insightsoftware

Power ON Visual Documentation and Troubleshooting Guide

Version 24.2

July 2024

Introduction to Power XL Table v24.2

This document describes Power ON's write-back capable tool, Power XL Table (PXLT). The product is a custom visual developed for Microsoft Power BI, that enhances the user experience by enabling end-users to make permanent changes to data (aka write-back) enriching it with text input controls, such as those commonly seen on modern HTML pages (e.g., date pickers, dropdowns, rich text) besides the traditional look and feel.

On the following pages you will see:

- A detailed overview of Power XL Table.
- How to use it in reports along with common use-case implementations.
- How to configure it correctly.
- What pitfalls one might encounter and how to overcome them.

Audience

The audience of this document are both technicians (developers, DBAs, BI professionals) who are aware of the depths of SQL Server, SSAS Tabular models, Azure Services, Excel and Power BI, but also users who mostly focus on building and preparing reports using Excel or Power BI Desktop.

Some sections are focusing on more technical specific subjects.

> **Note:**
> If you encounter any issues or queries and need assistance, reach out to your local IT team or log in to https://help.insightsoftware.com/ to submit a ticket, and our support team will provide you with the best service. Additionally, once you have registered and logged in, you can access the Power ON Knowledge Base articles that cover common use cases, tips, and troubleshooting tools.

Overview

Power XL Table lets users edit any type of row-level information in Power BI. Most typical use case is Master Data Management (e.g., customers, products, materials etc.) adding or removing records, editing existing members, changing attributes as well as performing simple data entry workloads. Users can consider this visual as a replacement for an existing, simple form like application, designed for master data.

Power XL Table offers the following built-in features and components:

- Writing-back values to the underlying data source
- Date Time or Month or Time picker for DATETIME data type columns
- Color picker for color like data type columns

- Check Box or Radio button control for true or false data type columns
- Dropdown list (called Selection) control
- Supports copy and paste selected cells to or from Power XL or Microsoft Excel
- Conditional formatting for any type of columns
- Setting tens of predefined table formats, cell styles, chart options, sparklines and many more

It is worth describing at a high level how write-back works for Tabular models. Depending on your data source (SSAS In-Memory, SSAS Direct Query or SQL Server, Power BI Premium dataset), Power ON Write-Back Service performs the following operations:

- Captures the modified value along with its tuple and the user context. After determining the necessary actions, the service identifies which T-SQL statements should be executed based on the changes and the structure of the model.
- Executes the compiled T-SQL statement against the underlying data source table to save the modifications
- In case of Semantic model (Power BI dataset or SSAS model), it will reprocess the table.
- Initiates a refresh on the visual, so that the changes will appear in the report.

Versions

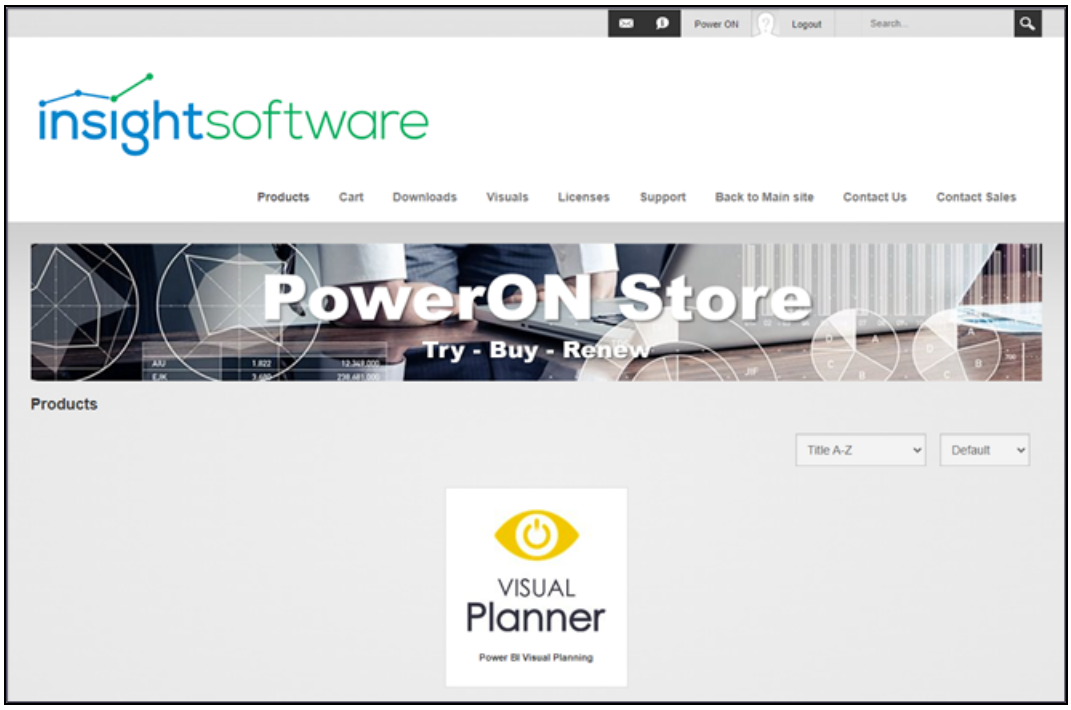
This document describes the latest version of Power XL Table.

The most recent visual is PowerXL_table.1.3.2.pbiviz.

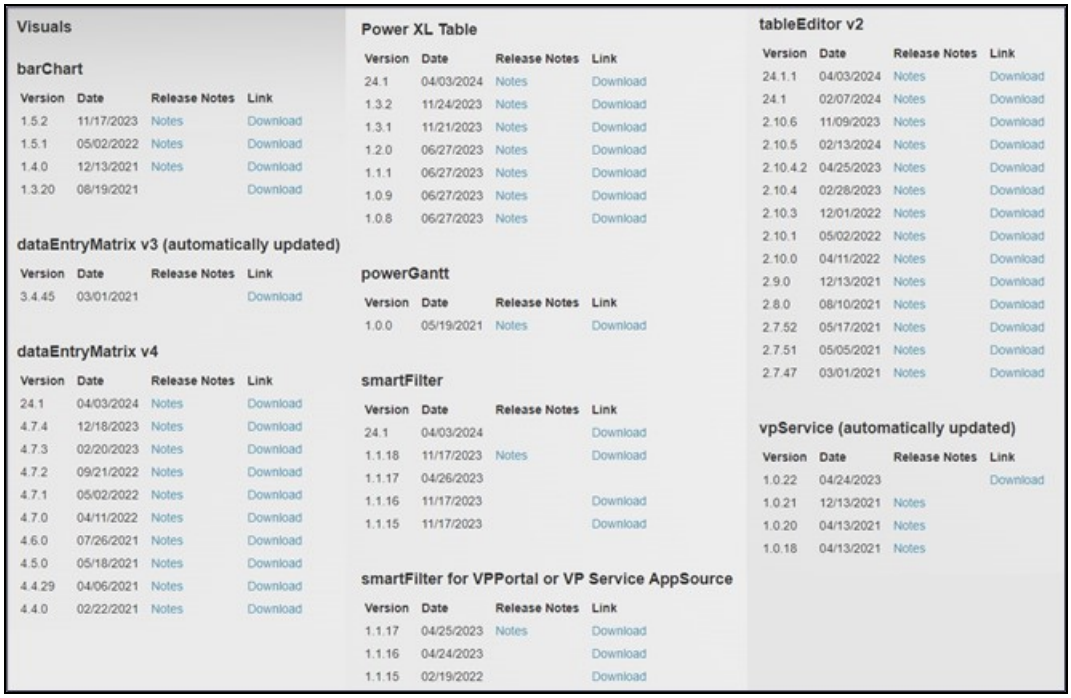
How can you download the different versions?

1. Register at **store.poweronbi.com** site with the same domain name you have registered originally. During the registration, please make sure to use the email address for both the user-name and email address field.

2. After login, you will see the following platform:



3. Click on **Visuals** in upper right corner to find our available write-back capable visuals and their different versions.



Visuals

barChart

Version	Date	Release Notes	Link
1.5.2	11/17/2023	Notes	Download
1.5.1	05/02/2022	Notes	Download
1.4.0	12/13/2021	Notes	Download
1.3.20	08/19/2021		Download

dataEntryMatrix v3 (automatically updated)

Version	Date	Release Notes	Link
3.4.45	03/01/2021		Download

dataEntryMatrix v4

Version	Date	Release Notes	Link
24.1	04/03/2024	Notes	Download
4.7.4	12/18/2023	Notes	Download
4.7.3	02/20/2023	Notes	Download
4.7.2	09/21/2022	Notes	Download
4.7.1	05/02/2022	Notes	Download
4.7.0	04/11/2022	Notes	Download
4.6.0	07/26/2021	Notes	Download
4.5.0	05/18/2021	Notes	Download
4.4.29	04/06/2021	Notes	Download
4.4.0	02/22/2021	Notes	Download

Power XL Table

Version	Date	Release Notes	Link
24.1	04/03/2024	Notes	Download
1.3.2	11/24/2023	Notes	Download
1.3.1	11/21/2023	Notes	Download
1.2.0	06/27/2023	Notes	Download
1.1.1	06/27/2023	Notes	Download
1.0.9	06/27/2023	Notes	Download
1.0.8	06/27/2023	Notes	Download

powerGantt

Version	Date	Release Notes	Link
1.0.0	05/19/2021	Notes	Download

smartFilter

Version	Date	Release Notes	Link
24.1	04/03/2024		Download
1.1.18	11/17/2023	Notes	Download
1.1.17	04/26/2023		
1.1.16	11/17/2023		Download
1.1.15	11/17/2023		Download

smartFilter for VPPortal or VP Service AppSource

Version	Date	Release Notes	Link
1.1.17	04/25/2023	Notes	Download
1.1.16	04/24/2023		Download
1.1.15	02/19/2022		Download

tableEditor v2

Version	Date	Release Notes	Link
24.1.1	04/03/2024	Notes	Download
24.1	02/07/2024	Notes	Download
2.10.6	11/09/2023	Notes	Download
2.10.5	02/13/2024	Notes	Download
2.10.4.2	04/25/2023	Notes	Download
2.10.4	02/28/2023	Notes	Download
2.10.3	12/01/2022	Notes	Download
2.10.1	05/02/2022	Notes	Download
2.10.0	04/11/2022	Notes	Download
2.9.0	12/13/2021	Notes	Download
2.8.0	08/10/2021	Notes	Download
2.7.52	05/17/2021	Notes	Download
2.7.51	05/05/2021	Notes	Download
2.7.47	03/01/2021	Notes	Download

vpService (automatically updated)

Version	Date	Release Notes	Link
1.0.22	04/24/2023		Download
1.0.21	12/13/2021	Notes	
1.0.20	04/13/2021	Notes	
1.0.18	04/13/2021	Notes	

4. Pick the preferable version of Power XL Table and click on Download.
   a. You can even download its Release notes, if you like.

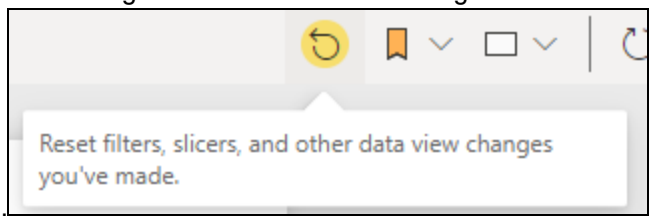
Rules and Limitations

As all custom visuals developed for Power BI, due to Microsoft's policies Power XL Table also has the following limitations:

- Renaming editable columns and tables in the Power BI report are not supported. They must be the same as they are called in the used data source.
- In case of Semantic models, the names should be equal to the names of the entities (visible in the model or the report Fields sections).
- In case of SQL data source, the names of the entities should be the same as they are in the SQL tables (e.g., if the table in SQL is called Customer, it should NOT be called Customers in the report). This rule should be also applied for the names of the fields or columns.
- The characters "[" and "]" (brackets) are not supported in measure names.
- The character "." (dot) is not supported in table names, but it can exist in schema names. If there are unsupported characters in your table name, we recommend creating a view on top of the table that complies with the required format and use that view when writing back.
- All custom visuals developed for PowerBI have a default limitation set by Microsoft which is that 30.000 records can be displayed at one time. If the underlying table has more rows, consider using slicers in the report. This will allow editing only a subset of the data and ensure a seamless workflow. However, as a best practice, reducing the displayed table size with slicers is recommended for better performance.
- Write-back to measures is not supported by Power XL Table. Only row level data can be modified. If there is a need to change the values of measures, the Data Entry Matrix (DEMx) visual should be utilized instead. Power XL Table is designed to be used in situations where modifications to descriptive, dimension-type data are required.
- Non-writable views that serve as a data source for the fact table are not supported for write-back. To address this, either materialize the view into a physical table, reduce the complexity of the view, or create INSTEAD OF INSERT | UPDATE triggers to manage the operation. Typically, non-writeable views contain complex SQL queries with multiple joins, CASE statements in WHERE clauses, CTEs (Common Table Expressions), and aggregations. The view can be tested by duplicating it under a different name and executing an INSERT statement against it.
- In case of on-premise Power BI Report Server at least January 2019 version is required along with Power BI Desktop for Report Server 2019 January or more recent versions.
- In the case of older versions of Power XL Table than v1.2.0 after changing to a bookmark, users may experience that a custom column appears where a PBI column was located previously. The solution to this situation is to save the original layout (not the bookmarked one)
  - The [Save Layout] button on the Designer Ribbon

- The [Save this report] button in Edit mode.

- If the “Reset filters, slicers, and other data view changes you've made.” Yellow button indicates that the report is modified, then it has to be pressed before going to ‘Edit’ mode. Otherwise, the visual will have the settings which were set in ‘Reading view’ to the bookmark, not the original report settings.



- Cutting rules
  - Cutting out the whole data table is allowed in ‘Edit’ mode, but not possible in ‘Reading View’
  - Cutting individual columns is disabled in ‘Edit’ mode no matter what the column type is.

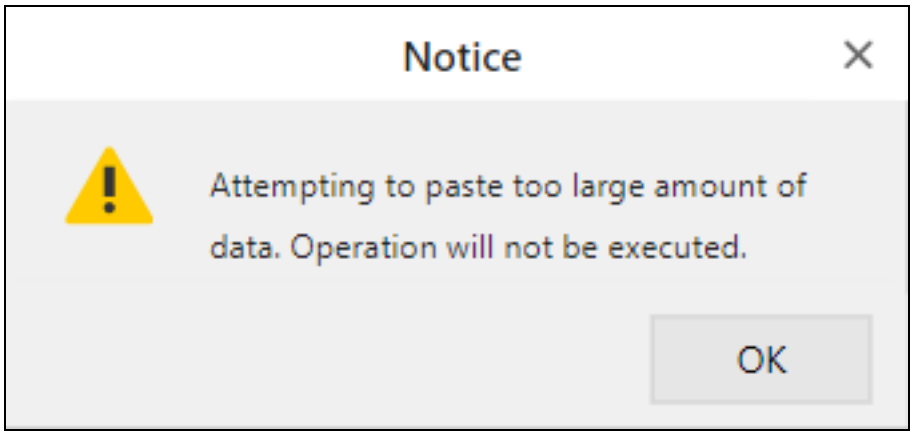
Copy-Paste Limitation

Starting from this version, there are limitations on the number of cells that can be pasted at once in the visual. The specific limits depend on the user's context and the type of sheet they are working on. The limitations are as follows:

- If the user is on a sheet that is not the Data sheet (applies to Edit Mode, Reading View, and other views):
  - The limit is 40,000 cells for pasting operations.
- If the user is on the Data Sheet in Reading View:
  - The limit is 40,000 cells for pasting operations.
- If the user is in Edit Mode and wants to paste into a datatable that contains formulas:
  - The limit is 3,000 cells for pasting operations.
- If the user is in Edit Mode and wants to paste into a datatable that contains the =IFISROWDIRTY() formula outside of the range where the user intends to paste:
  - The limit is 1,000 cells for pasting operations.

If the user attempts to paste more cells than the specified limitations, a pop-up message will appear to notify them about the exceeding limit. This is done to ensure optimal performance and prevent potential issues that may arise from pasting an excessive number of cells.

By adhering to these limitations, users can work within the defined boundaries and maintain the overall functionality and performance of the visual.


Notice
Attempting to paste too large amount of data. Operation will not be executed.
OK

Power XL Table Prerequisites

This topic outlines the necessary prerequisites for using the Power XL Table visual.

Write-back Service

There must be a working and configured Write-Back Service (PPWebService) installed in your environment. If the Write-Back Service is installed and configured, in order to build a report, first import the Power XL Table and VPService pbiviz files into the Power BI Desktop instance. This must be done for each report which intends to use PXLT.

The visuals are stored in the report itself, so it allows using the write-back capabilities when opened. In case of PowerBI cloud services there is an option to store the custom visual in a centralized repository for easier management. For more information, refer to the following link: https://docs.microsoft.com/en-us/power-bi/developer/visuals/power-bi-custom-visuals-organization

For more information about configuring the PPWebService, log in to https://help.insightsoftware.com/ and refer to the PowerON Knowledge Base articles.

In case of Azure, the Web Applications are accessible by default. The Web Service must have a dedicated connection string configured for the data source in the web.config 'connectionStrings' section that is pointing to:

- In case of the used Power BI dataset or SSAS model: to the used SSAS Cube
- In case of SQL only models: the used SQL database.

The web.config file can be found under the WWWRoot folder of the PPWebService installation directory.

Your IT team should configure connections to the data sources. Refer to the How to Add a New Data Source for Write-back article in the knowledge base for more information. You can find valid connection string examples for different data sources at https://www.connectionstrings.com/.

The PPWebservice service account used in the connection string must have the following permissions:

- In case of SSAS models: administration right on the cube, plus data reader and data write roles on the underlying data source database of the cube.
- In case of SQL-only models: data reader and data write roles on the SQL database.

The end users (or the user / Active Directory group they are in) must have:

- In case of SSAS models: data reader membership.
- In case of SQL only models: data reader role.

If impersonation is enabled, then the end users must have data writer role on the underlying SQL databases.

VPService Visual

The VPService is a helper visual that establishes the connection between PXLT and the Write-Back Service. VPService is updated automatically from the web, eliminating the need for manual updates, unlike other non-legacy versions of the Visuals.

Key benefits:

- Faster Release Processes - Switching from one version of a visual to another becomes much faster. This results in quicker support from our side, especially when implementing new features.
- Easy Version Switching - Users can seamlessly switch between different versions of the visuals.

Key functions:

- Saving Changes - VPService ensures that any modifications made by users are saved appropriately.
- Saving Comments - In visuals like the Data Entry Matrix, VPService handles the saving of comments.
- User Information - VPService sends relevant user information as needed.
- Pop-Up Windows - For visuals like the Table Editor, VPService handles the opening of pop-up windows.

Power XL Table Setup and Usage

This topic covers the fundamental aspects of creating a write-back capable report using the Power XL Table visual.

It describes the available configuration options for Power XL Table and includes short step-by-step instructions for building a simple report using PXLT. The details of the configuration elements will be presented later in this topic.

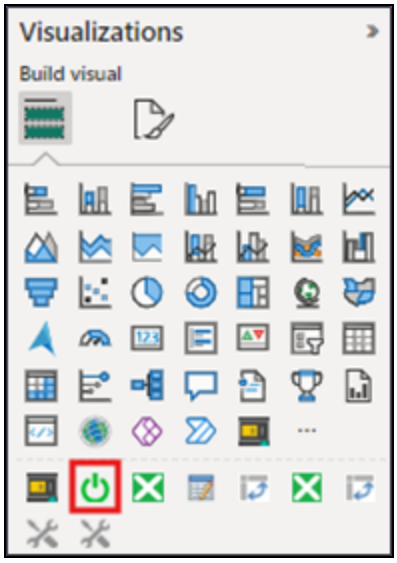
Setting up the Basics

After Power BI Desktop is launched, a connection to a data source is established, and the visuals (VPService, Power XL Table) are imported into your report, the VPService visual needs to be configured.

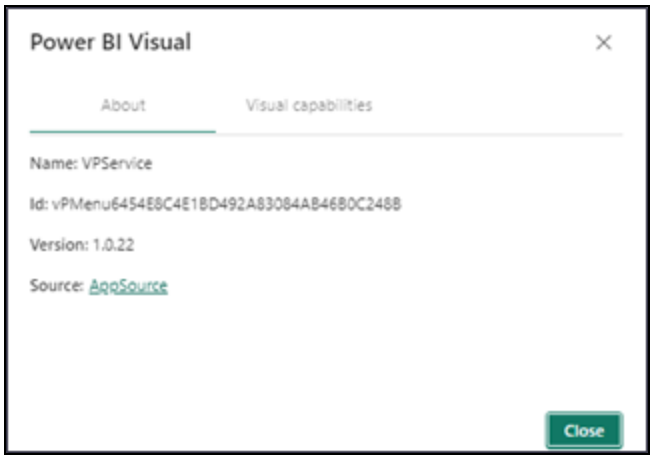
We have to tell the visuals where they can find the Write-Back Service and which connection they should use during the process.

Configuring the Visual

1. After importing VPService, the following icon appears on the **Visualizations** tab.

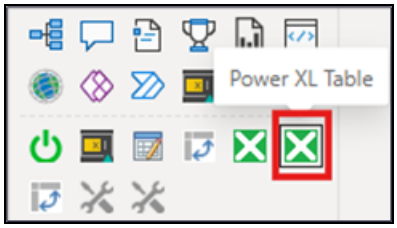


The visual version can be checked by right-clicking on the icon and selecting **About** from the menu. A pop-up window will appear with the version information:

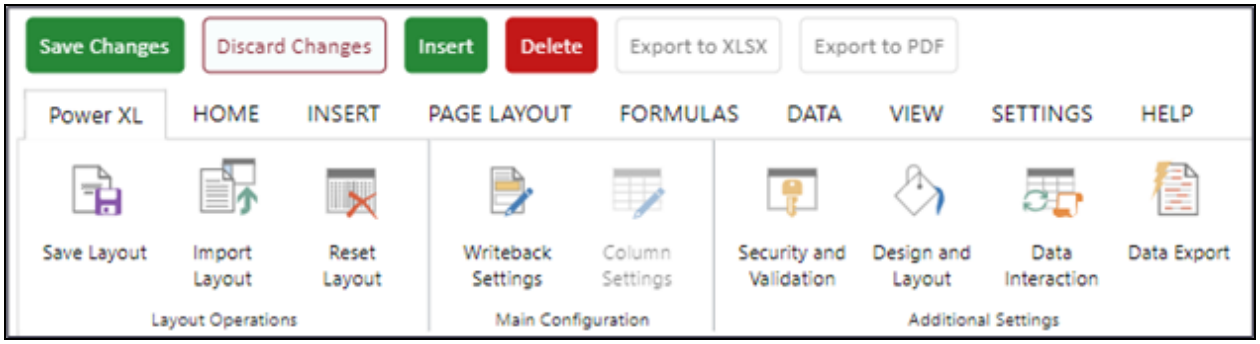


2. Add the VPService visual to the report by clicking the icon described above.
3. Add a field to the VPService visual, for example, an ID field.

4. Add the Power XL Table visual to the report by clicking the following icon.

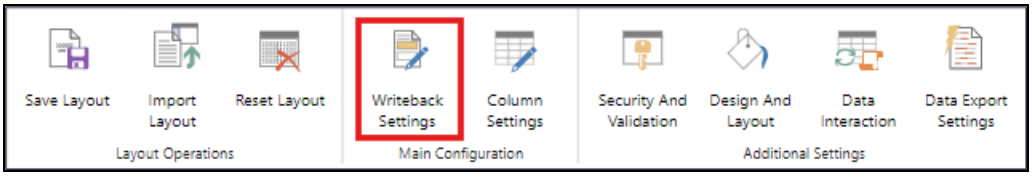


5. Add a field to the Power XL Table visual. This will cause the Designer Ribbon to appear, where the configuration options for the visual are available.

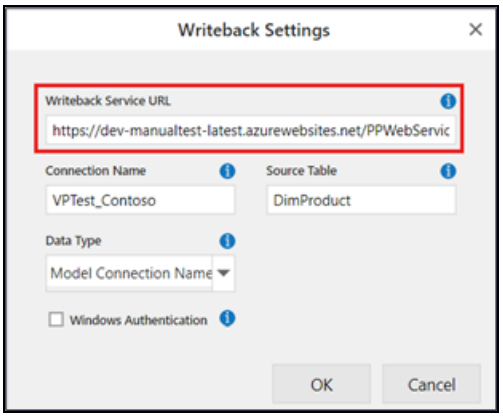


6. The Web Service URL of the VPService visual and the Writeback Service URL of the Power XL Table visual must be identical.
   a. The Web Service URL is on the Visualizations » Format Your Visual Tab of the VPService visual.

b. The Writeback Service URL setting of the Power XL Table visual can be accessed by clicking the following icon.



The following pop-up window opens. The Writeback Service URL can be set up in the marked are below.



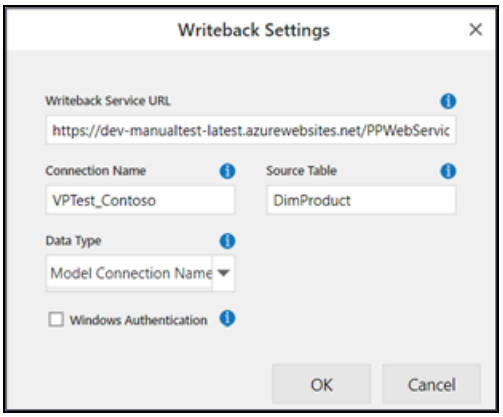
Specific Details

The target table for the write-back must have a primary key defined with the following to keep in mind:

- Composite keys are supported.
- In the case of Semantic data source, the primary key must be part of the Semantic model and visible for end users, and it must be added to the visual as a field.
- If there are any specific business rules for creating a key for a new record, either a custom trigger needs to be created in the database to provide the new value during the operation, or if

the key can be computed based on existing data visible in the model with DAX, the Computed Column property can be utilized.

The Writeback Service URL, Source Table, Connection, and Type properties must be set in the **Power XL** tab of the designer ribbon. Clicking on the **Writeback Settings** button causes the following pop-up window to appear.



Writeback Service URL

Here, the Write-Back Service URL must be set, which should point to the hosting machine - or app service - where it was installed. Typical URL is:

http(s)://WEBSERVICE_COMPUTER_NAME/PPWebservice/PPWebservice.svc

Be careful not to have an extra slash at the end of the URL.

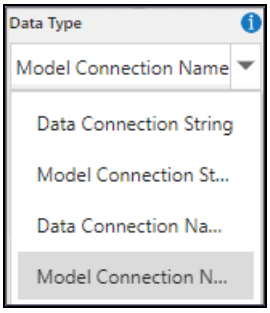
Connection

Specify the name of the connection string defined in the web.config file of the Webservice, or the connection string directly, or leave it blank depending on the “Type” setting.

Note: The maximum length of this property is 250 characters.

Data Type

This option refers to the connection type that the Write-Back Service uses.



The setting is required.

- Data Connection Name (former SQL Datasource): Connection is set in the PPWebService web.config. It must be referred to by name in the “Connection” property.
- Model Connection Name (former SSAS Datasource): Set a connection name in the “Connection” property defined in the web.config of the webservice.
- Data Connection String (former SQL): Set the connection string directly in the “Connection” property. If the default “SQLConnection” is set in the web.config, leave it blank, and it will automatically look for that connection.
- Model Connection String (former SSAS): Set the connection string directly in “Connection” property. Leave it blank if the default "SSASConnection" is set in the web.config of the webservice.

These last two options are used for testing and developing purposes primarily.

Source Table

Ensuring the correct configuration of this property is crucial, as any misconfiguration could result in save errors. We highly recommend thoroughly reviewing this chapter for precise guidance.

This property instructs the Write-Back Service to identify the target table for data modifications. Ensure the accurate entry of the target table's name here. Note that the property is case-sensitive. For instance, if fields from the Product table have been incorporated into the visual, the property value should match (“Product”), indicating the intention to save modifications to the product table.

Tables or columns in the report should not be renamed.

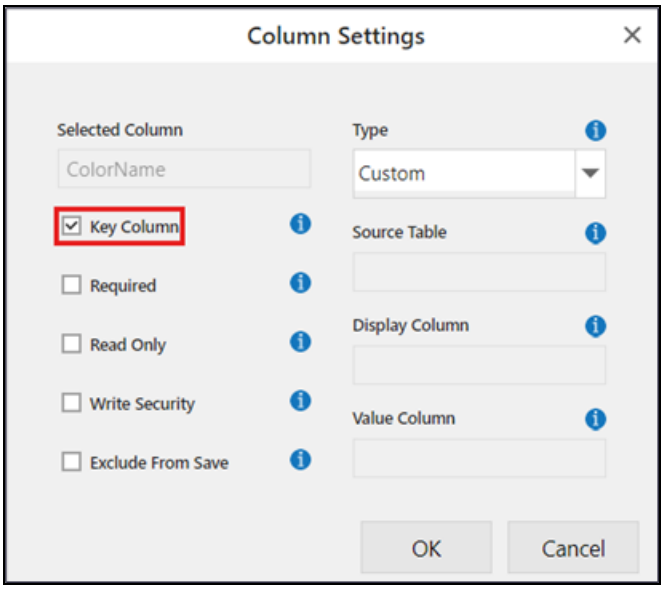
In the case of a Semantic model, the value must be the name of the entity specified in the model.

For SQL-only data sources, the value should match the name of the SQL table utilized with the visual.

When dealing with SQL connections where the table resides in a schema other than “dbo”, certain adjustments are necessary. When importing a table with a specific schema into a report, Power BI automatically attempts to rename the table. For example, if the hr.SalesPerson table is imported, it might appear in the Fields section as "Hr Sales Person" after renaming. This renaming process can pose challenges for Power XL Table in determining the appropriate table for initiating the write-back process. To address this issue, please follow these steps:

1. Rename the table in the Fields list so that it contains exactly the actual SQL table name without the schema (in the example it is SalesPerson)
2. Remove and re-add the columns to the Power XL Table
3. Enter the full qualified table name into the “Source Table” property in the format of schema.table (in our example: hr.SalesPerson).

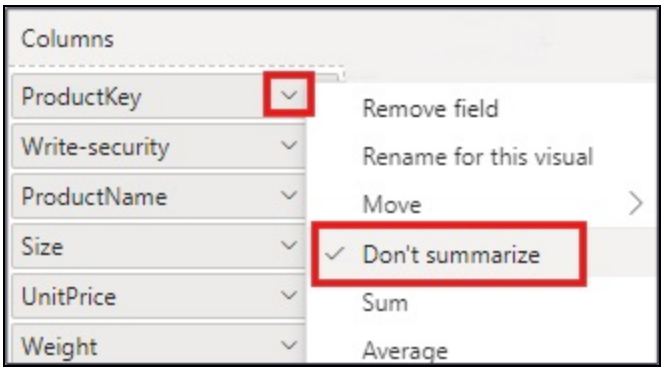
Setting Key Column



On the **Power XL** tab of the designer ribbon, click the **Column Settings** button to cause the above pop-up window to appear. The Key Column can be set up here.

The primary key can be a single column or even multiple ones as a composite key. In the latter case, each column representing the primary key in the table needs to be set up as Key Column.

Numeric fields without aggregation on top of them must be set to **Don't summarize** in the **Visualizations** > **Columns** section by pressing the following icon.



This is particularly important to ensure proper functionality, especially in the case of an auto-increment primary key. Failure to do so will result in the inability to change values in columns with defined aggregations.

On the other hand, aggregated values can still be displayed in the visual, but modifying those inputs will not be possible.

Publishing and Testing the Report

After finishing the report, publish it on either the Power BI Report Server or PowerBI service. Write-back functionality only operates once the report is published, preventing full functionality testing in Power BI Desktop during authoring.

After publishing, attempt to modify a cell value and click on **Save Changes** button. The modifications should then be visible in the report.

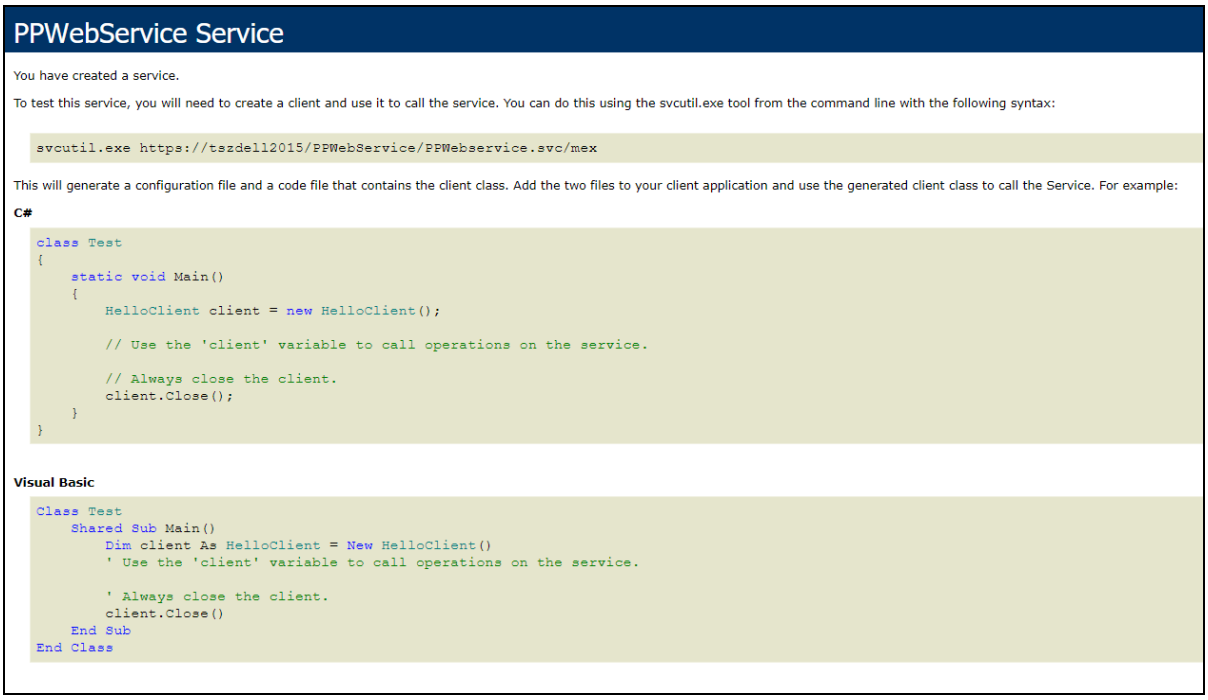
After following these steps, a working, simple, write-back-capable report should be created using the Power XL Table visual.

Validate the Write-back Service

After installation, it's advisable to ensure that the Write-Back Service is installed properly. This can be checked by navigating to the deployed URL, which is in the following format:

http(s)://yourserverName/PPWebService/PPWebService.svc

Seeing the following page in the web browser indicates that the Write-Back Service is up and running.



PPWebService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe https://tszdell2015/PPWebService/PPWebservice.svc/mex
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

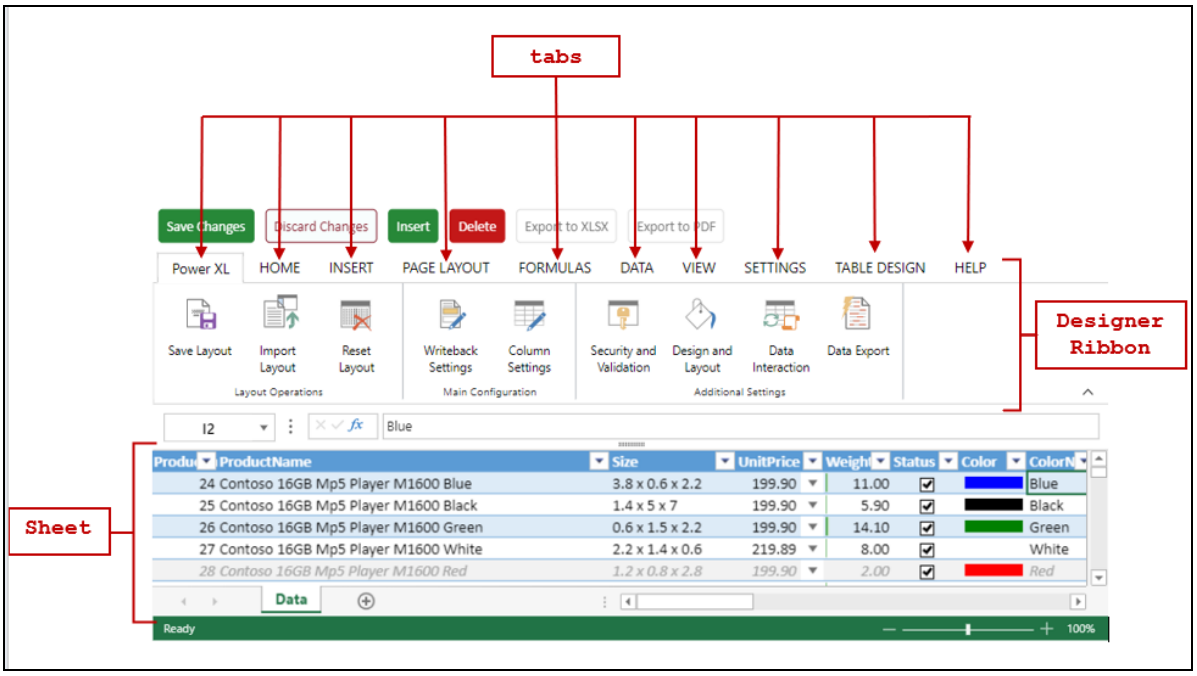
C#

```
class Test
{
    static void Main()
    {
        HelloClient client = new HelloClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
    Shared Sub Main()
        Dim client As HelloClient = New HelloClient()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class
```

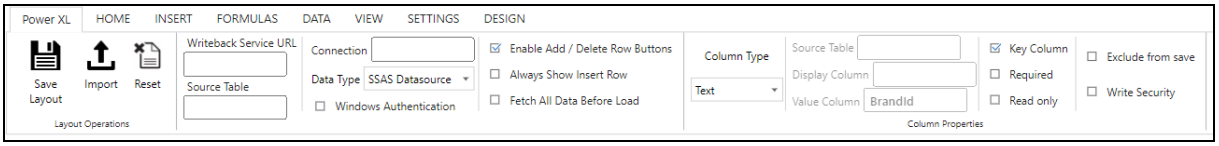
Power XL Table Legend



Designer ribbon can be collapse with ^ symbol in its right bottom corner.

Helper tooltips appear if you hover above the icons:

Power XL Tab



Save Layout

The **Save Layout** button is used to save the configurations and settings that have been configured either on the designer ribbon or within the sheets. It does not directly write the data back to the database, but rather saves the layout preferences for future use.

When the save layout process is initiated, a loading circle in the top right corner serves as a visual indicator that the system is actively working on saving the layout.

Once the saving process is completed, a message “Layout Saved” pops up to inform the user that their configurations and settings have been successfully saved.

This message confirms that the layout changes have been captured and stored for future use.

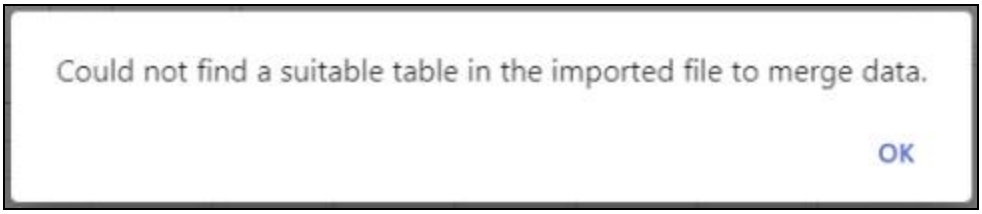
It's important to note that the **Save Layout** functionality focuses on preserving the layout preferences and settings without directly impacting the database. It provides users with the convenience of saving and recalling their customized layouts without affecting the underlying data.

Import Layout

This button can be used to import an Excel file, which may contain multiple sheets with tables and data. If another file is imported on top of the existing one, the new import overwrites the previous import.

If Power XL Table lacks a data table to merge with the Excel data table, the error message below appears:

Importing data underneath the data sheet is not supported.



Importing data underneath the data sheet is not supported.

Reset Layout

Clicking the **Reset Layout** button will delete the layout, remove the content of external cells, and cancel any ongoing imports. However, it does not reset Power BI settings found on the designer ribbon.

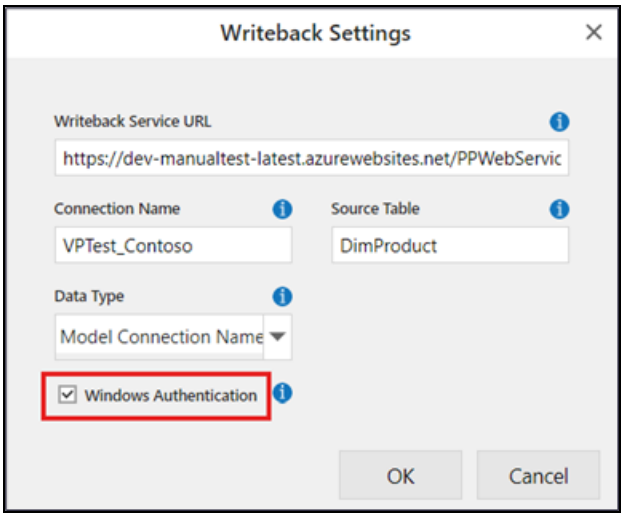
When the **Reset Layout** button is clicked, a confirmation window will appear to verify the reset action.

Writeback Settings

All the settings that are accessible from **Writeback Settings** have already been explained in the Setting Up the Basics section, except for Windows Authentication.

Windows Authentication

To access this setting, click the **Writeback Settings** button on the **Power XL** tab. You can enable/disable it on in the following pop-up window:



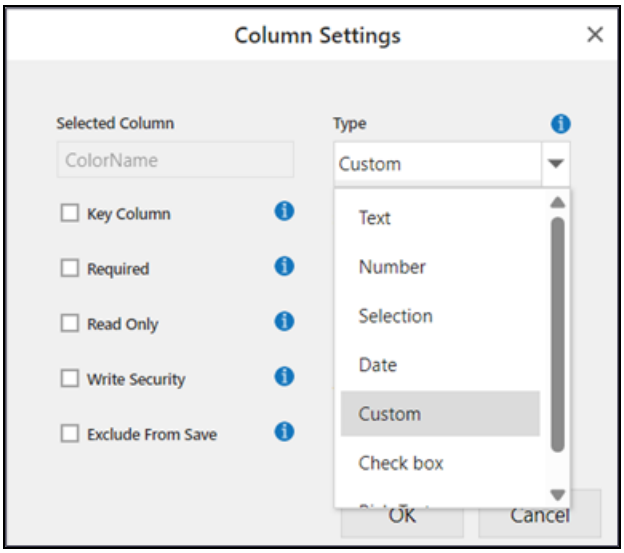
In an On-premises environment using Power BI Report Server, this setting must be turned ON so that the Windows credentials can be properly passed back to the underlying data source. In case the reports are published to the Power BI service using Azure AD, this setting should be turned OFF to utilize AD credentials when accessing the data source.

Enabling Windows Authentication in Gateway will result in the visual posting the Windows login context instead of setting Power BI service credentials within the Write-Back Service request.

This value (e.g.: domain\user instead of username@domain.com) will be set when using USERNAME() in computed and/or default value columns (see later in this document) as well as in SQL context variables. Additionally, this enables impersonation and must be enabled if Windows Authentication is required for authentication in IIS for the Write-Back Service.

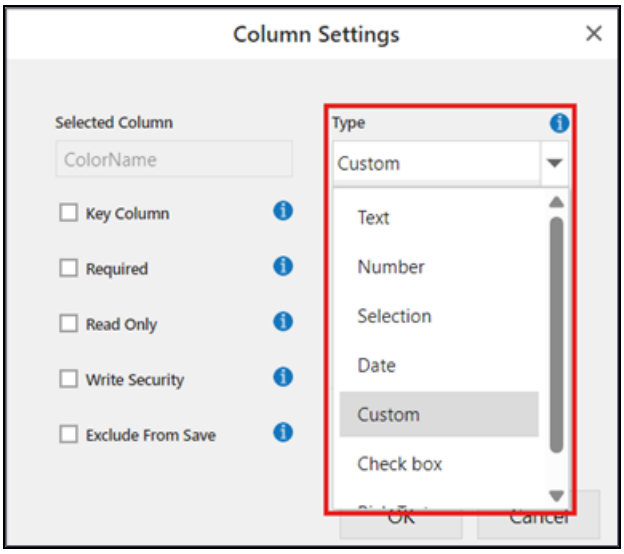
Column Settings

To access these settings, click the **Column Settings** button on the **Power XL** tab. The following pop-up window appears.



Column Type

To access this setting, click the **Column Settings** button on the **Power XL** tab and select the dropdown menu in the highlighted area.



The data types for the columns on the visual do not need to be set because they are automatically recognized by Power BI. However, changes made in this setting will be influenced by how the cell is rendered and how it behaves.

Types can only be set for columns for which aggregation is set to **Don't summarize** in the report field's list window, as only these fields can be saved during write-back. For columns which are not part of the Power BI dataset (local columns, created in the Power XL Table visual), this setting remains hidden.

The following types can be picked:

Text

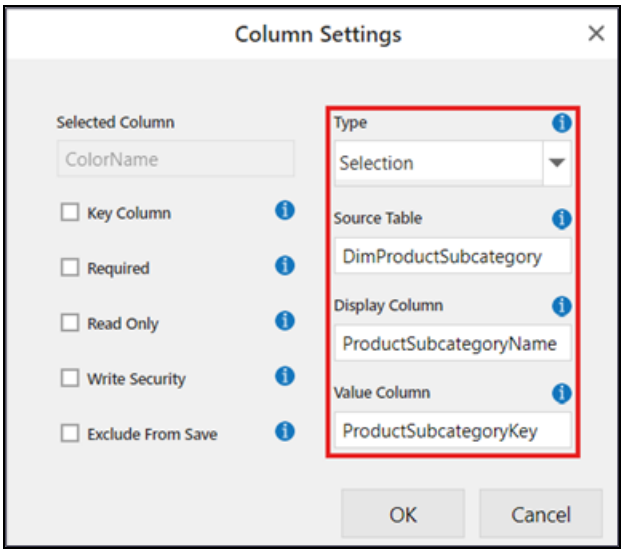
One-line Textbox in the grid.

Number

This type should be used in all numeric data types: decimal, currency, and integers.

Selection

Using this column type, a dropdown control will be rendered for each cell in that column. In order to use this type, additional properties must be set, as the values are coming from another related – or lookup – table.



In this case, the table has a foreign key with a numeric column that is in a relationship with another table. The intention is, instead of showing the numbers, to display their actual text representations and provide the possibility to choose from a set of values.

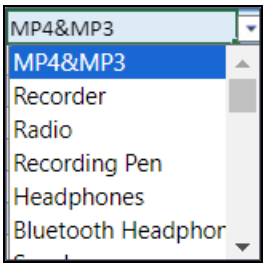
Using the **Selection** type column:

1. Set **Column Type** to **Selection**.
2. The values for the dropdown list are contained in the **Source Table**. The name of the table must be entered as it is displayed in the **Fields** panel. If connected to a SQL database, the name of the SQL table needs to be provided here. If it is in a different schema than “dbo”, that needs to be provided in the following format: schemaName.TableName
3. **Display Column** value will be displayed after the lookup. Usually, this is the text field of the lookup table.
4. **Value Column** is the value that will be matched in the lookup table using the value in the current cell. Usually, this is the key column in the lookup table.

When a source table is provided there can be two ways of accessing the data for the visual. By default, the visual uses the Write-back Service to retrieve the data by querying the underlying data source. With this method, RLS defined in SSAS will not be honored, as all data will be fetched from the source SQL table. In some cases, this extra round trip to the server can be slow. The performance can be increased and any existing RLS can be honored with the following technique:

1. Import SmartFilter Helper visual to the report and add it to the report page. Use the one provided in the setup kit or download it from Power ON Store (store.poweronbi.com).
2. Add those fields (ID and name) to the SmartFilter visual, that are used for the dropdown, **Display Column** and **Value Column**.
3. This method ensures that the available values for the dropdown are stored in the SmartFilter. It will be recognized by Power XL Table, thereby preventing it from querying the Write-back Service.

If the dropdown value list is lengthy and not all values are visible, a scroll bar will automatically appear on the right side.



Additionally, searching among the values by typing the first letter will result in the first value starting with that letter being jumped to and highlighted. For example, when "h" is typed on the keyboard, the "Headphones" value will be automatically jumped to.

Filter on SmartFilter

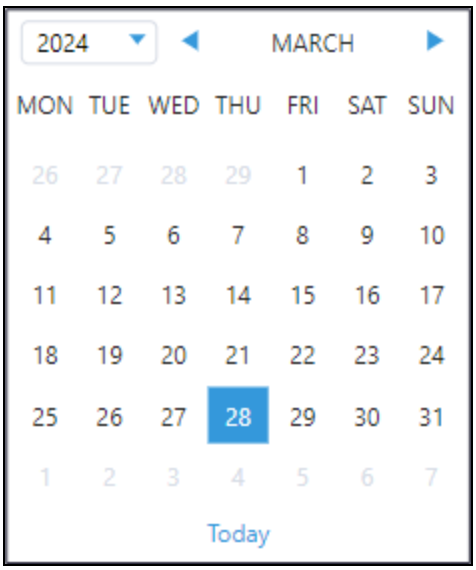
It's possible to restrict the available values in the dropdown list.

After the fields (ID and name) are added to the SmartFilter in the report, a visual level filter can be applied on the SmartFilter to restrict the available values. In this case, not all possible values will be seen in the dropdown for selection purposes; however, the rows in the Power XL Table visual that have other values than the available ones will still be visible.

Date

A data picker is brought up by the **Date** type where the date can be set.

The SQL Database field should be of the type **Date** or **DateTime**.



Custom

Modifying the cell type, validation, or settings of an existing column will result in setting the column type to custom.

This type of column can be deleted by clicking the **Reset** button.

Checkbox

Column types for Boolean values can be set to **Checkbox**.

Rich Text

The content of this type of column can be edited in the Custom Rich Text Editor in both Edit or Reading View mode. The value is displayed in the same way as in TEv2 rich text type.

The editor opens instantly when they are double-clicked or the cell is typed into, AND the content is not (!) read-only.

The editor does not open if the column / cell is read-only or if a write security rule makes the column / cell read-only.

The rich text column content can be saved with the blue floppy icon inside the editor.

Power XL Table can display the rich text type columns created in Table Editor, and vice versa.

Auto resize row height

When the content is changed in a rich text column type cell; the row will resize to show the whole content of the cell. The row height follows the size of the content.

When the visual is loaded, the rows will be displayed according to the content, and the column width will not change.

Visibility of Rich Text Editor option in context menu

The context menu appears when the cell is right-clicked.

The Rich Text Editor option in the context menu is only visible when the selected cell is outside of the data table. In this case, the original Rich Text Editor opens, not the custom Rich Text Editor.

Key Column

To access this setting, click the **Column Settings** button on the **Power XL** tab.

Power XL Table requires a primary key column (a unique value per row) to be defined in the table, which might be an identity column from SQL Server. If the key is not automatically determined by the SQL Server when inserting new records, a value must be entered for it, or some DAX calculations must be used to define a default value.

In **Key Column**, the proper column(s) should be checked as representing the primary key. Composite primary keys can also be used.

Required

To access this setting, click the **Column Settings** button on the **Power XL** tab.

Turning this setting ON makes the column mandatory. If the required field has a blank value, changes will not be saved.

Read Only

To access this setting, click the **Column Settings** button on the **Power XL** tab.

Turning this setting ON prohibits any kind of update on the given column.

After it's turned ON for a column, it will become read-only when the report is saved and exits Edit mode.

If a column originates from a different table than the source, it will be automatically set to read-only.

Write Security

To access this setting, click the **Column Settings** button on the **Power XL** tab.

Modifications on a row can be restricted by creating a calculated column or measure in a model or report. The complexity of the formula is unrestricted, allowing any logical evaluation of the existing columns and filter context to determine if the selected row is editable.

One of the following values must be returned by the write security column:

- To disable modifications, it needs to be returned: 0, "0", null, "null", "NULL", false, "false", "FALSE", or ""
- To allow modifications, any other value needs to be returned, preferably 1 or TRUE.

For example, the following measures can be utilized:

- CanWrite = IF(VALUES('Product'[BrandId]) = 1, "", 1)
- The above measure makes products belonging to the Brand with ID 1 read-only.

For more complex cases, where security control is based on user privileges, the following pattern can be used:

- IsWriteable = MAXX('UserMapping', IF('UserMapping'[UserName] = USERNAME(), "", 1))
- This measure must be added to the Power XL Table, and the write security property for this column must be enabled.

For a detailed example, please refer to the following article:

https://help.insightsoftware.com/s/article/implement-complex-write-security-per-entity

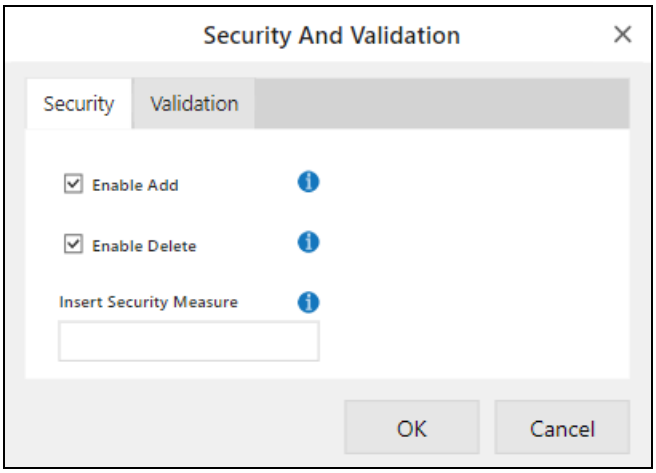
Exclude From Save

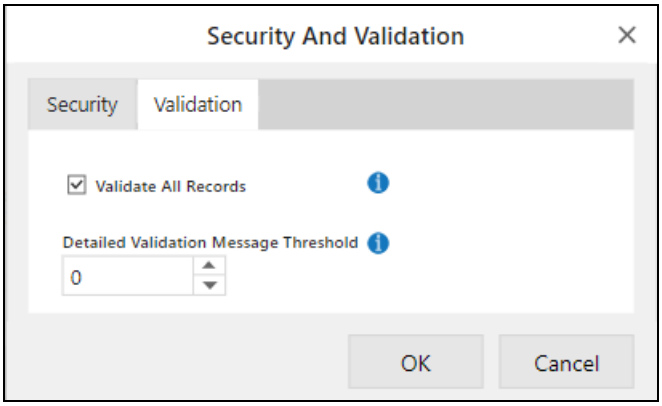
To access this setting, click the **Column Settings** button on the **Power XL** tab.

If enabled, the value will not be retained during the write-back process. The Read Only setting should be enabled when **Exclude From Save** is selected.

Columns originating from a different table than the source, or those that are aggregated, will be automatically set to be excluded from the save.

Security and Validation





Enable Add

To access this setting, click the **Security and Validation** button on the **Power XL** tab.

The **Enable Add** feature allows adding a new row to the data table. After the new row is added, new data can be added to populate the fields within. If the checkbox is not selected, a new row cannot be inserted into the data table using any method in Reading View.

However, it's essential to note that pressing the buttons alone does not automatically save the changes. To finalize the action and ensure that the changes are saved, users must click the **Save Changes** button. This step guarantees that any modifications made, including the insertion or deletion of rows, are committed. It ensures that changes are permanently saved.

Enable Delete

To access this setting, click the **Security and Validation** button on the **Power XL** tab.

The **Enable Delete** option allows users to permanently delete the first row from the data table. If this checkbox is disabled, rows cannot be deleted using any other method in Reading View.

However, it's essential to note that pressing the buttons alone does not automatically save the changes. To finalize the action and ensure the changes are saved, users must click the **Save Changes** button. This step guarantees that any modifications made, including the insertion or deletion of rows, are committed. It ensures that changes are permanently saved.

Insert Security Measure

To access this setting, click the **Security and Validation** button on the **Power XL** tab.

Managing record insertion in Reading View is facilitated by the Insert Security Measure setting.

To utilize this setting, the SmartFilter visual must be added to the report with a DAX measure. Subsequently, in the input field of the setting, the name of the same DAX measure must be written in the following format:

"TableName.FieldName" (e.g., Table.InsertSecurity).

If the SmartFilter is absent or if the name of the DAX measure is misspelled, even if the DAX measure is configured in the setting, row insertion is automatically disabled.

The Insert Security values are as follows:

TRUE - If the value of the DAX measure is not 0, true, or an empty string, then new row(s) cannot be inserted. In such cases, the **Insert** button does not appear on the top row.

FALSE - If the value of the DAX measure is null, 0, false, "false", "FALSE", or an empty string, then new row(s) can be inserted. In such cases, the **Insert** button appears on the top row.

Validate All Records

To access this setting, click the **Security and Validation** button on the **Power XL** tab and select the **Validation** tab.

This setting allows the user to control which records are affected by the Data Validation settings.

When the setting is ON: The Data validation is performed for all records. The **Save Changes** button will not allow saving until all records, including old and unmodified ones, pass the validation. Upon attempting to save and invalid value into the column, if the user clicks **Save Changes** button, an error message will appear.

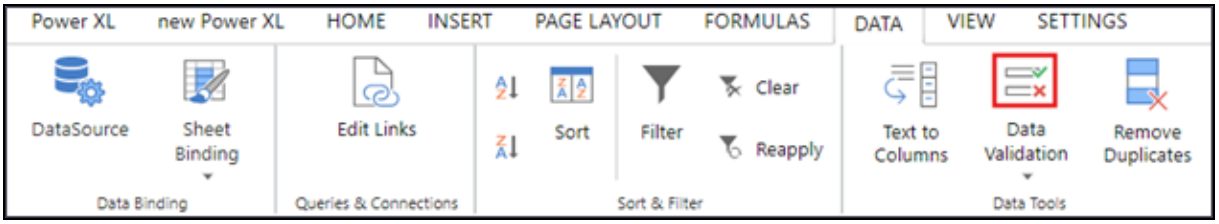
When the setting is OFF:

- The validation is only checked for modified and inserted records. The [Save Changes] button will not allow saving until the modified and inserted records pass the validation.
- Row validation will not be triggered when a modified cell belongs to an inserted custom column.
- New data validation rules can be saved by saving the layout and the report.
- To use the OFF state of **Validate All Records**, at least one column needs to be set as the **Key Column** setting. If no key column is set and **Validate All Records** is OFF, a warning message will appear.
- Upon attempting to save and invalid value into the column, if the user clicks **Save Changes** button, an error message will appear.

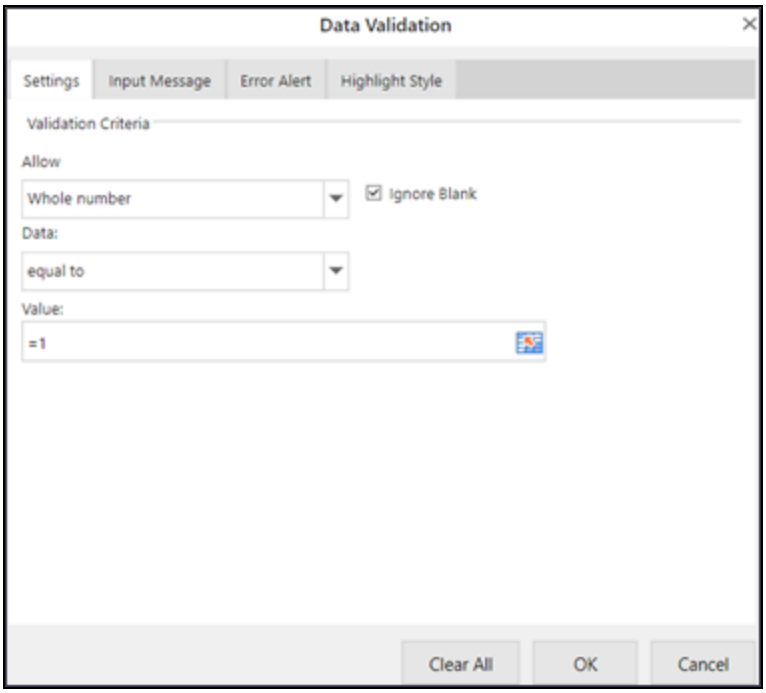
If the setting is switched from OFF to ON, the data validation is not reapplied, so the data validation must be set again.

Data Validation

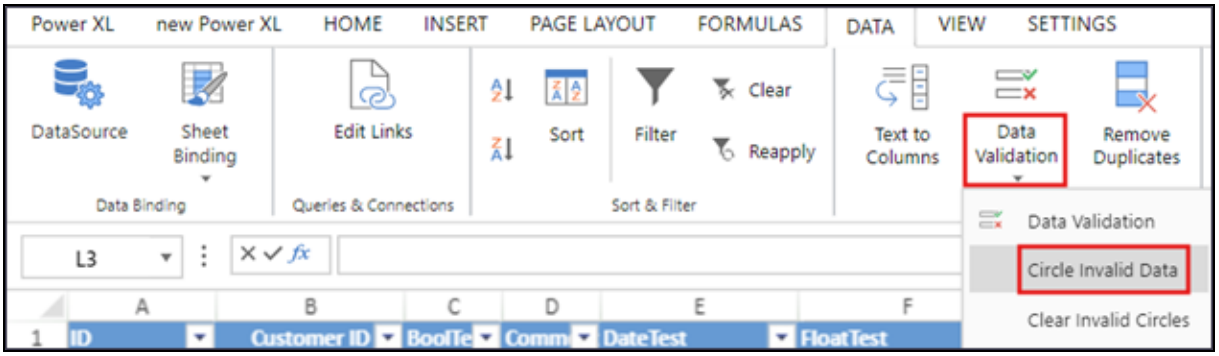
Data validation can be set up for columns in order to ensure that they contain data that agrees to set criteria. Data validation can be set on a column by selecting a cell in the column, navigating to the **Data** tab, and clicking on the following icon.



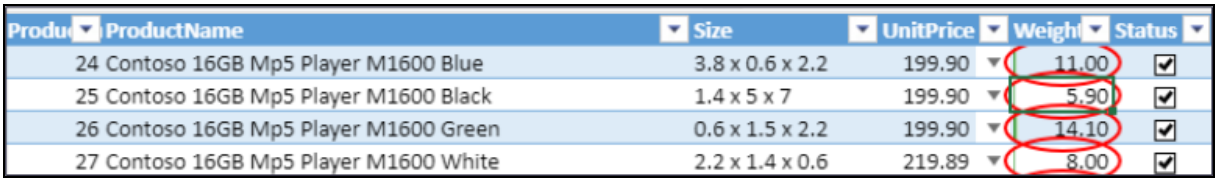
In the following pop-up window, the Validation Criteria can be set up and applied to the selected column by pressing **OK**.



In order to make the invalid data visible, the **Circle Invalid Data** setting should be enabled on the **Data** tab by activating the dropdown menu by clicking on the **Data Validation** setting.



As a result, all invalid data points (whole numbers that do not equal to 1) will be circled.



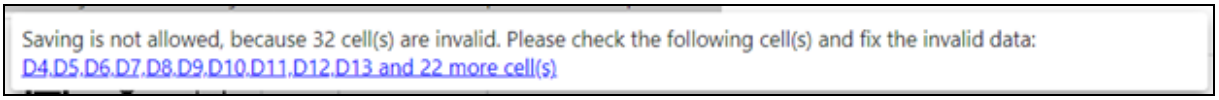
Produc	ProductName	Size	UnitPrice	Weight	Status
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	11.00	☑
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90	☑
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10	☑
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00	☑

Detailed Validation Message Threshold

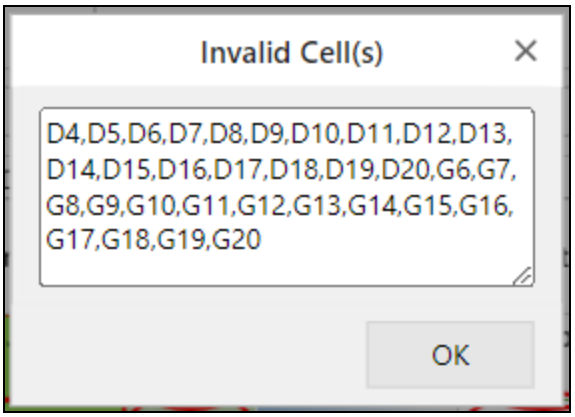
To access this setting, click the **Security and Validation** button on the **Power XL** tab and select the **Validation** tab.

The **Detailed Validation Message Threshold** field allows the specification of how many invalid cell coordinates should be visible in the data validation error message.

For instance, if the threshold value is set to 10, as shown in the following image, the coordinates of the first 10 invalid cells will be displayed in the error message.

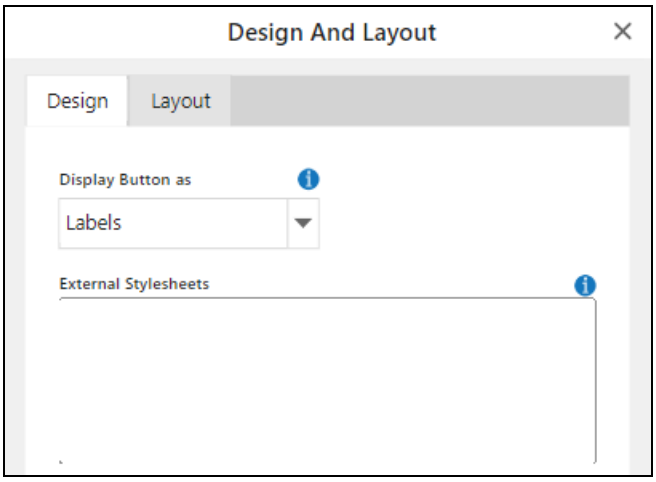


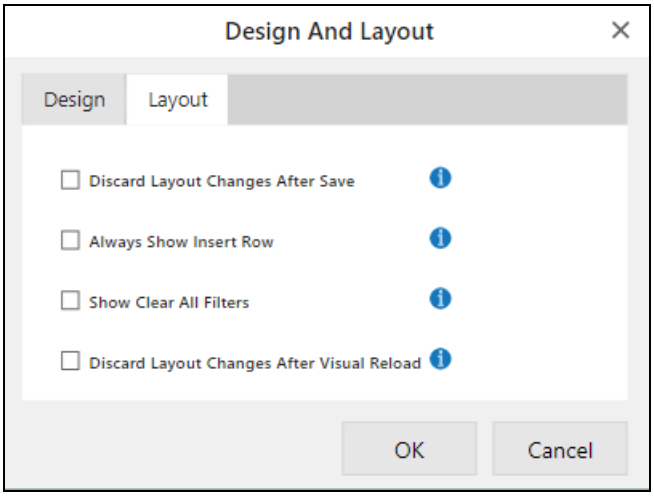
By clicking **more cell(s)** in the message, you can view the coordinates about the additional invalid cells in a new dialog box.



The default value of the setting is 0. If you retain the default value, then the warning message appear as "Some of your cells are invalid".

Design and Layout





Display Buttons as

To access this setting, click the **Design and Layout** button on the **Power XL** tab.

The default value of **Display Buttons as** is **Labels**.

- **Labels** - The top row buttons will be displayed as labeled buttons.

- **Icons** - The top row buttons will be displayed as icons.

External Stylesheet

To access this setting, click the **Design and Layout** button on the **Power XL** tab.

The **External Stylesheet** setting provides an option to inject CSS code to customize any html element, except the canvas, where the sheets are displayed. Users can either supply a CSS file as a URL or simply paste inline CSS code. Both solutions are HTML-safe.

If a wrong URL is provided, an error message appears.

To ensure that formatting from CSS appears correctly, it may be necessary to include the !important command at the end of some CSS properties after adding the value.

For example:

```
example-visual-item {
color: white !important;
background-color: #5500ff !important;
}
```

Discard Layout Changes After Save

To access this setting, click the **Design and Layout** button on the **Power XL** tab, and select the **Layout** tab.

When the setting is ON: After successfully saving to the database, any unsaved layout changes will be discarded, and the layout will revert to its last saved state.

Before successful save:

Produ	ProductName	Size	UnitPrice	Weight
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	11.00
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00

After successful save:

Produ	ProductName	Size	UnitPrice	Weight
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	11.00
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00

When the setting is OFF: After successfully saving to the database, any unsaved layout changes will be retained.

Data table before successful save:

Produ	ProductName	Size	UnitPrice	Weight
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	11.00
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00

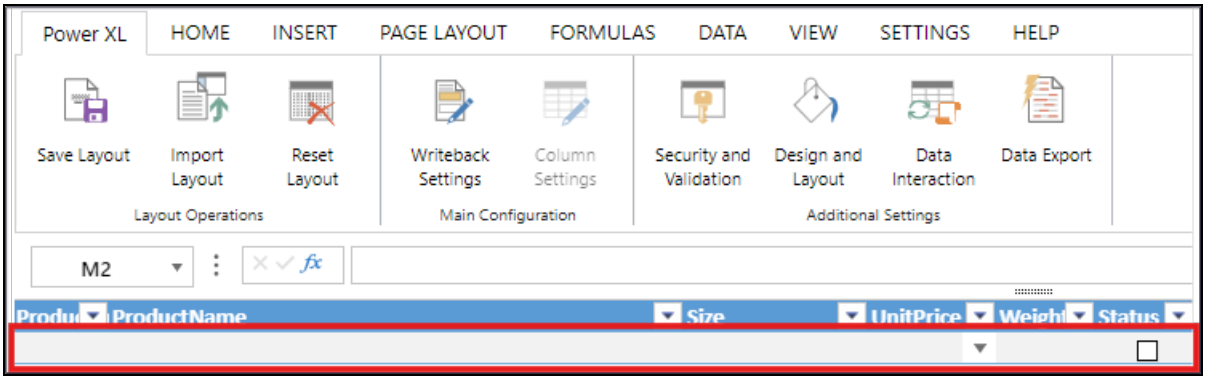
Data table after successful save:

Produ	ProductName	Size	UnitPrice	Weigh
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	12.00
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00

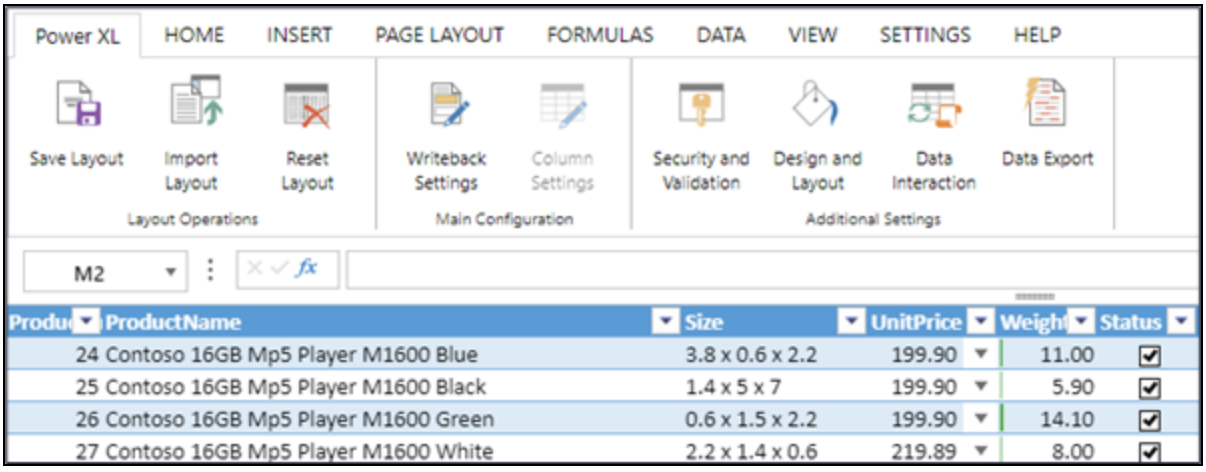
Always Show Insert Row

To access this setting, click the **Design and Layout** button on the **Power XL** tab, and select the **Layout** tab.

When this option is enabled, and if there is no data shown in the Power XL table within the current filter context, a new empty row will be displayed at the top. This empty row allows users to insert new values into the source table:



The row will not be displayed when there are already existing rows to show.



Show Clear All Filters

The **Clear All Filters** button enables the removal of all the filters that have been applied to the column headers in the visual.

To access this setting, click the **Design and Layout** button on the **Power XL** tab, and select the **Layout** tab.

By default, this setting is turned OFF.

The **Clear All Filters** button affects the **Apply Filters to Entire Report** setting. Consequently, using the **Clear All Filters** button removes filters not only from the current visual but also from other visuals in the report.

The **Clear All Filter** icon is shown in the following image.

Clear All Filter button is shown in the following image.

Discard Layout Changes After Visual Reload

To access this setting, click the **Design and Layout** button on the **Power XL** tab, and select the **Layout** tab.

The **Discard Layout Changes After Visual Reload** setting allows specifying a preference for saving data.

By default, this setting is OFF.

When turned ON, it affects how the visual handles unsaved layout modifications after specific events occur on the page.

The visual reloads and discards unsaved layout changes if any of the following actions occur:

- **Slicer Change**: If a slicer is modified or adjusted.
- **Filter Application**: When a filter is applied to the visual.
- **Cross Filtering**: If cross-filtering occurs with a different visual.
- **Value Saving**: This behaves similarly to the Discard Layout Changes After Save setting.

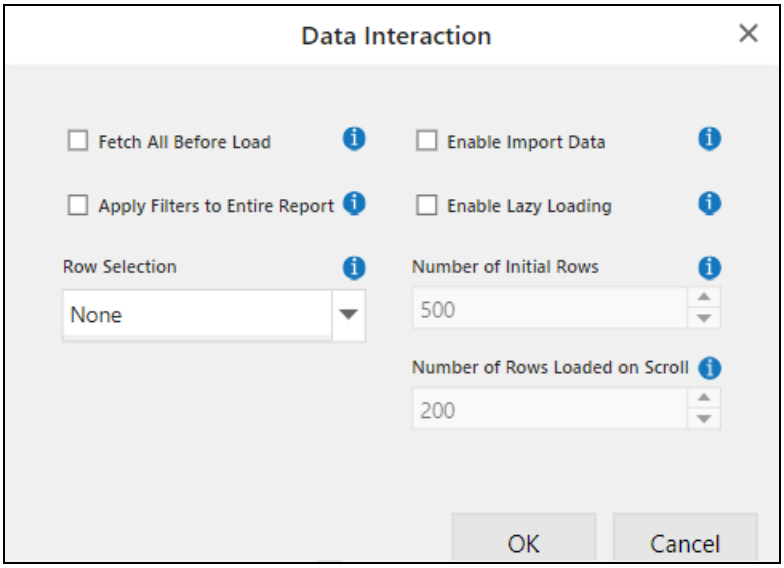
Data table before Visual Reload:

Produ	ProductName	Size	UnitPrice	Weigh
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	11.00
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00

Data table after Visual Reload:

Produ	ProductName	Size	UnitPrice	Weigh
24	Contoso 16GB Mp5 Player M1600 Blue	3.8 x 0.6 x 2.2	199.90	11.00
25	Contoso 16GB Mp5 Player M1600 Black	1.4 x 5 x 7	199.90	5.90
26	Contoso 16GB Mp5 Player M1600 Green	0.6 x 1.5 x 2.2	199.90	14.10
27	Contoso 16GB Mp5 Player M1600 White	2.2 x 1.4 x 0.6	219.89	8.00

Data Interaction



Power BI, as a default, can only render 30000 records. However, in case of more sizeable datasets, we need a way to bypass this limitation. Power XL Table has two settings that allow the user to control the amount of records that the visual renders:

- **Fetch All Before Load**
- **Enable Lazy Loading**

Fetch All Before Load

To access this setting, click the **Data Interaction** button on the **Power XL** tab.

By default, this setting is turned OFF.

- OFF: The Power XL Table visual will load maximum 30.000 rows from the source table.
- ON: The Power XL Table visual can load all data from the source table, even it has more than 30.000 rows. This may cause a longer loading time for the visual.

Enable Lazy Load

To access this setting, click the **Data Interaction** button on the **Power XL** tab.

The setting allows the Power XL Table visual to render records incrementally by specifying the size of the initially loaded segment and the size of the subsequently loaded segments. This setting bypasses the inherent limitation of Power Bi only being able to load 30000 records at most.

Enabling Lazy Loading will automatically turn off and disable **Fetch All Before Load** and enable the following settings:

- **Number of Initial Rows**
  - The value of this setting must be between 200 and 30000. Trying to apply a value outside this interval will cause the value to return to either 200 or 30000.
  - Upon loading a report, the visual will render a number of records equal to the value of this setting.

- **Number of Rows Loaded on Scroll**
  - The value of this setting must be between 200 and 30000. Trying to apply a value outside this interval will cause the value to return to either 200 or 30000.
  - Upon scrolling to the end of the initial segment, the visual will load a number of records equal to the value of this setting.

Apply Filters to Entire Report

To access this setting, click the **Data Interaction** button on the **Power XL** tab.

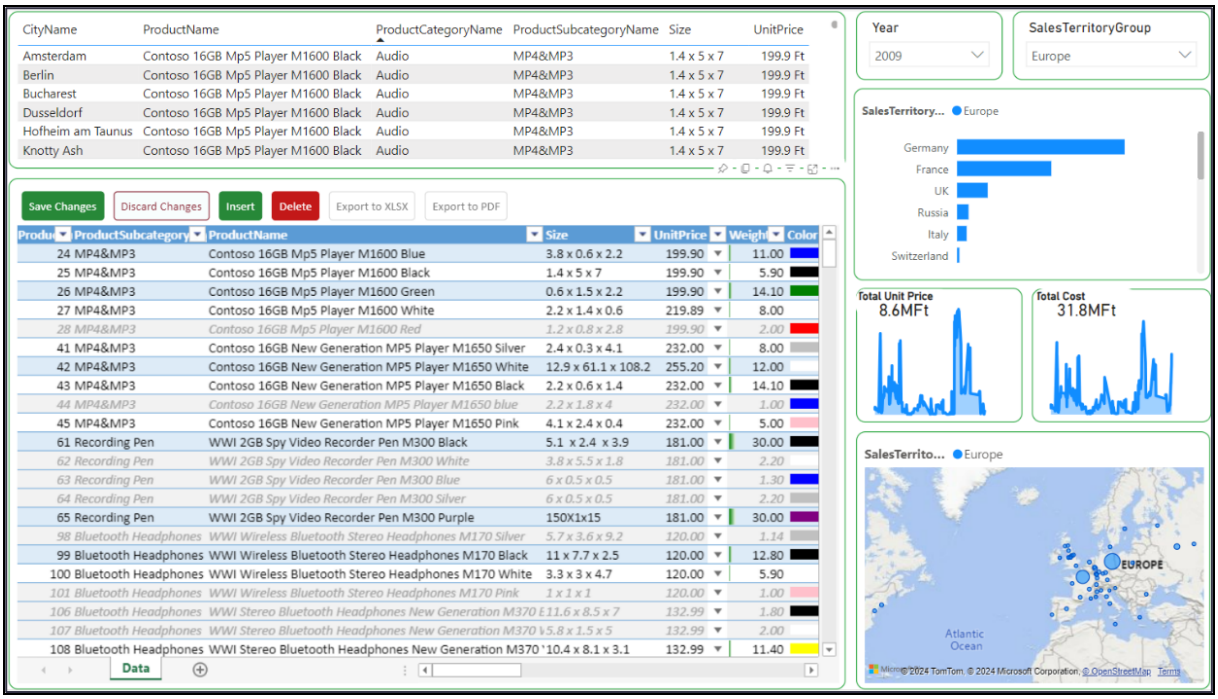
This setting allows the column filtering of the visual to be applied to all the other visuals on the same report page.

- When the setting is OFF, only the visual associated with that column will be filtered. The filter will not affect other visuals in the report. Additionally, the filters will not be saved into the layout. This means they will not persist when the report is reopened or reloaded.

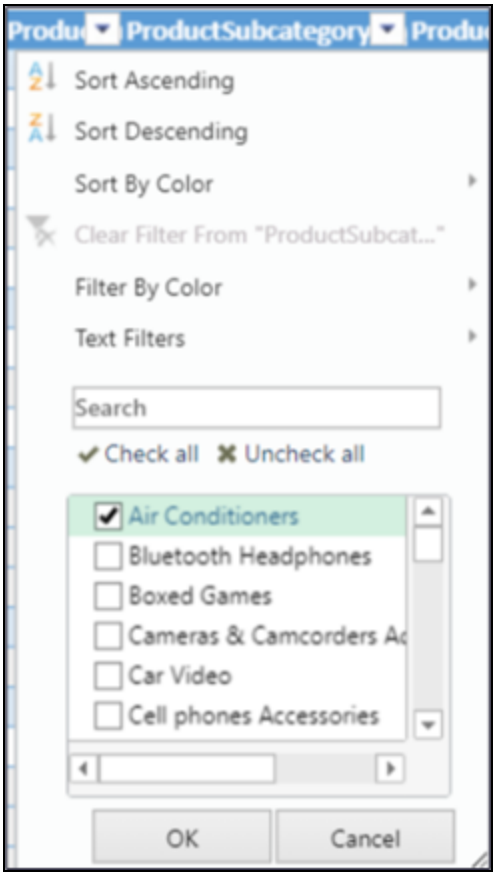
- When the setting is ON, if a filter is applied to a column header, the filter effect will be applied to the entire report, affecting all visuals.

The setting is demonstrated in the use case below.

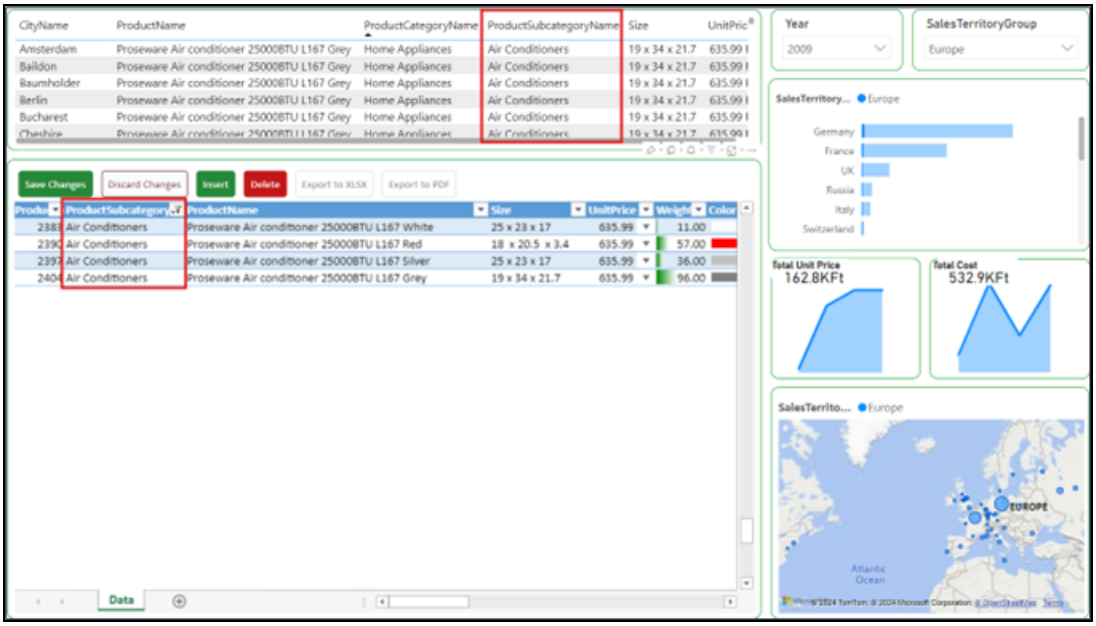
Before applying a Column Filter all records are visible in all the visuals of the report:



A filter is applied on the Product Subcategory column:



Upon pressing the **OK** button the filter is applied on the PXLT visual and all the other visuals of the report as well:



Row Selection

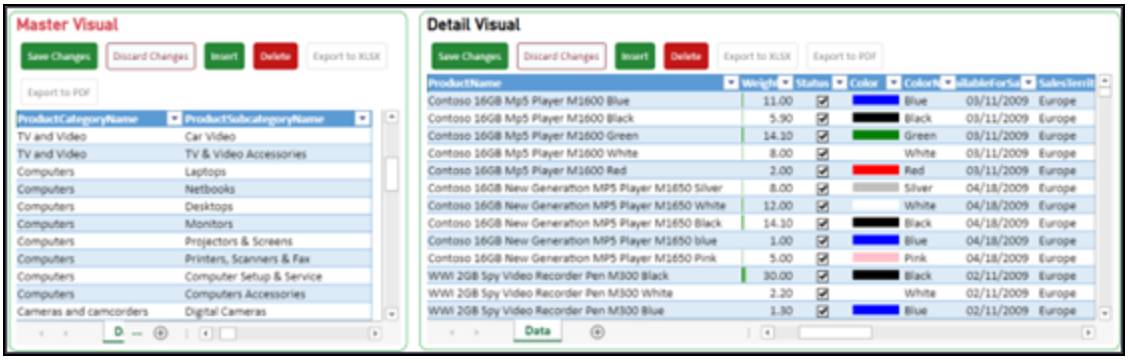
To access this setting, click the **Data Interaction** button on the **Power XL** tab.

Row Selection allows filtering records through a Master visual. By selecting a row in the Master visual, the Detail visual will re-render with the records that fit the ID of the rows selected in the Master visual.

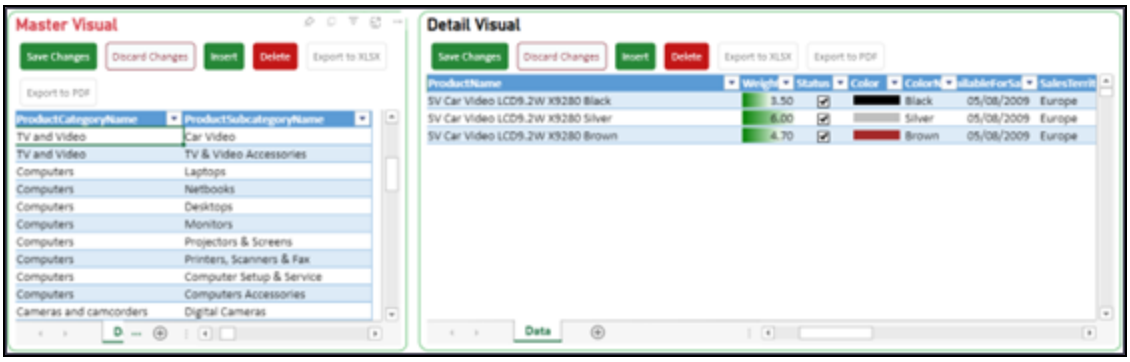
In the following example, the master entity is the ProductSubcategory table and the detail entity is the Product table. The Product table has a many-to-one relationship with the ProductSubcategory table.

In order for this to work, the “Row Selection” setting must be set up in the Master visual.

Before Row Selection:



After Row Selection:



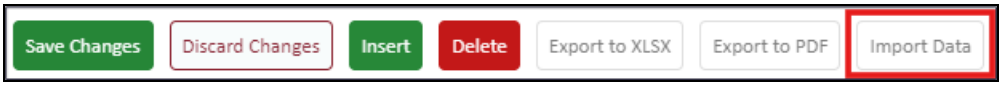
The **Row Selection** dropdown box has the following settings:

- **NONE** - Disables Row Selection. The Detail visual will not re-render upon selecting rows in the Master Visual.
- **CELLS** - Allows selecting a single cell or a range of cells to trigger Row Selection.
- **ENTIRE ROW** - Row Selection is triggered only when one or more row headers are selected in the Master visual.
- **BOTH** - Row Selection is triggered if one or more cell(s) or entire row(s) are selected in the Master visual.

Enable Import Data

To access this setting, click the **Data Interaction** button on the **Power XL** tab.

Upon enabling the setting and clicking **OK**, the **Import Data** button appears in the menu.



The **Import Data** button allows importing data using the following file extensions:

- xlsx
- csv (comma- or semicolon-separated)
- tsv

Importing Data

To import the data, follow these steps:

1. Click the **Import Data** button. The **Upload File** popup dailog window appears.

2. Drag and drop the file into **Upload** window to upload the file. Alternatively, you can also search the file in your browser that you want to upload.
3. Click the **Import** button to initiate the import process. The **Import Finished** message appears if the data import is successful.

If the file extension is not supported, an error message appears.

Import File Requirements

Before you import a file, ensure the file adheres to the following conditions::

- Import files should consist of a single sheet only.
- Data in the file must start from cell A1.
- Include column headers in Row 1, matching the data table's columns.
- Formatting won't be imported with the data.
- Optionally, it can contain an operation column with keywords like "add," "upd," or "del" for adding, updating, and deleting data. The operation column is not mandatory
- Remember, using the operation column is not mandatory.
- If no operation column exists, the backend checks for existing records by ID.
  - If found, it is treated as an update.
  - If not found, the backend attempts to add the record.

Merging Data Examples

When merging data, consider the following scenarios:

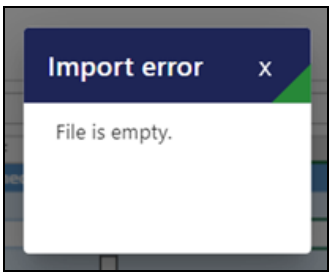
- If the imported Excel file contains two columns, but the data table in the visual has only one column, the extra column from the imported file will not be displayed. Only the column that matches the data table will appear. Example: If the Excel file has columns A and B, but the visual's data table has only column A, only column A data will be shown.

- When the visual's data table has two columns, but the imported Excel file contains only one column, an error occurs during the import process. Ensure that all columns from the source table are included in the import file.
- If there is no common column between the visual's data table and the XLSX file, the import is considered invalid. In such cases, the backend will generate an error message: "No matching columns found."

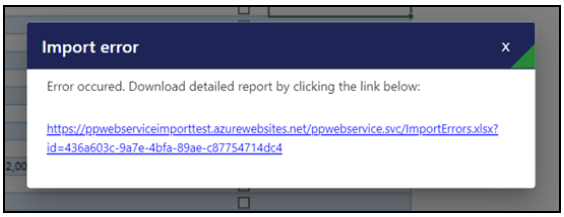
Error Handling

When an error occurs during the data import process, the following scenarios are handled:

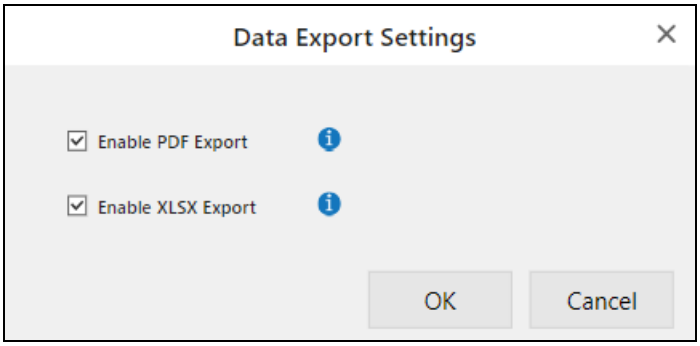
- If a fatal error occurs, a pop-up displays the corresponding error message. In such cases, the visual will not import the file.



- When some records can be successfully imported, but there are invalid records in the imported file, users have an option.
  - They can download a report file that highlights the rows with errors.
  - The report file allows users to correct the problematic rows and import again.



Data Export Settings

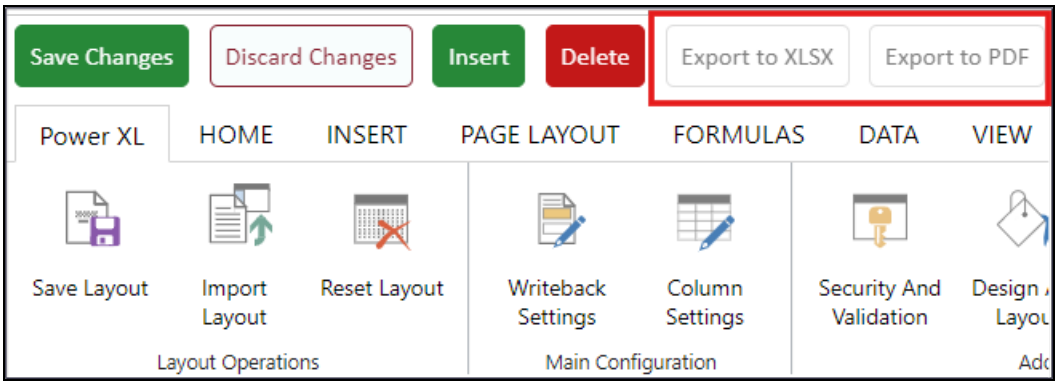


Enable XLSX Export / Enable PDF Export

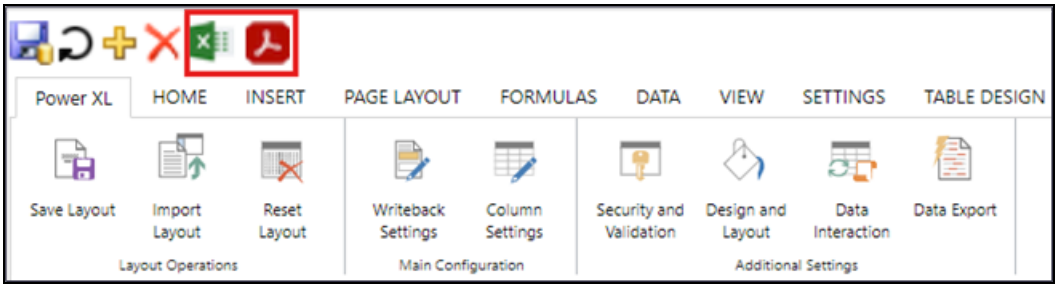
To access these settings, click the **Data Export Settings** button on the **Power XL** tab.

When these settings are turned ON, the **Export to XLSX** and **Export to PDF** buttons will be displayed in the top row, positioned above the Designer Ribbon.

If operation buttons are displayed as Labels:



If operation buttons are displayed as Icons:



Debug

Diagnostic Mode

Enabling this setting will reveal debug settings.

Other features

Power XL Specific Formulas

USERNAME()

Returns the whole name of the current user (First name + Last name)

DOMAIN()

Returns the domain of the current user (e.g.: poweronbi.com)

USERPRINCIPALNAME()

Returns the whole email address of the current user.

IFISROWDIRTY()

Evaluates and returns its parameter when any field of the current row has been changed by the user.

IFISRANGEDIRTY()

Evaluates and returns its first parameter when any field within the specified range (as the second parameter) has been changed by the user.

CUBE.FILTER(TableName, ColumnName)

This formula is typically used for master-detail tables or in conjunction with SmartFilter.

A typical usage example is: =IFISROWDIRTY(CUBE.FILTER("SalesTest", "Customer ID"))

The CUBE.FILTER function retrieves all SmartFilter values associated with the provided Table and returns the value if there is only one unique value for the specified Field. However, if there are multiple values for the Field, the function returns null.

CUBE.FILTER supports column types such as Text, Number, Date, and Selection.

Note: The formula is not editable in Reading view for selection column types.

Implementation of Computed / Default Values

To implement the same functionality in Power XL Table as in the TableEditor visual, you can utilize the IFISROWDIRTY() formula. This formula allows you to set computed or default values based on whether a row has been modified by the user.

For example, let's say the Power XL Table has an audit trigger on the database side that requires capturing the current user's email address (modified by) and the current date and time (modified when).

To achieve this, follow these steps:

1. Select the first row of the "Modified By" column and use the following formula:

=IFISROWDIRTY(USERPRINCIPALNAME())

This formula will populate the "Modified By" column with the email address of the user whenever they make changes to a row.

2. Select the first row of the "Modified When" column and use the following formula:

=IFISROWDIRTY(NOW())

This formula will update the "Modified When" column with the current date and time whenever a user modifies a row.

The Power XL Table visual will automatically copy these formulas to all other rows. Once the configuration is complete, whenever a user makes any modification to a row, the "Modified By" column

will be updated with their email address, and the "Modified When" column will be updated with the current date and time.

It's worth noting that not only can you use these functions as parameters, but you can also incorporate other formulas or calculations as needed.

! Please keep in mind, that when we support our clients and partners we are focusing on our product and not on Excel functionalities.

Power XL Table Troubleshooting Guide

This topic explains how to diagnose and fix some of the problems that may occur when working with the Power XL Table visual.

> **Note:** To review all cases, log in to https://help.insightsoftware.com/ and access the Power ON Knowledge Base articles. For further assistance, you can also submit a ticket through this support portal.

Network Error

Symptom: A network error message is displayed when you try to save data.

Cause: Listed below are some of the possible reasons for this error:

- The Write-Back Service URL is not set correctly – or it is malformed - in the Data Entry settings
- The Write-Back Service is not reachable or off-line
- Bad connection name or/and type specified
- License service is stopped unexpectedly
- If it is a on-premises or Gateway installation, the Windows Authentication setting is available by pressing the **Writeback Settings** button in the **Power XL** tab.
- Missing files in the web service folder, or typo / malformed strings in the web.config file
- Windows / Kerberos authentication issue
- Missing Service Principal Names
- Missing active directory permissions for service accounts
- Report server URL web service URL format mismatch
- Invalid SSL certificate

Solution: Ensure that the Write-Back Service is operational and accessible, without interference from the firewall. Confirm the existence of the referenced connection in the Write-Back Service configuration, and ensure that the right connection type is selected.

- To understand the license service failure type, refer to the following article:
  - https://help.insightsoftware.com/s/article/error-the-communication-object-system-servicemodel-channels-servicechannel-cannot-be-used-for-communication-because-it-is-in-the-faulted-state

- If you encounter the CORS issue, check the Write-Back Service's web.config file for typos. Also check for missing DLL-s and config files inside the web service folder. If the issue is still not resolved. update the web service file following the guidelines provided in the following articles:
  - https://help.insightsoftware.com/s/article/how-to-update
  - https://help.insightsoftware.com/s/article/how-to-update-the-service-manually-azure-cloud

- If an SSL error occurs, ensure that the certificate is issued by a trusted authority for the fully qualified domain name of the IIS server. Alternatively, confirm that the client's browser is set to ignore the certificate if it is self-signed. If the certificate is internally issued by your organization and you are attempting to access the report outside of the organization's domain, take necessary precautions.

- If utilizing Reporting Services On-premises, ensure that the report server URL and the web service URL formats align. Both URLs should either reference the machine name or the fully qualified domain name, and both should be either HTTP or HTTPS. Additionally, if Windows Authentication is configured in the IIS settings, ensure that Windows Authentication is enabled in your visual under the [Data Entry] group.

- For an On-premises installation within a domain utilizing Windows Authentication, ensure that Service Principal Names (SPNs) are created for your SQL and SSAS servers, and that the correct domain users are assigned to the services. An SPN is also required for the service account responsible for running the PPWebService on the IIS machine as the Application Pool user. Verify that delegation is enabled in your Active Directory from the App Pool's user (which must be a trusted user) to the SQL/SSAS services. Incorrect configurations in these areas can lead to Kerberos authentication issues, potentially resulting in HTTP 403 or 404 errors, or prompting login popups. Refer to the following article and liaise with your internal IT team for further assistance:
  - https://help.insightsoftware.com/s/article/configure-iis-for-kerberos-authentication

The Visual is not Working in Power BI Desktop or the Settings are not Shown

Symptom: The visual fails to render, or Power XL-specific settings are not displayed.

Cause: The following are the two potential reasons for this error:

- Your system might be running out of memory, resulting in Power BI Desktop's inability to render elements correctly.

- Your Power BI Desktop cache is outdated.

Solution: To resolve these issues, free up memory on your system by closing other applications. For instructions on clearing the Power BI Desktop cache, refer to the Clear Power BI Desktop cache article available in the Knowledge Base.

Save Failed

Symptom: You encounter a Save Failed message when attempting to write-back to the selected cell.

Cause: This error may occur due to a configuration error in the visual, improper configuration of the Write-Back Service, or due to one of the following reasons:

- A custom validation implemented that prohibits write-back
- SQL objects are interfering with the data modification TSQL statements (like security policies, triggers, unique constraints, etc.)
- The service account used by the Write-Back Service does not have permission on the underlying SQL database to make the necessary modification on the source table

Solution:

- Check if the service account has the necessary permissions, the password has not expired
- Verify that RLS policies or triggers are not prohibiting the operations