

Senturus Analytics Connector

User Guide

Cognos to Power BI

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 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.

Custom Data Connector

The Analytics Connector can be accessed from Power BI by using our Custom Data Connector. This Custom Data Connector enables DirectQuery capabilities, which means queries are passed at run time to the Cognos system for execution under the current user's credentials.

First, 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).

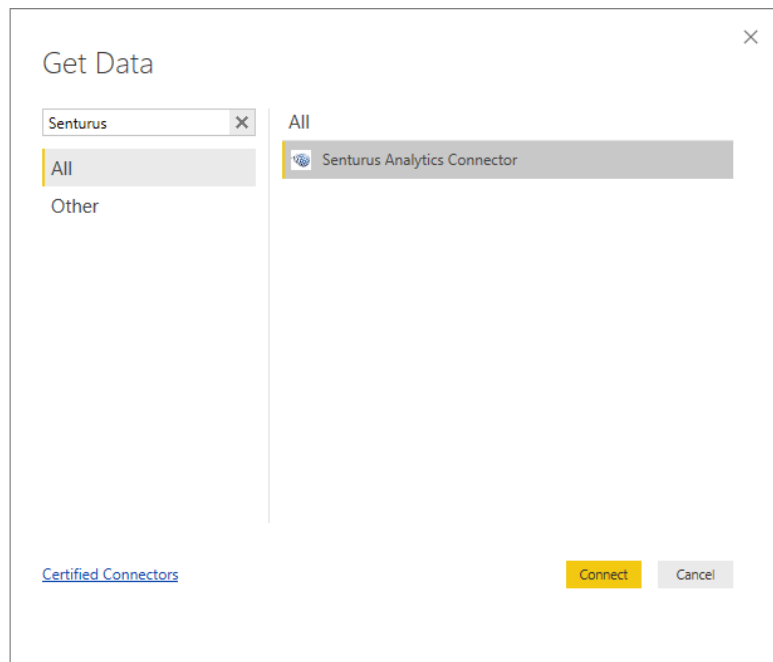
Then start Power BI Desktop, go to Files -> Options and Settings -> Options, select Security on left panel, and under "Data Extensions" option, check "(Not Recommended) Allow any extensions to load without validation or warning." You need to restart Power BI Desktop.

Connecting to Cognos from Power BI Desktop

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.

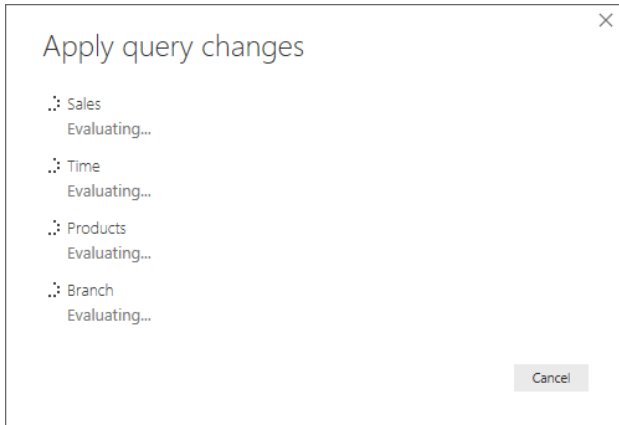
To manage or delete the stored authentication information, go to **File → Options and settings → Data source settings**.

Working with Tables

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

Select the tables you wish to include.

Click **Load**.



When complete, navigate to the Relationships view using the left navigation bar.

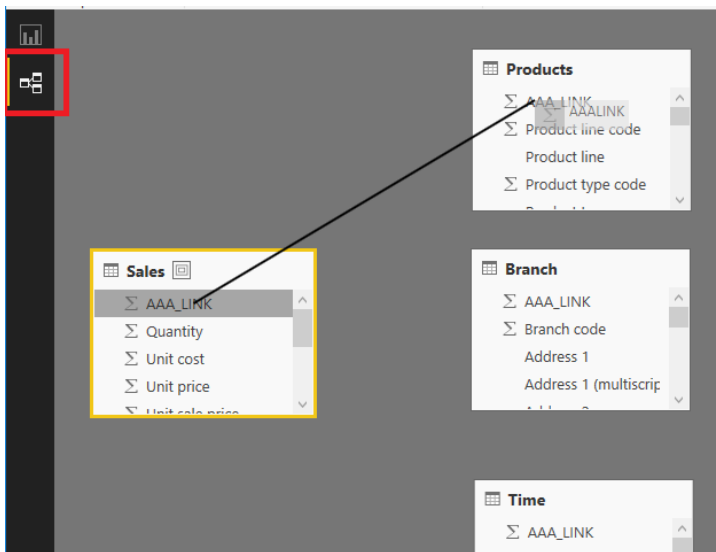
Power BI requires relationships between the tables, but Cognos typically does not present primary key and foreign keys at the business user view.

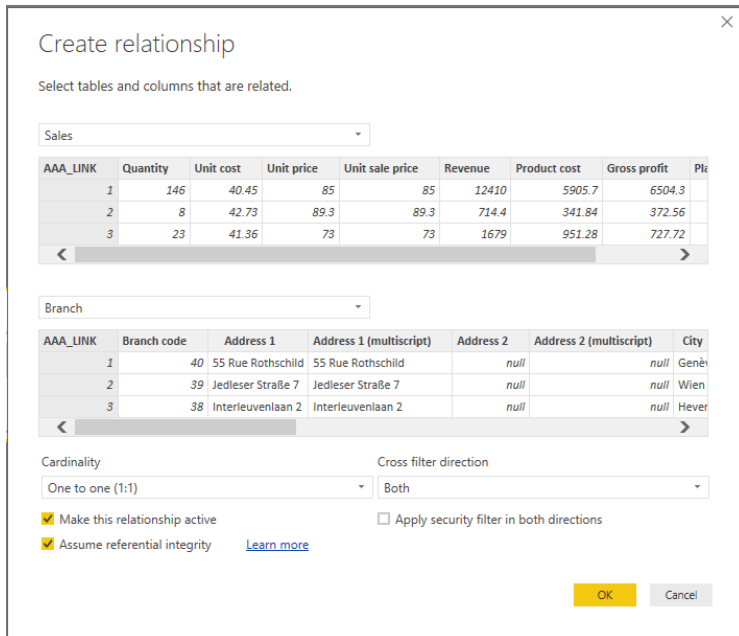
The Analytics Connector injects `AAA_LINK` columns into the tables so you can build relationships between the tables. The Analytics Connector ignores these columns when passing queries to Cognos to execute.

Create relationships from fact tables to dimension tables using **One to One (1:1)** joins as described below.

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, change the cardinality to *One to one (1:1)*.

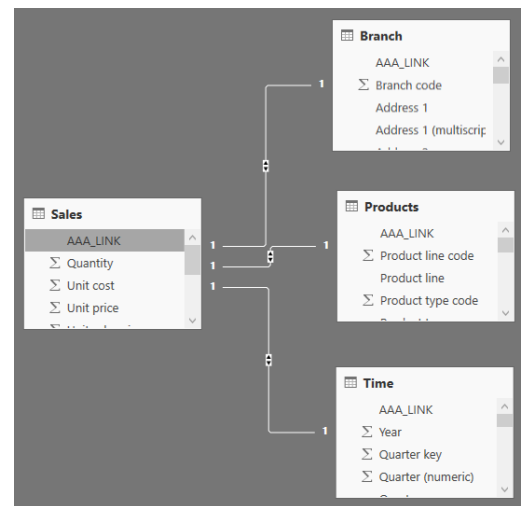
Change the Cross filter direction to *Both*.

Check the box next to *Assume referential integrity*.

Click **OK**.

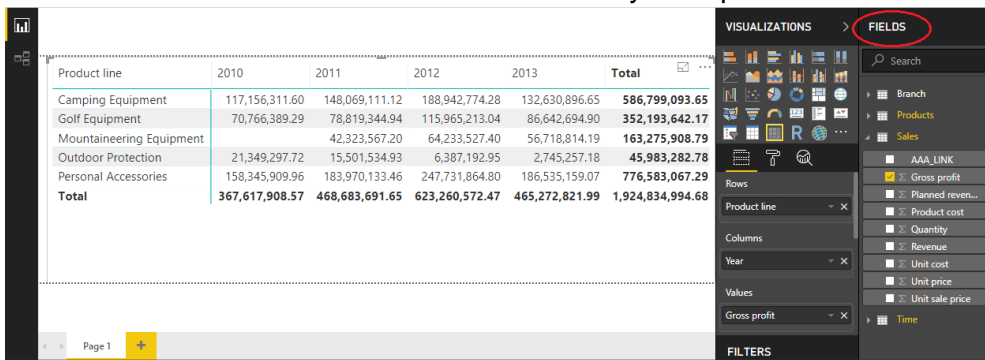
Repeat these steps until the fact table has relationships to all dimension tables.

If you have multiple fact tables, create a relationship to one or more of the dimension tables. The key is that the other fact tables are joined to the model and not isolated. Redundant relationships will be created as inactive relationships and are not necessary.



Change to the **Report** view.

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



Working with Calculations

The Analytics Connector supports working with Cognos Calculations in Power BI.

The calculations table name can be changed using the Configuration Utility. The *Standalone Calculation Table Name* setting must be set to *AAA_CALCULATIONS* as Power BI does not work with tables that begin with an underscore.

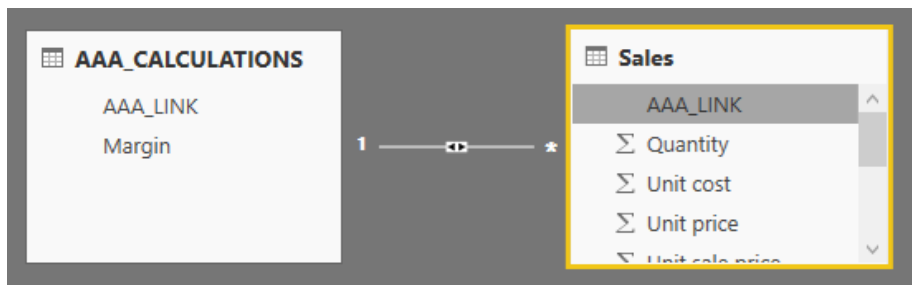
Select the *AAA_CALCULATIONS* table along with the other tables for your data set.

Navigator

AAA_CALCULATIONS
Preview downloaded on Tuesday, Nc

AAA_LINK	Margin
1	0.53636094
2	0.51427175
3	0.2812865
4	0.39511568
5	0.47232605
6	0.48575949
7	0.54
8	0.62826421
9	0.52083333
10	0.25769724
11	0.49600278
12	0.28383862
13	0.2737203
14	0.27523308
15	0.2945126
16	0.29182156

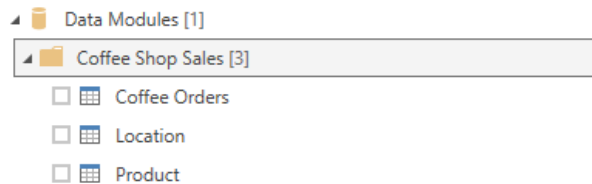
The table will also contain a link column. This special calculation table is joined to other tables in the data set using the link column as described in [Working with Tables](#).



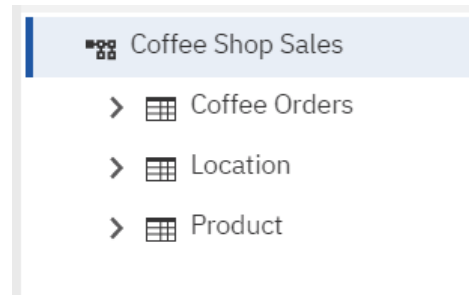
Working with Data Modules

When data modules have been configured in the Analytics connector, they work similarly to packages. Data modules are configured to display as databases in Power BI. After connecting to the database, you will see a list of tables from the data module.

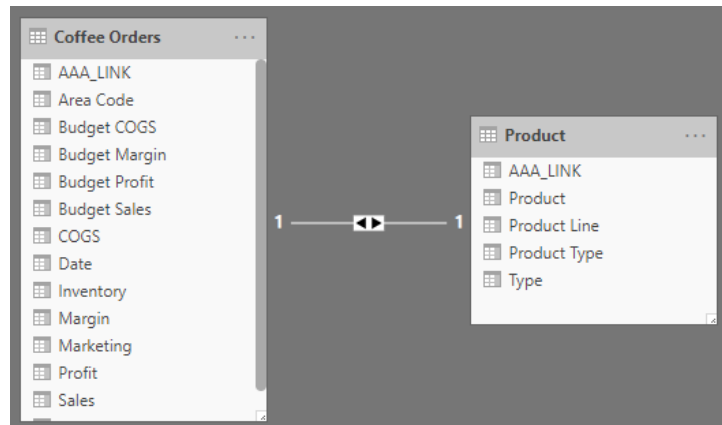
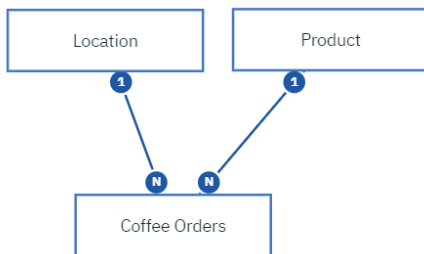
Power BI View



Cognos View



If the data module tables have a defined relationship in Cognos, you can join them using the AAA_LINK column, even if the relating columns are hidden.



Tables can also be joined in Power BI if the columns that relate the tables are exposed by the data module.

Working with Parameterized Tables

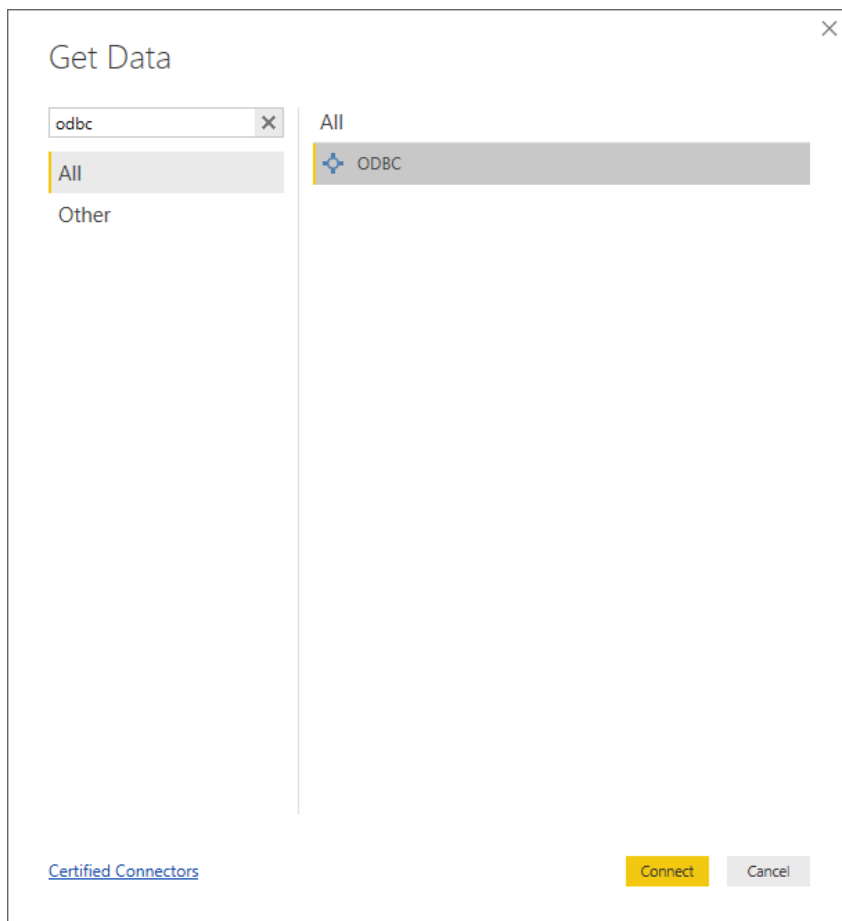
The Analytics Connector supports working with Cognos parameterized tables in Power BI.

To use a parameterized table in a Power BI report and pass parameter values, you must first set up a DSN. For information on how to setup a DSN, see the *Senturus Analytics Connector Installation Guide*.

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

Find the *ODBC* data source in the list.

Click **Connect**.



The *From ODBC* dialog will display.

From ODBC

Data source name (DSN)
Senturus Colo .45

Advanced options

Connection string (non-credential properties) (optional) ⓘ
Example: Driv..

SQL statement (optional)
Select * 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 "Parameterized Table"

Supported row reduction clauses (optional)
(None) Detect

OK Cancel

Using the *OpenTable* function, you can pass parameters to a parameterized Cognos table.

The first parameter is “[*schema name*].”[*table name*]” of the Cognos parameterized table. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

Example passing static valued parameters:

```
Select * 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 "Parameterized Table"
```

Select the appropriate DSN from the list.

Place the query in the SQL statement section of the *From ODBC* dialog.

Click **OK**.

A preview of the table will display. Click **Load**.

ODBC (dsn=Senturus Colo .45 Select * From (OpenTable("Sales (query)". "Para...

AAA_LINK	Base product number	Discontinued date	Introduction date	Product	Product brand	Product brand
1	101	null	12/15/2009 12:00:00 AM	Hailstorm Steel Irons	Hailstorm	
2	115	null	12/27/2009 12:00:00 AM	Course Pro Gloves	Course Pro	
3	109	null	12/10/2009 12:00:00 AM	Course Pro Putter	Course Pro	
4	103	null	12/10/2009 12:00:00 AM	Lady Hailstorm Steel Irons	Hailstorm	
5	105	null	12/27/2009 12:00:00 AM	Hailstorm Titanium Woods Set	Hailstorm	
6	102	null	12/10/2009 12:00:00 AM	Hailstorm Titanium Irons	Hailstorm	
7	104	null	12/18/2009 12:00:00 AM	Lady Hailstorm Titanium Irons	Hailstorm	
8	106	null	12/5/2009 12:00:00 AM	Hailstorm Steel Woods Set	Hailstorm	
9	110	null	12/10/2009 12:00:00 AM	Blue Steel Putter	Blue Steel	
10	107	null	1/13/2010 12:00:00 AM	Lady Hailstorm Titanium Woods Set	Hailstorm	
11	108	null	12/27/2009 12:00:00 AM	Lady Hailstorm Steel Woods Set	Hailstorm	
12	111	null	12/15/2009 12:00:00 AM	Blue Steel Max Putter	Blue Steel	
13	114	null	12/15/2009 12:00:00 AM	Course Pro Golf Bag	Course Pro	
14	112	null	1/10/2010 12:00:00 AM	Course Pro Golf and Tee Set	Course Pro	
15	113	null	1/15/2010 12:00:00 AM	Course Pro Umbrella	Course Pro	

Load Edit Cancel

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

Product	Product line	Product size
Blue Steel Max Putter	Golf Equipment	One-size
Blue Steel Putter	Golf Equipment	One-size
Course Pro Gloves	Golf Equipment	Medium
Course Pro Golf and Tee Set	Golf Equipment	Unspecified
Course Pro Golf Bag	Golf Equipment	Unspecified
Course Pro Putter	Golf Equipment	One-size
Course Pro Umbrella	Golf Equipment	Large
Hailstorm Steel Irons	Golf Equipment	Men's
Hailstorm Steel Woods Set	Golf Equipment	Men's
Hailstorm Titanium Irons	Golf Equipment	Men's
Hailstorm Titanium Woods Set	Golf Equipment	Men's
Lady Hailstorm Steel Irons	Golf Equipment	Women's

You can also join parameterized tables with other tables in one Power BI data source. These tables will also join using the generated Link Columns. Example advanced query:

Select

```
"Products"."Product line",  
"Products"."Product type",  
Sum("Sales"."Revenue") as "Revenue"
```

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 "Products"  
join "Sales (query)"."Sales" as "Sales"  
  on ("Products"."AAA_LINK" = "Sales"."AAA_LINK")
```

Group by "Products"."Product line"

Cont'd

Working with Reports

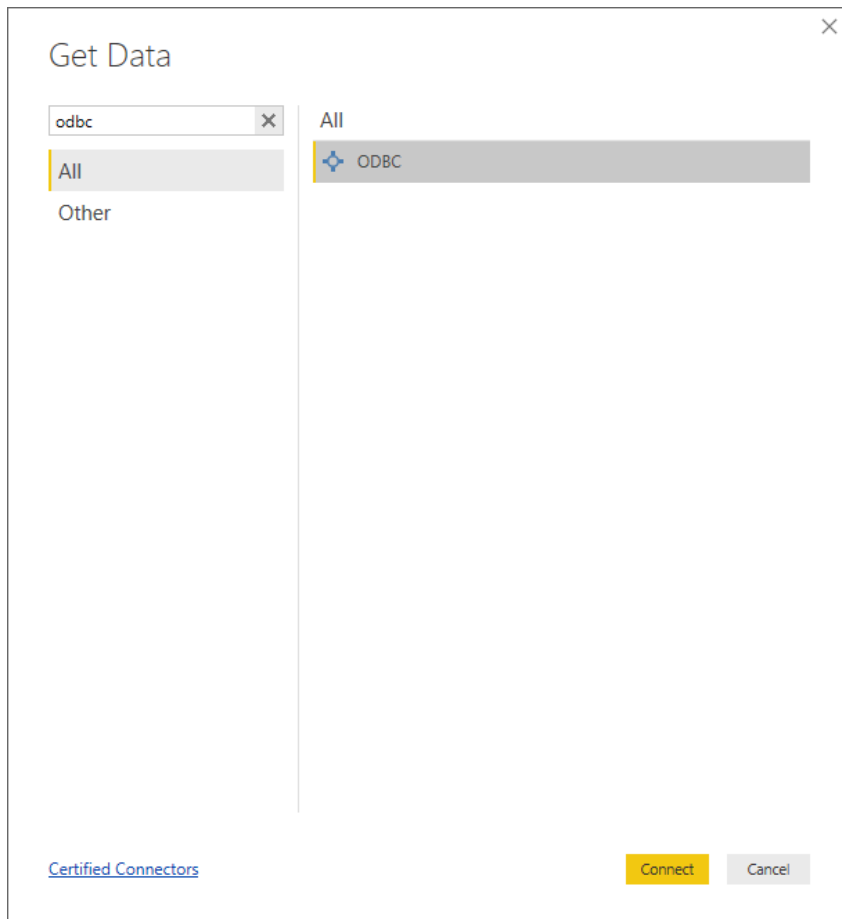
The Analytics Connector supports working with Cognos reports in Power BI.

To use a parameterized table in a Power BI report and pass parameter values, you must first setup a DSN. For information on how to setup a DSN, see the *Senturus Analytics Connector Installation Guide*.

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

Find the *ODBC* data source in the list.

Click **Connect**.



The *From ODBC* dialog will display.

Using the *RunReport* function, you can pass parameters to a Cognos report.

The first parameter is "[schema name].[report name]" of the Cognos report. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

Example passing static valued parameters:

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

Select the appropriate DSN from the list.

Place the query in the SQL statement section of the *From ODBC* dialog.

Click **OK**.

The screenshot shows the 'From ODBC' dialog box with the following fields and values:

- Data source name (DSN):** Senturus Colo.45
- Advanced options:**
 - Connection string (non-credential properties) (optional):** Example: Driv...
 - SQL statement (optional):**

```
Select * From (  
  RunReport (  
    "Connector Reports"."Parameter Report",  
    "p_product line code"="[991,992,993,994,995]",  
    "p_Product Line"="Golf Equipment",  
    "p_Category"=["Books', 'Electronics', 'Movies']",  
    "p_Date"="2010-01-01",  
    "p_Datetime"="2010-02-01 19:00:00"  
  )  
) as "Report"
```
- Supported row reduction clauses (optional):** (None)

Buttons: OK, Cancel

A preview of the report will display.

Click **Load**.

ODBC (dsn=Senturus Colo .45 Select * From (RunReport ("Connector Reports"....

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

Load Edit Cancel

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

Product line	Product type	Revenue	Year
Golf Equipment	Golf Accessories	864,227.83	2010
Golf Equipment	Irons	4,654,281.10	2010
Golf Equipment	Putters	2,459,044.00	2010
Golf Equipment	Woods	5,138,304.03	2010
Total		13,115,856.96	8040

Visualizations > **FIELDS** >

Search

Query1

- Product line
- Product line c...
- Product type
- Product type c...
- Revenue
- Year

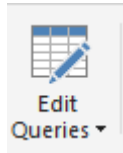
Values

Product line

Product type

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.

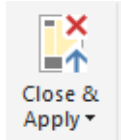
The image shows the "Parameters" dialog box in Power BI. On the left, there is a list of parameters with "Start_Date" selected. On the right, the configuration for "Start_Date" is shown: Name: Start_Date, Description: Start date to pass to the parameterized table, Required: checked, Type: Date, Suggested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

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.

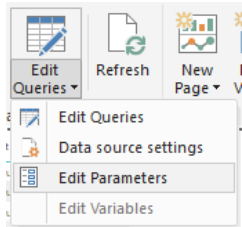
The below example formats a date parameter to text and appends it to the SQL statement.

```
= Odbc.Query("dsn=Senturus Colo .45", "Select * From (#(1f)OpenTable(#(1f) ""Sales (query)"". ""Parameterized Products"", #(1f) ""p_product line code""=""[991, 992, 993, 994, 995]""", # (1f) ""p_Product Line""=""Golf Equipment'""", #(1f) ""p_Date""="" & Date.ToText(Start_Date, ""yyyy-MM-dd"")) & """"", #(1f) ""p_Datetime""=""2010-01-01 19:00:00'""""#(1f))#(1f)) as ""Parameterized Table""")]
```

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



Click **Close & Apply**.



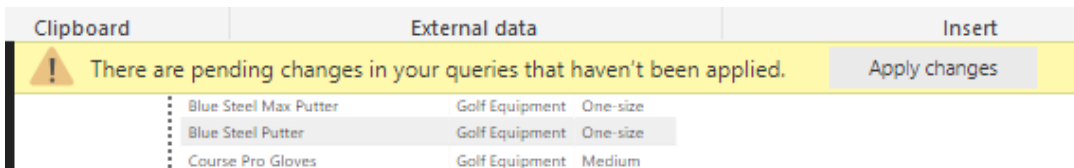
To change the Parameter value(s), select *Edit Parameters* under **Edit Queries**.

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

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



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



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, as well as to install Senturus Analytics Connector client on the computer where Power BI Gateway is running. For more information, please refer to Senturus Analytics Connector Installation Guide.

Once you published a report onto powerbi.com, you will see a report as well as a dataset (with the same name as your report) in the destination folder. But you cannot open that report on powerbi.com because the dataset is not associated with any data source yet.

First, you need to create a data source.

Data Source Settings Users

Data Source Name

New data source

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

user

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):

The screenshot shows the 'Senturus Analytics Connector' dialog box. It contains the following fields and options:

- Server:** A dropdown menu with 'internal-bi1.senturus.com' selected.
- Port:** A dropdown menu with '5432' selected.
- Database:** A dropdown menu with 'GO Sales (query)' selected.
- Data Connectivity mode:** Two radio buttons: 'Import' (unselected) and 'DirectQuery' (selected).
- Buttons:** 'OK' (yellow) and 'Cancel' (white) buttons at the bottom right.

After that, you need to associate the new data source with your report dataset:

The screenshot shows the 'Datasets' tab in Power BI. The left sidebar lists several datasets, with 'GO Sales (query) data connector postgres' selected. The main area displays the settings for this dataset:

- Settings for GO Sales (query) data connector postgres**
- The data gateway bi1.gateway is online and your data sources are online.
- [Refresh history](#)
- Gateway connection**
- To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click apply.
- Use a data gateway: On
- | Gateway | Department | Contact information | Status | Actions |
|-------------|------------|---------------------|-------------------------|---------|
| bi1.gateway | | xgao@senturus.com | Running on INTERNAL-BI1 | ⚙️ |
- Data sources included in this dataset:**
- ```
Extension{"extensionDataSourceKind":"SenturusAnalyticsConnector","extensionDataSourcePath":{"server":"internal-bi1.senturus.com","port":"5432","database":"GO Sales (query)"}}
```
- Maps to: Data Connector to AC\_... (dropdown)
- 

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

## Technical Reference

### Function 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.

## Supported ODBC Features

All identifiers (catalog, schema, table and column names) are case insensitive. This ODBC driver supports 15 Cognos data types. Refer to the data type mapping in following table.

| Cognos Data Type  | ODBC Data Type | Max Length/Precision | Note              |
|-------------------|----------------|----------------------|-------------------|
| int16             | smallInt       | 5                    |                   |
| int32             | integer        | 10                   |                   |
| int64             | bigInt         | 19                   |                   |
| float32           | real           | 7                    |                   |
| float64           | double         | 15                   |                   |
| decimal           | decimal        | 38                   |                   |
| character         | char           | 8000                 |                   |
| characterLength16 | varchar        | 8000                 |                   |
| nChar             | nchar          | 4000                 |                   |
| nVarChar          | nvarchar       | 4000                 |                   |
| date              | date           | 10                   |                   |
| time              | time           | 8                    | no milliseconds   |
| datetime          | timestamp      | 19                   | no milliseconds   |
| textBlob          | varchar        | 8000                 | mapped to varchar |
| unknown           | varchar        | 8000                 | mapped to varchar |

This driver has been tested against SQL Server, DB2 and Oracle databases (relational and DMR model). It may not support all data types/functions for other databases.

This driver only supports a) packages with one data source or b) packages with multiple data sources. However, all database connections are of the same type (e.g. SQL Server native connection). It may work for other packages, but some functions may fail.

### Query features supported by this driver

Numeric, string, date/time, null literals

Cast/convert

Simple calculation (+, -, \*, /, and %)

String concatenate\And, or, not logical operators

Comparison (<, >, =, <=, >=, !=, between, not between, in, not in, like, not like)

Is null, is not null

Case (if else) statement

Parenthesis

Sub queries

### Functions supported by this driver

**Aggregate function:** AVERAGE, AVG, COUNT, MAX, MAXIMUM, MIN, MINIMUM, SUM, TOTAL.

**Numeric functions:** ABS, ACOS, ASIN, ATAN, CEILING, COS, COT, DEGREES, EXP, FLOOR, LOG, LOG10, MOD, POWER, RADIANS, ROUND, SIGN, SIN, SQRT, TAN, , TRUNC, TRUNCATE.

**Date and timestamp functions:** CURRENT\_DATE, CURRENT\_TIMESTAMP, DAY, DAYNAME, DAYOFMONTH, DAYOFWEEK, DAYOFYEAR, HOUR, MINUTE, MONTH, MONTHNAME, QUARTER, SECOND, TIMESTAMPADD, TIMESTAMPDIFF, WEEK, YEAR.

**Character functions:** ASCII, CHAR, CHAR\_LENGTH, CHARACTER\_LENGTH, CONCAT, CONTAINS, ENDS\_WITH, LCASE, LCASE, LEFT, LEN, LENGTH, LOCATE, LTRIM, LTRUNC, REPLACE, RIGHT, RTRIM, RTRUNT, SPACE, STARTS\_WITH, STRPOS, SUBSTR, SUBSTRING, TEXTCAT, TO\_CHAR, TO\_DATE, TO\_TIMESTAMP, TRIM, UCASE, UCASE.

**Others:** CONVERT, IFNULL, COALESCE.

[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](mailto:CustomerSupport@senturus.com).