your function, put multiple values in a pair of square brackets ([]) and separate each value by a comma (,) just like CSV format.

Publish to powerbi.com

After creating a Power BI report, you can publish it to powerbi.com and share it with others. Simply click on Publish button and select a destination you want to publish.

In order to allow powerbi.com access your on-premises data, you need to install and configure Power BI Gateway. If you use custom Data Connector in your Power BI report, you need to install Senturus Analytics Connector client on the computer where Power BI Gateway is running and copy custom connector file to it. For more information, please refer to Senturus Analytics Connector Installation Guide.

First, you need to create a data source.

If you are using SQL server connections in your report, you can create data sources just like you do for other SQL server data source. Here is an example:

Data Source Settings Users

Data Source Name
GO Sales (query)

Data Source Type
SQL Server ▼

Server
internal-bi1.senturus.com

Database
GO Sales (query)

Authentication Method
Basic ▼

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Username
cognos

Password
.....

Skip Test Connection

> Advanced settings

Add Discard

If you are using custom data connector, you need to select “Senturus Analytics Connector” from the Data Source Type list. Here is an example:

Data Source Settings Users

Data Source Name

GO Sales (query)

Data Source Type

Senturus Analytics Connector

Server

internal-bi1.senturus.com

Port

5432

Database

GO Sales (query)

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Username

cognos

Password

.....

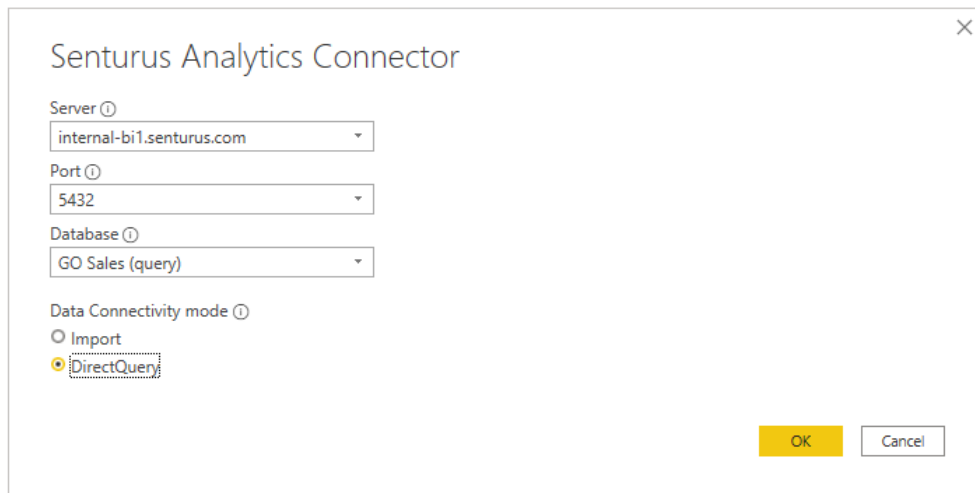
Skip Test Connection

> Advanced settings

Add

Discard

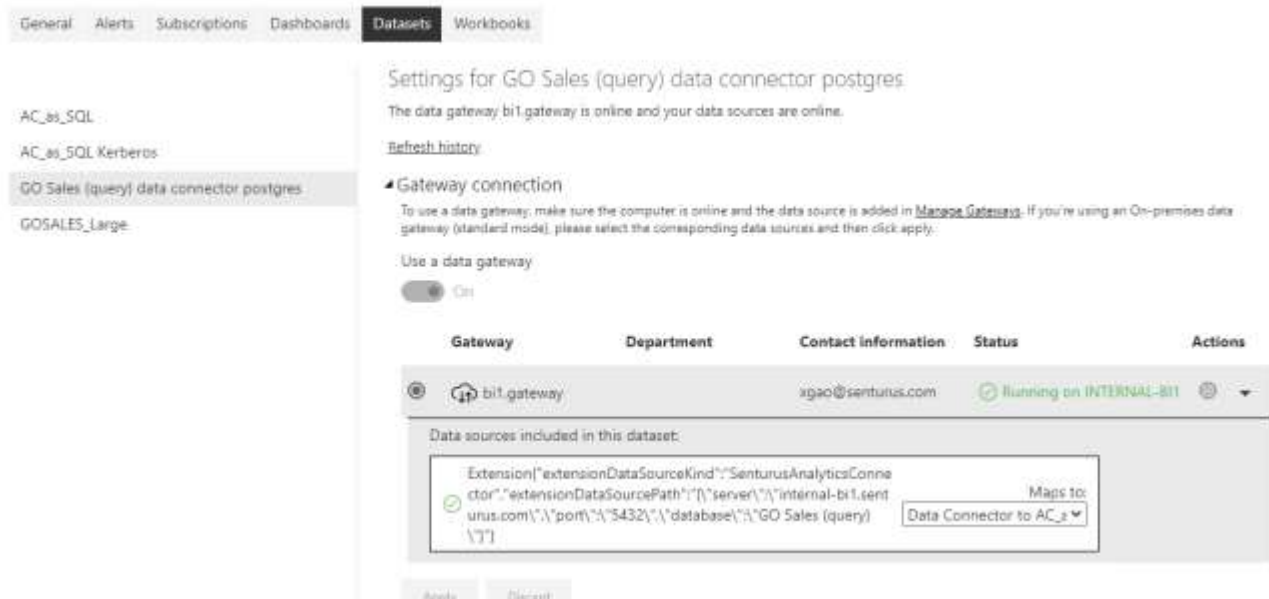
Please note, you need to type in the exact server, port and database here as you have in Power BI Desktop (case sensitive):



Once you set up data sources on powerbi.com, you can publish your reports from Power BI Desktop.

For any new reports, you may see this error when you try to open it on powerbi.com: “There was a data source access error. Please contact the gateway administrator.” This is because the underlying data set is not bound to a data source yet.

Click on “...” right to the associated dataset and select “Settings”, expand “Gateway connection” and map the data source:



Finally, you can open your report and/or create new reports with your dataset.

CONTACT US

The latest version of the Analytics Connector along with related documentation and contact information can be found at <https://www.senturus.net/connector-download/>.

If you have any additional questions, please contact us at CustomerSupport@senturus.com.