insightsoftware

Power ON Visual Documentation and Troubleshooting Guide

Power XL Table

Version 25

December 2025

Introduction to Power XL Table v25.4

Power ON's Power XL Table (PXLT) is a custom visual developed for Microsoft Power BI. It enhances the user experience by enabling end-users to make permanent changes to data (write-back), enriched with text input controls commonly seen on modern HTML pages (example, date pickers, dropdowns, rich text) alongside the traditional look and feel.

The Power XL Table product documentation provides a detailed overview of the visual and how to use it in reports. It covers common use-case implementations, proper configuration, and strategies for overcoming potential pitfalls. This documentation is intended for both technicians (developers, DBAs, BI professionals) familiar with SQL Server, SSAS Tabular models, Azure Services, Excel, and Power BI, as well as users who primarily focus on building and preparing reports using Excel or Power BI Desktop. Some sections focus on more technical subjects.

Power XL Table allows users to edit any type of row-level information in Power BI. The most typical use case is Master Data Management (example, customers, products, materials), including adding or removing records, editing existing members, changing attributes, and performing simple data entry tasks. Users can consider this visual as a replacement for a 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, 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/false data type columns
- Dropdown list (called Selection) control
- Supports copy and paste of 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 more

Following is the high-level overview of how write-back works for Tabular models. Depending on your data source (SSAS In-Memory, SSAS Direct Query, or SQL only) Power ON's write-back service performs the following steps:

1. The service captures the modified value along with its tuple (the intersection of dimensions used in the calculation of measure for the given cell) and the user context. Based on the SSAS model structure (relationships, table queries, and measure definitions) the service composes T-SQL statements for execution.
2. It executes the compiled T-SQL statement against the underlying data source (fact table) to save the modifications.
3. For SSAS in-memory models, the service reprocesses the table. See Performance Optimization for

more information.

4. It refreshes the visual to display the changes in the report.

Important: The target table for the write-back must have a primary key defined. Note the following:

- Composite keys are supported.
- For Semantic data sources, the primary key must be part of the Semantic model, visible to end users, and added to the visual as a field.
- If there are specific business rules for creating a key for a new record, you must either create a custom trigger in your database to provide the new value during the operation, or if the key can be computed based on existing data visible in the model using DAX, the Computed Column property can be utilized.

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.

Power XL Table Prerequisites

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

Write-Back Service

To use the write-back capabilities in Power BI, ensure that a working and configured Write-Back Service (PPWebService) is installed in your environment. If the Write-Back Service is installed and configured, follow these steps to build a report:

- Import the Power XL Table and VPService pbiviz files into the Power BI Desktop instance. This must be done for each report that intends to use PXLT.
- The visuals are stored within the report itself, enabling write-back capabilities when opened. For Power BI cloud services, you can store the custom visual in a centralized repository for easier management. For more information, refer to https://docs.microsoft.com/en-us/power-bi/developer/visuals/power-bi-custom-visuals-organization.
- For more details on configuring the PPWebService, log in to https://help.insightsoftware.com/ and refer to the Power ON Knowledge Base articles.
- In Azure, 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, pointing to:
  - Power BI dataset or SSAS model: The used SSAS Cube.
  - SQL-only models: The used SQL database.

  The web.config file is located 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. Valid connection string examples for different data sources can be found at https://www.connectionstrings.com/.
- The PPWebService service account used in the connection string must have the following permissions:
  - SSAS models: Administration rights on the cube, plus data reader and data write roles on the underlying data source database of the cube.
  - 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:
  - SSAS models: Data reader membership.
  - SQL-only models: Data reader role.

  If impersonation is enabled, end users must have the 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 Configuration

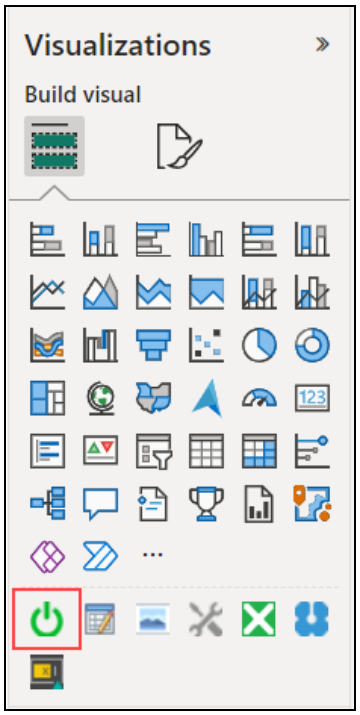
This topic outlines the steps to set up the Power XL Table visual to create a report.

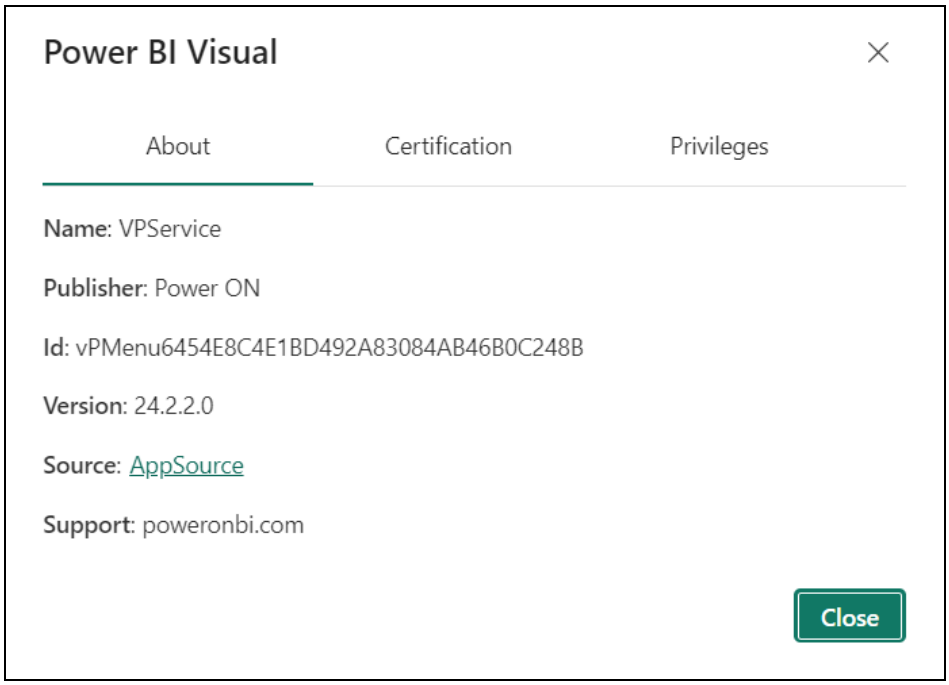
Set Up the Basics

When 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. This section outlines the process of configuring the VPService visual and setting up the connection between the visuals and the Write-Back Service.

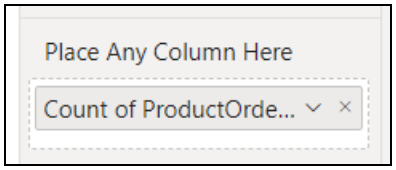
Configure the Visual

1. After importing VPService, the following icon appears in the **Visualizations** tab. Right-click the icon and select About to review the version information.



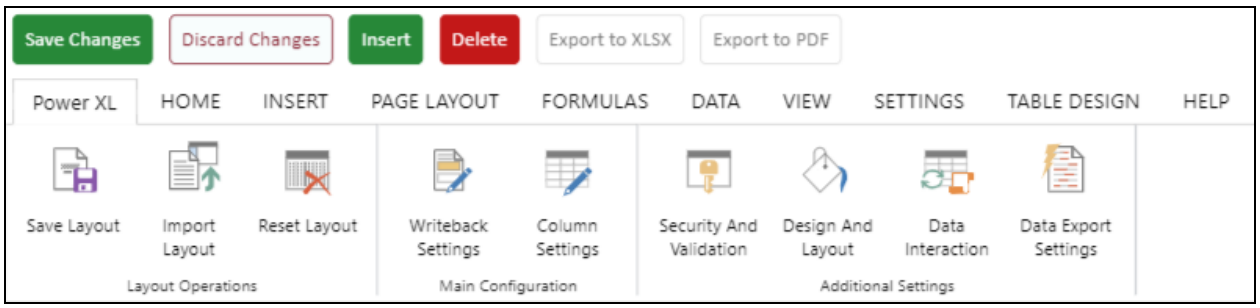


2. Click the icon to add the VPService visual to the report.
3. Add a field to the VPService visual. Drag and drop the field from the **Data** tab to the **Place Any Column Here** section of the **Visualizations** tab.

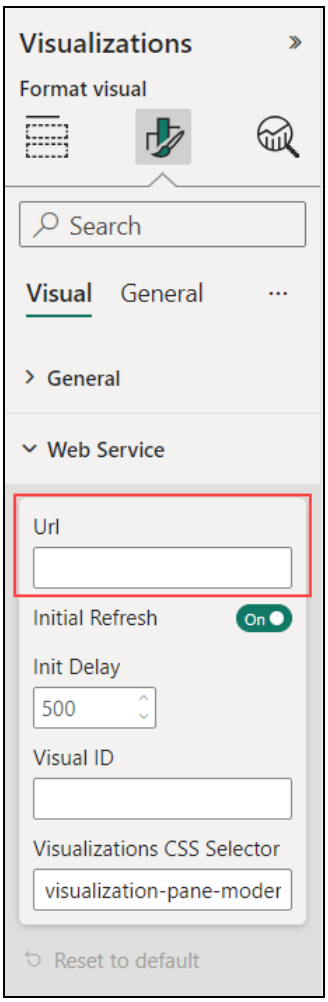


Important: If you have a mixed report where the source table is connected through a direct query in your model, and you also have additional tables in import mode, it is recommended to pull the arbitrary field into the VPService visual from one of the import mode tables for better performance.

4. Click the icon to add the Power XL Table visual to the report.
5. Add a field to the Power XL Table visual by dragging and dropping it under **Columns**. This will the display designer ribbon which includes the configuration options for the visual.

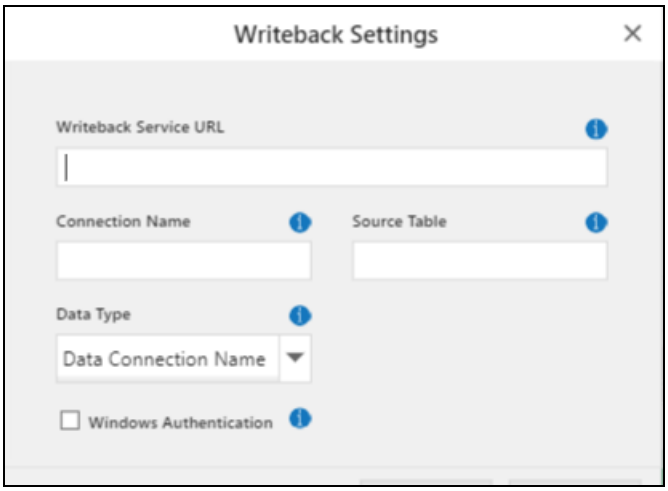


6. Configure the Web Service URL of the VPService visual and the Writeback Service URL of the Power XL Table visual. These URLs must be identical.
   a. For the VPService visual, select the **Format your visual** icon in the **Visualizations** tab and then enter the URL under **Web Service**.



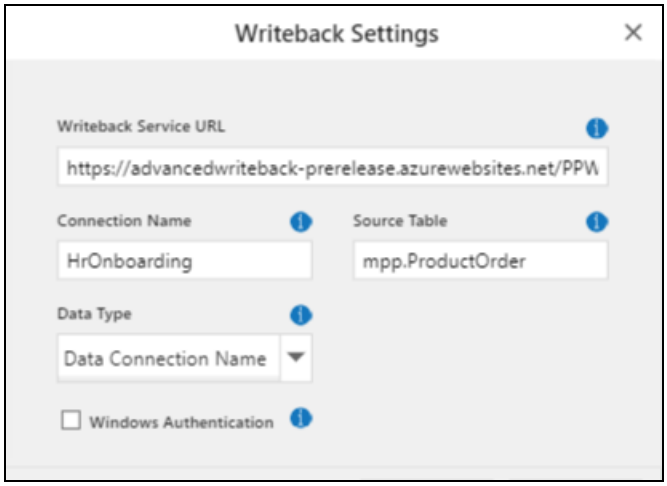
   b. For the Power XL Table visual, select **Writeback Settings** in the designer ribbon and then

enter the Writeback Service URL.



Configure the Required Writeback Settings

1. Select **Writeback Settings** in the **Power XL** tab of the designer ribbon.



2. Configure the following settings:
    a. **Writeback Service URL** - Specify the URL that points to the hosting machine or the app service where it was installed. Typical URL: http(s)://WEBSERVICE_COMPUTER_NAME/PPWebservice/PPWebservice.svc. Ensure there is no extra slash at the end of the URL.
    b. **Connection -** Specify the name of the connection string defined in the web.config file of the Webservice, enter the connection string directly, or leave it blank, depending on the **Data Type** setting. The maximum length of this field is 250 characters.
    c. **Data Type** - Select the connection type used by the Write-Back Service.

i. **Data Connection Name** (formerly SQL Datasource) - The connection is configured in the PPWebService web.config file. It should be referenced by its name in the **Connection** field.

ii. **Model Connection Name** (formerly SSAS Datasource) - Specify the connection name in the **Connection** field within the web.config of the web service.

iii. **Data Connection String** (formerly SQL) - Directly set the connection string in the **Connection** field. If the default SQLConnection is defined in the web.config, leave this field blank to automatically use that connection. This option is primarily used for testing and development purposes.

iv. **Model Connection String** (formerly SSAS) - Directly set the connection string in the **Connection** field. If the default SSASConnection is defined in the web.config of the web service, leave this field blank to automatically use that connection. This option is primarily used for testing and development purposes.

d. **Source Table** - Specify the name of the target table for the Write-Back Service to identify it for data modifications. This field is case-sensitive. For instance, if fields from the Product table have been incorporated into the visual, the field value should match (**Product**), indicating the intention to save modifications to the Product table.

> **Important:** Ensuring the correct configuration of this field is crucial, any misconfiguration could result in save errors.

Tables or columns in the report should not be renamed.

For 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:

- Rename the table in the **Fields** list so that it contains exactly the actual SQL table name without the schema (in this example, SalesPerson).
- Remove and re-add the columns to the Power XL Table.
- Enter the fully qualified table name into the **Source Table** field in the format of schema.table (in this example, hr.SalesPerson).

Configure the Key Column

1. Select **Column Settings** in the **Power XL** tab of the designer ribbon.
2. Select the **Key Columns** checkbox to set the column representing the primary key of your source table. The primary key can be a single column or multiple columns, forming a composite key. In the case of a composite key, each column representing the primary key in the table must be configured as a Key Column.

3. Set the numeric fields without aggregation to **Don't summarize** in the **Visualizations** > **Columns** section.

> **Important:** This is crucial for ensuring proper functionality, particularly with an auto-increment primary key. Failure to configure this correctly will prevent changes to values in columns with defined aggregations.

Aggregated values can be displayed in the visual, but these inputs cannot be modified.

Publish and Test the Report

After completing the report, publish it to either the Power BI Report Server or the Power BI service. Write-back functionality only operates once the report is published, so full functionality testing in Power BI Desktop during authoring is not possible.

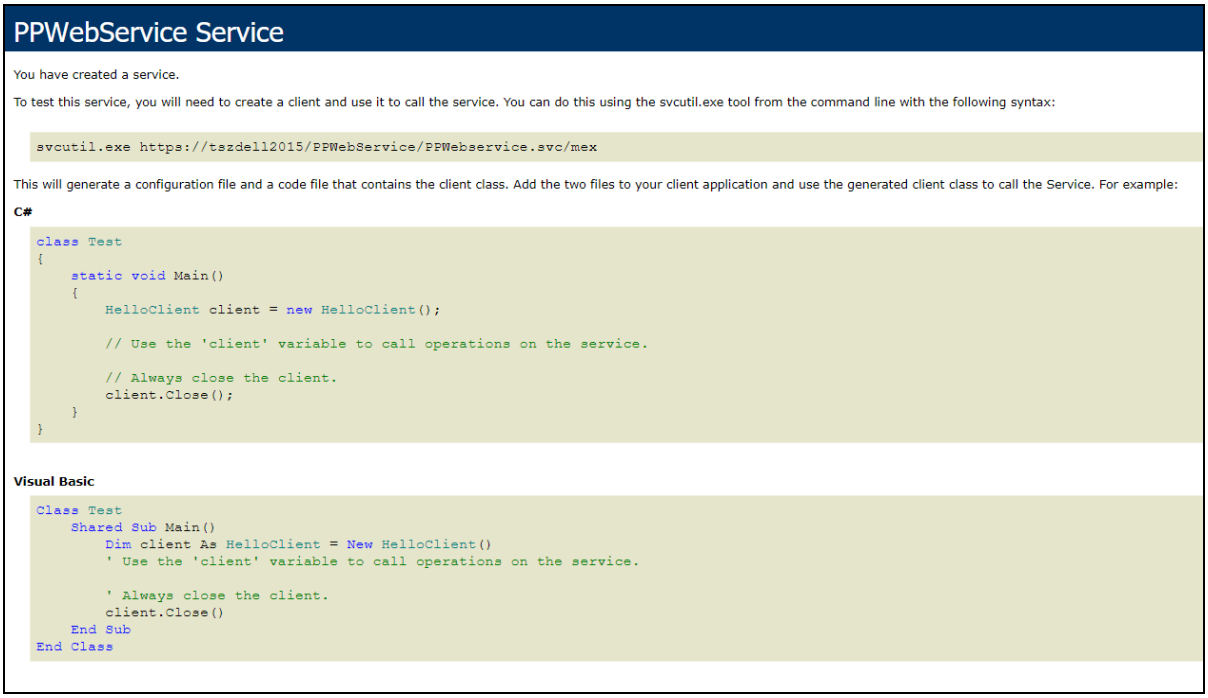
Once published, modify a cell value and click the **Save Changes** button. The modifications should then be visible in the report.

By following these steps, you will create a simple, functional report with write-back capabilities using the Power XL Table visual.

Validate the Write-Back Service

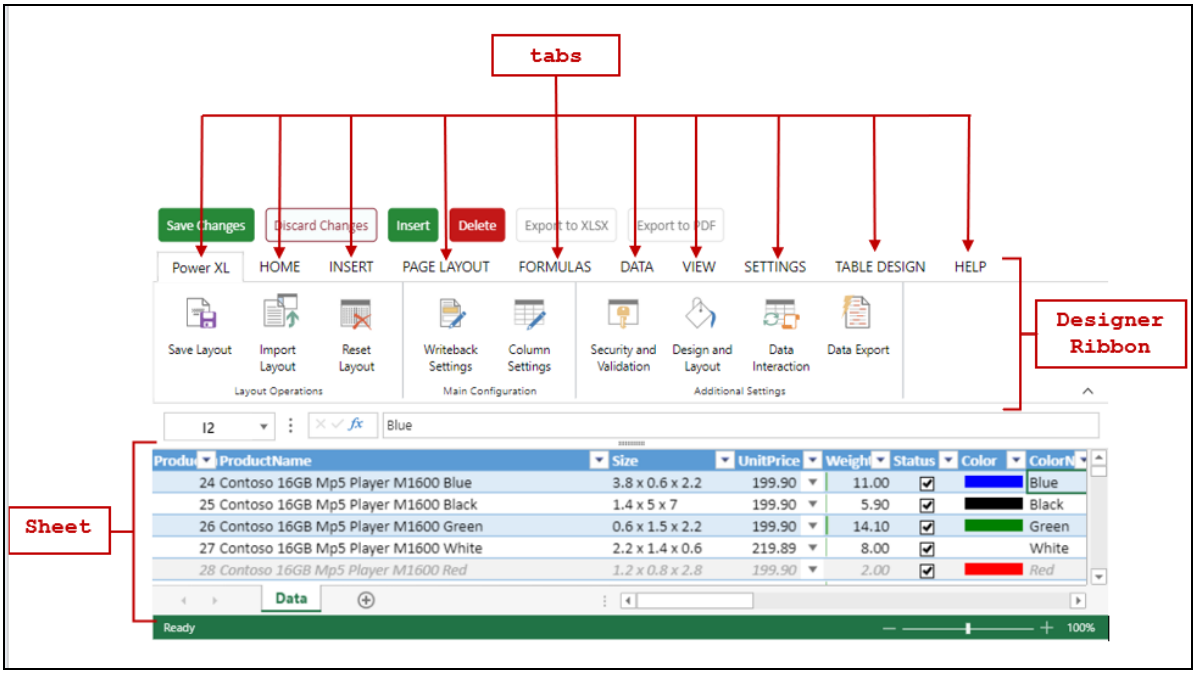
After installation, it is advisable to verify that the Write-Back Service is properly installed. You can check this by navigating to the deployed URL in the following format: http (s)://yourserverName/PPWebService/PPWebService.svc

If you see the following page in your web browser, it indicates that the Write-Back Service is up and running.



Power XL Table Legend

This section provides an overview of the Power XL Table legend.



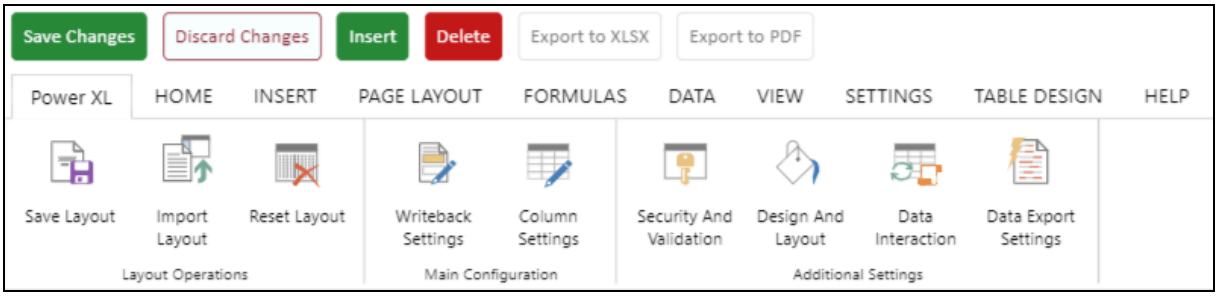
The designer ribbon can be collapsed by clicking the ^ symbol in its bottom right corner.

Helper tooltips appear if you hover above the following icons.

Power XL Table Usage

This topic outlines how to use the Power XL Table visual to create comprehensive reports.

Power XL Tab



Layout Operations

Save Layout

The **Save Layout** button is used to save the configurations and settings made on the designer ribbon or within the sheets. When the save layout process begins, a loading circle in the top right corner indicates that the system is actively saving the layout. Once the process is complete, a *Layout Saved* message appears, confirming that the configurations and settings have been successfully saved.

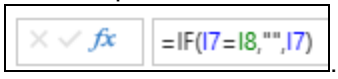
> **Important:** The **Save Layout** functionality focuses on preserving layout preferences and settings without directly impacting the database. This feature allows users to save and recall their customized layouts conveniently, without affecting the underlying data.

If you have a complex PXLT report with calculations that refer to different columns and plan to add more columns to the table, please consider the following recommendations:

- Save the layout after adding new fields: Whenever you add a new field to your report, ensure you save the layout. When the layout order changes, a formula recovery code runs automatically in the background. If you make further changes without saving the layout, those unsaved changes will be ignored, treating all modifications as simultaneous and complicating code management. To avoid this, always save the layout after adding a new field or changing the column order.

- Use actual column names: It is recommended to use actual column names, for example, @ColorName instead of references based on column letters, such as 'B6' or 'AD6'. This practice helps maintain clarity and reduces errors in your calculations.

  For example, use this 

instead of this 

.

Import Layout

This button allows you to import an Excel file that may contain multiple sheets with tables and data. If a new file is imported over an existing one, the new import will overwrite the previous data. If Power XL Table does not have a data table to merge with the Excel data table, the following error message appears: *Importing data underneath the data sheet is not supported.*

Reset Layout

Click the **Reset Layout** button to delete the current layout, clear the content of external cells, and cancel any ongoing imports. It does not reset Power BI settings on the designer ribbon. A confirmation message appears to verify the reset action once done.

Main Configuration

Writeback Settings

- **Save Confirmation Pop-up** - Set the time interval (in seconds) for displaying save confirmation pop-ups. A default value of 0 keeps the pop-up on screen until manually closed. You can configure up to 30 seconds for the pop-ups to close automatically.

- **Windows Authentication** - In an on-premises environment using Power BI Report Server, enable this setting to properly pass Windows credentials to the underlying data source. If the reports are published to the Power BI service using Azure AD, disable this setting to utilize AD credentials when accessing the data source.

  Enabling Windows Authentication in Gateway will cause the visual to post the Windows login context instead of using Power BI service credentials within the write-back service request. This value (for example, domain\user instead of username@domain.com) will be set when using USERNAME() in computed and/or default value columns, 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.

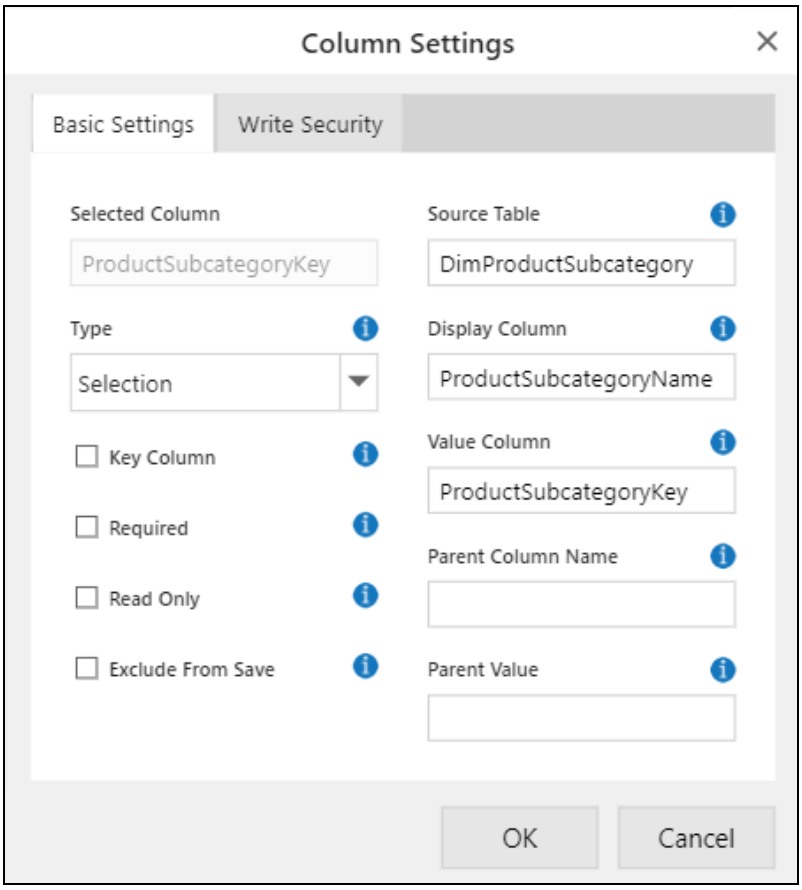
- **Auto Refresh** - When enabled, the visual refreshes automatically after saving. When disabled, it does not refresh after the **Save Changes** action, allowing you to view the changes. This setting is enabled by default.

- **Empty Cell Writeback Behavior** - Allows you to control how empty cells are handled during writeback operations. Select one of the following options:
  - **Write As Empty String** (Default) – Sends "" to represent empty cells, which is the standard behavior for most systems.
  - **Write As Null** – Sends NULL to represent empty cells for systems that explicitly require null values during writeback.

Refer to Power XL Table Setup and Configuration to review the other settings in this panel.

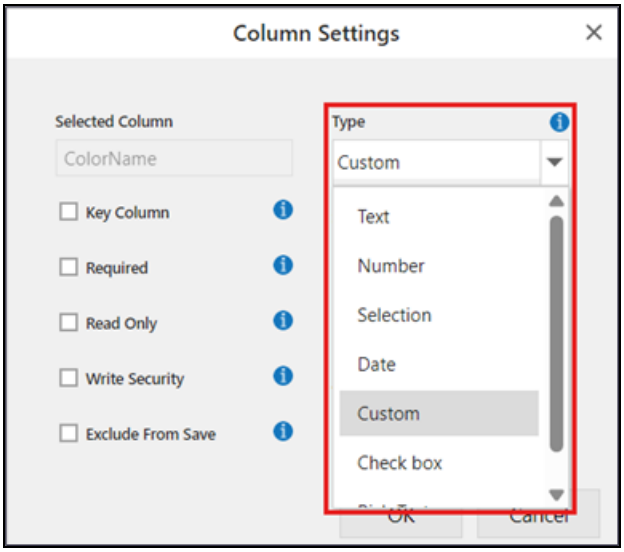
Column Settings

To access these settings, click the **Column Settings** button on the **Power XL** tab. There are two tabs in the **Column Settings** panel - **Basic Settings** and **Write Security**.



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

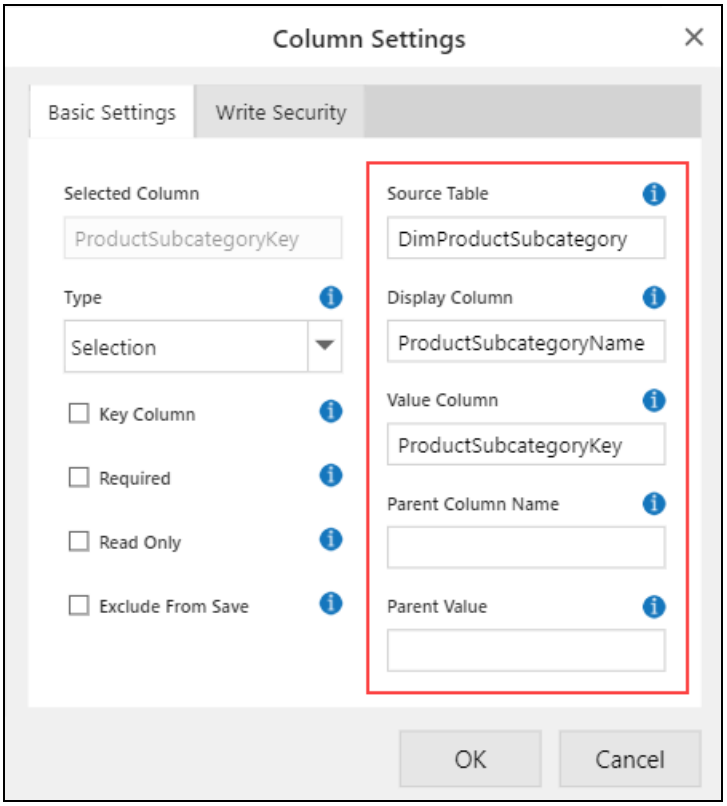
One-line Textbox in the grid.

Number

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

Selection

The values in the drop-down are sourced from a related or lookup table, requiring additional properties to be configured for this column type. For instance, when a data table has a foreign key with a numeric column that is in a relationship with another table, you can display the corresponding text representations instead of numeric values and allow selection from a set of values.



Configure the following properties:

- **Source Table** - The drop-down list values are sourced from the source table. Enter the table name as displayed in the Fields pane. For SQL databases, provide the SQL table name. If the table is in a different schema than "dbo", use the format: schemaName.TableName.

When populating this field, the visual can access the data in two ways. By default, it uses the write-back service to query the underlying data source. However, this method does not honor RLS defined in SSAS, as all data is fetched from the source SQL table. Additionally, this extra round trip to the server can sometimes be slow. To improve performance and honor existing RLS, you can use the following technique:

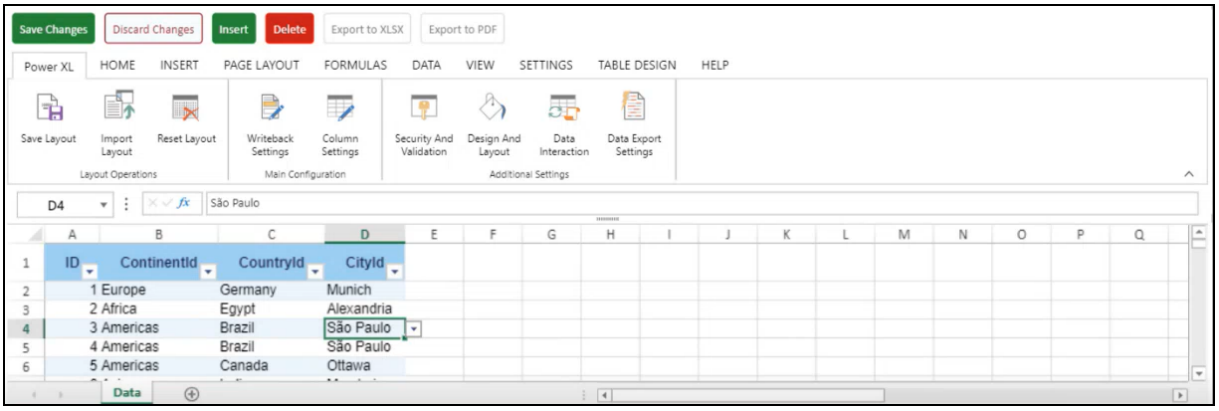
- Import the SmartFilter Helper visual into the report and add it to the report page. You can use the one provided in the setup kit or download it from the Power ON Store (store.poweronbi.com).
- Add the fields used for the drop-down to the SmartFilter visual (Display Column (name), Value Column (ID), Parent Value).

This method ensures that the available values for the dropdown are stored in the SmartFilter, allowing Power XL Table to recognize them and preventing it from querying the write-back service.

- **Display Column** - Specify the display name for the column. This is the text field from the lookup table that is displayed after the lookup.
- **Value Column** - Specify the value ID behind the display name. This is the key column in the lookup table that corresponds to the value in the current column.
- **Parent Column Name** - Specify the parent column name to enable cascading validation for the selected column. This child column will present options in the drop-down based on the parent column. Use the column name as per the data table, not the field name.
- **Parent Value** - Specify the parent column value as in the SmartFilter. This is an optional field; if the **Parent Column Name** value matches the field name in the SmartFilter, the system searches for that value. For cascading validation, the SmartFilter can handle multiple fields.

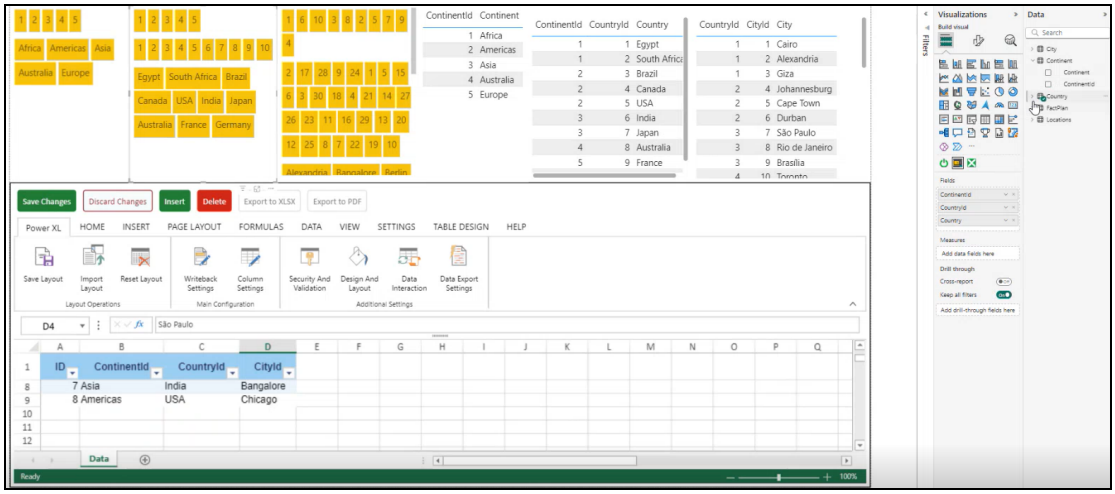
Cascading Validation

Cascading drop-downs ensure that subsequent drop-down lists display options based on the selection made in the preceding drop-down. For example, selecting Europe in the Continent drop-down will filter the Country drop-down to only show countries in Europe. If you then select Hungary as the country, the City drop-down will display cities in Hungary. The **Parent Column Name** and **Parent Value** fields enable this feature. Cascading validation also works with other column types, but all levels must have the same type.

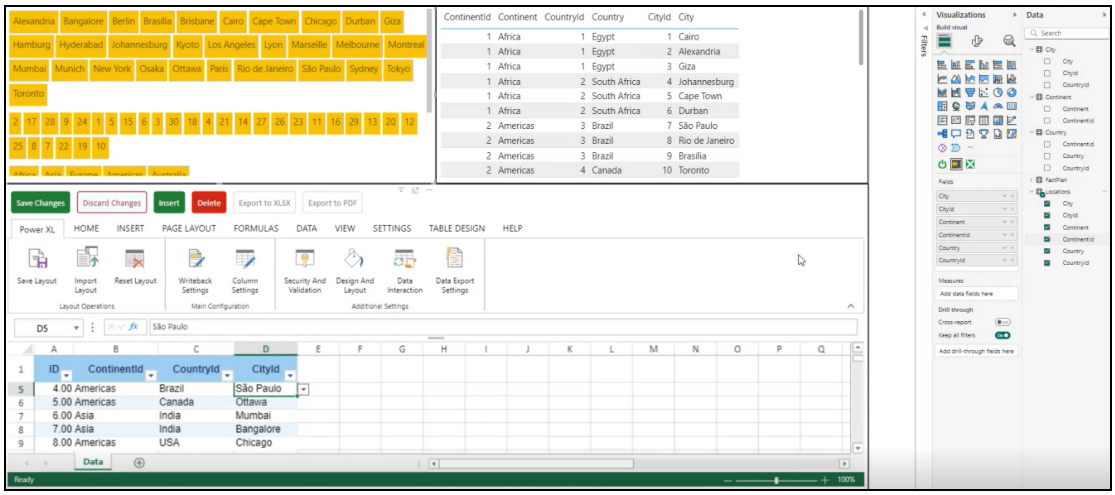


Cascading drop-downs work with both hierarchical and flat tables:

- Hierarchical Data Table - Data tables are arranged hierarchically, where the lower-level table references the table above it, such as the Country table referencing the Continent table. For hierarchical data structures, you need multiple SmartFilters, one for each data table. When using a SmartFilter as the source table, the **Source Table** field must be configured with the filter associated with that column, for example, continent for continent, country for country, and so on. In the SmartFilter, users can combine fields from different hierarchies.



- Flat Data Table - This is a consolidated table containing all the data in a single table. With a flat table, you can use either multiple or a single SmartFilter to retrieve the full data table. When using a single SmartFilter as the source table, the **Source Table** field must be configured with the single filter, for example, Location for the continent, country, and city columns.



Recommendations for using cascading validation:

- Use a single SmartFilter to retrieve values from a flat data table.
- Use a single SmartFilter to retrieve values from a hierarchical data table, unless the fields are not connected.

Limitations of cascading validation:

- Currently, it supports up to 30,000 records in the drop-down list. If you select a value near the end of the list, subsequent child drop-downs may not display all values due to exceeding the 30,000 record limit in larger datasets.

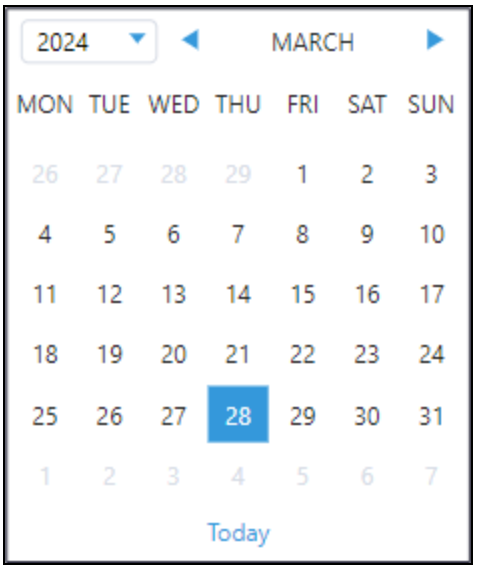
Tip: If the drop-down list is lengthy and not all values are visible, a scroll bar will appear on the right side. Additionally, you can search among the values by typing the first letter, which will jump to and highlight the first value starting with that letter. For example, typing "h" will automatically jump to and highlight the "Headphones" value.

Note: You can restrict the available values in the drop-down list. After adding the fields to the SmartFilter in the report, apply a visual-level filter on the SmartFilter to limit the available values. Consequently, not all possible values will be visible in the drop-down for selection purposes; however, rows in the Power XL Table visual with values outside 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.

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.

Write Security

To access this setting, click the **Column Settings** button on the **Power XL** tab and then select the **Write Security** 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:

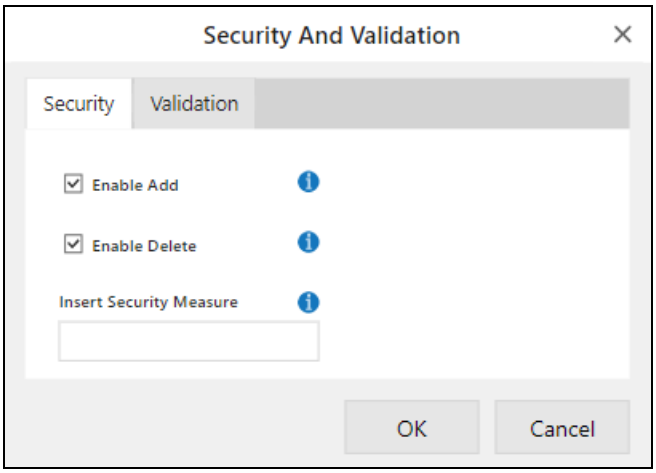
https://help.insightsoftware.com/s/article/implement-complex-write-security-per-entity

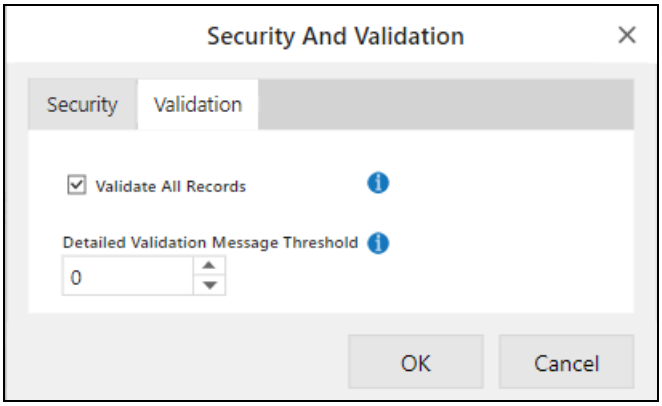
The **Apply to Selected Columns** checkbox becomes available only when the **Write Security** setting is enabled. It allows you to specify which columns should be subject to write security measures. If this checkbox is disabled, write security applies to all columns. This setting is also available under **Column Properties** for each individual column. Enabling it marks the column as a Protected Column.

- **Available Columns** - Lists all the available columns in the grid.
- **Protected Columns** - Move the required available columns to this list to make them non-editable and ensure all cells within these columns are read-only. Select the required column or click **Select All** and then click **Add to Protected**.

A column will now be fully editable only when the visual is in Edit mode and no write security measures are applied to it, which means the write security column must not return a value of 0 or null.

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 rows 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.

Delete Selected Row(s)

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

When enabled, you can delete the selected row by selecting either the entire row or a cell from that row, even if the cell is outside the data table. When disabled, the **Delete** action will delete the first row. This setting is available only when the **Enable Delete** checkbox is enabled.

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 or true, 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.

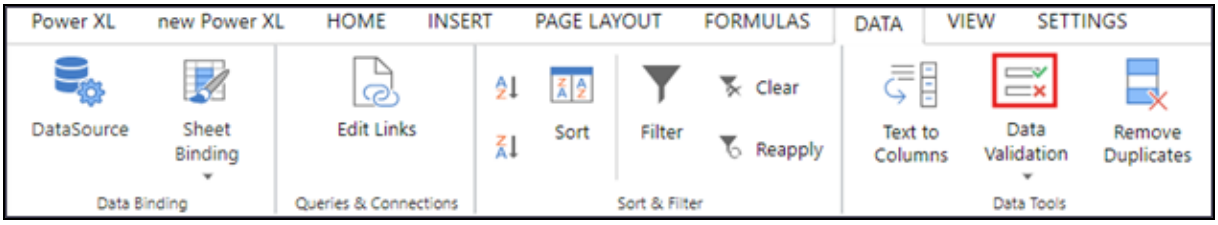
Example of complex reports:

Consider a scenario where calculations always refer to a different row. Building the dependency tree takes significantly longer than activating the **Validate All Records** setting. If this setting is turned OFF, the calculation will process the entire table.

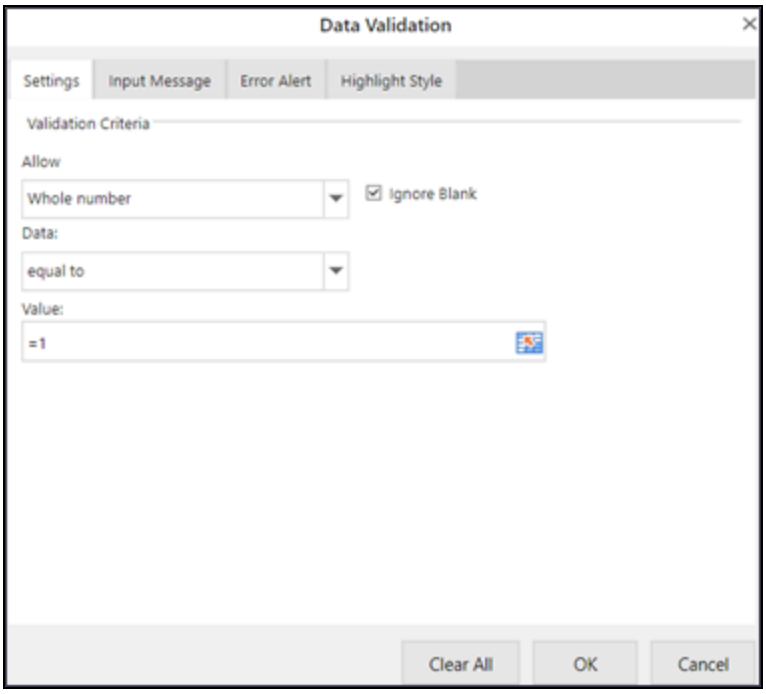
Additionally, when you modify a value, you might notice that typing becomes slow in the affected cells, even with a table containing just a hundred rows. Therefore, when the **Validate All Records** setting is disabled, be cautious with cell references, as referring to another row will trigger the validation of the entire table.

Data Validation

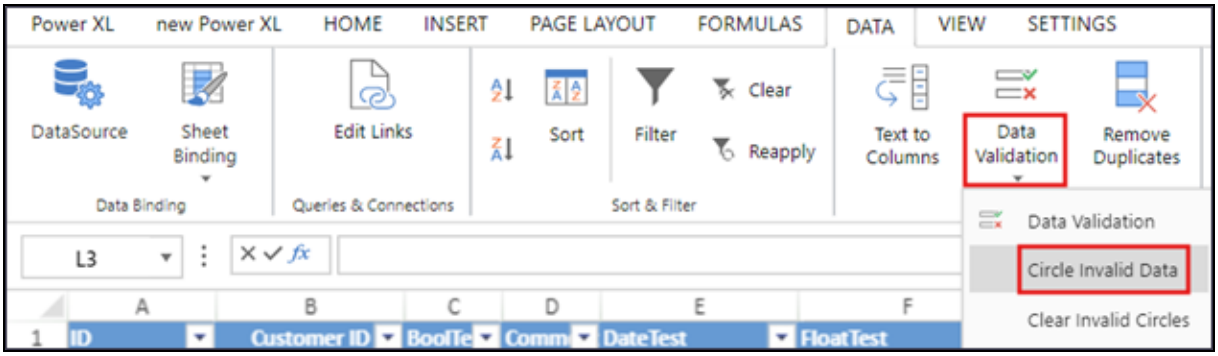
Data validation can be set up for columns in order to ensure that they contain data that agrees to set criteria. Data validation can be set on a column by selecting a cell in the column, navigating to the **Data** tab, and clicking on the following icon.



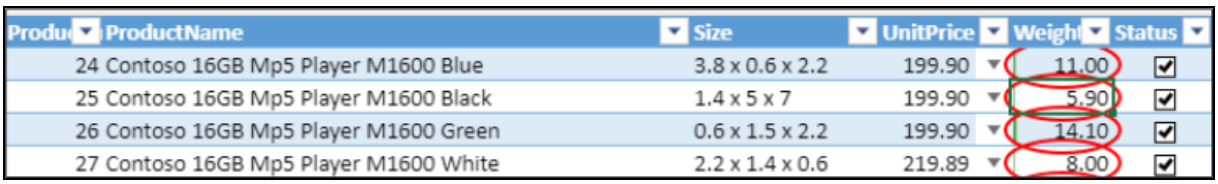
In the following pop-up window, the Validation Criteria can be set up and applied to the selected column by pressing **OK**.



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.

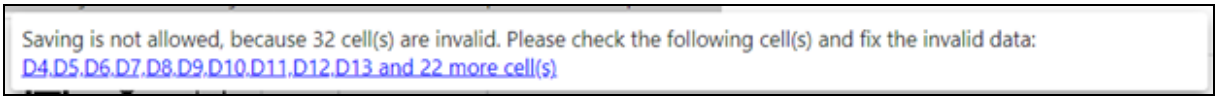


Detailed Validation Message Threshold

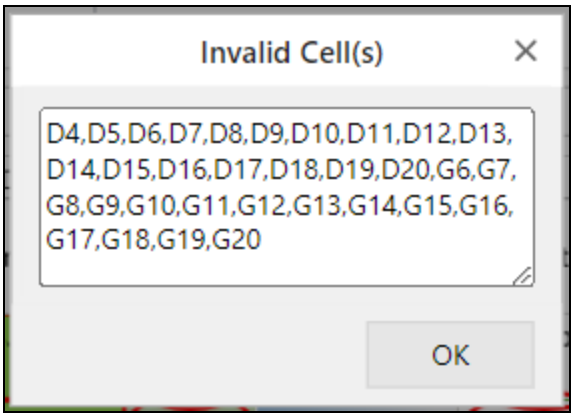
To access this setting, click the **Security and Validation** button on the **Power XL** tab and select the **Validation** tab.

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.

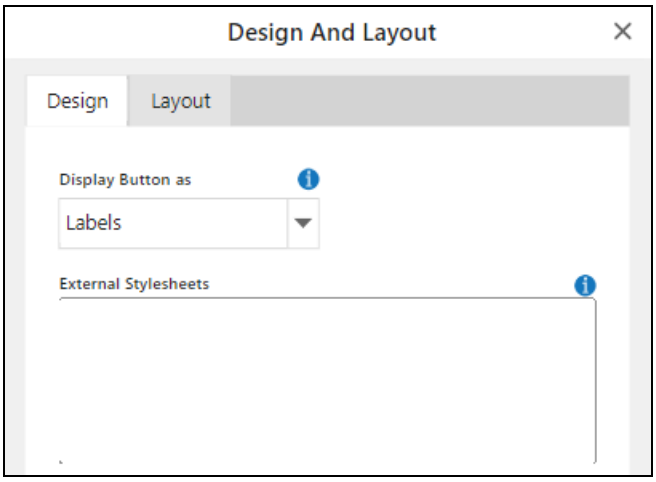


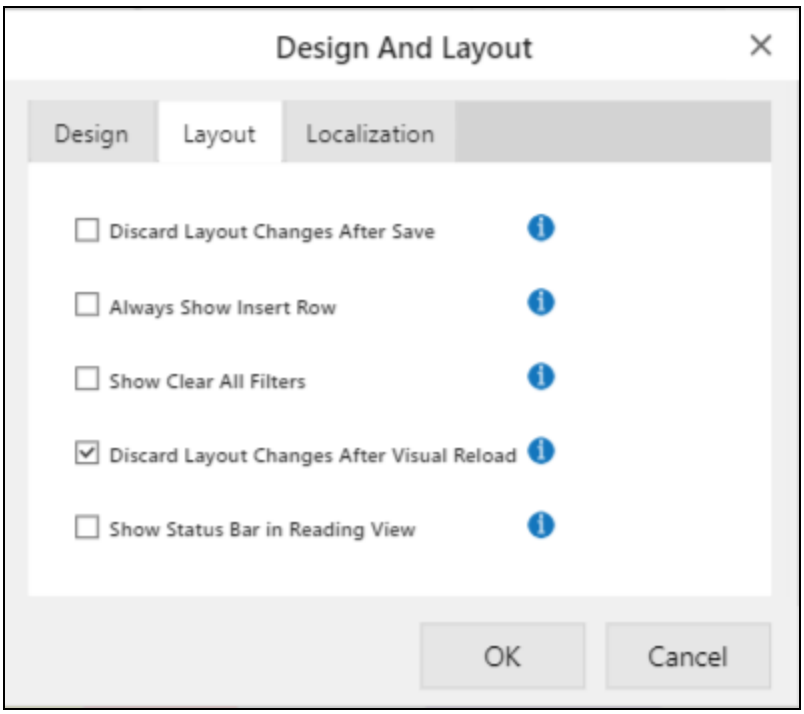
By clicking **more cell(s)** in the message, you can view the coordinates about the additional invalid cells in a new dialog box.

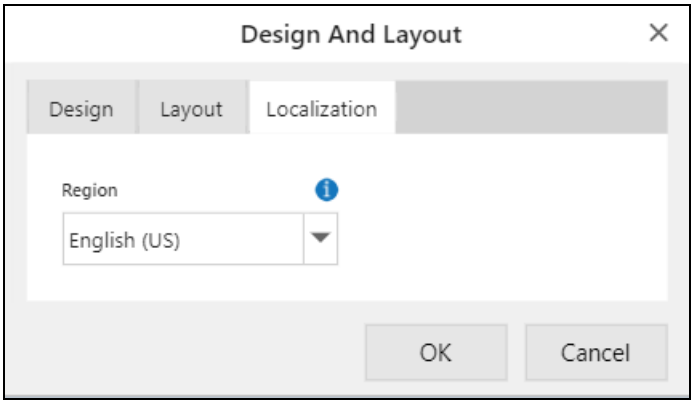


The default value of the setting is 0. If you retain the default value, then the warning message appear as "Some of your cells are invalid".

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;
}
```

Below are some CSS examples to style the buttons in Power XL Table.


```
}
.save-button {
background: #ffbf00 !important;
border-color: #ffbf00 !important;
border-width: 0px !important;
color: #000000 !important;
```

```
}

.save-button:hover {
background: #153050 !important;
border-color: #153050 !important;
border-width: 0px !important;
color: #ffffff !important;
}
```

To apply fill color and hover effects:

```
.discard-button {
background: #ffbf00 !important;
border-color: #ffbf00 !important;
border-width: 0px !important;
color: #000000 !important;
}

.discard-button:hover {
background: #153050 !important;
border-color: #153050 !important;
border-width: 0px !important;
color: #ffffff !important;
}
```

To change the labels, rename the Save Changes button to Save:

```
.save - button {
position: relative;
color: transparent;
}
.save - button::after {
content: "Save";
background - color: green;
color: white;
position: absolute;
top: 0;
```

```
left: 0;
width: 100 % ;
height: 100 % ;
display: flex;
align - items: center;
justify - content: center;
}
```

To hide the Delete button:

```
.control - buttons__button.delete-row - button {
display: none
}
```

To hide the tooltips:

```
.MuiTooltip - popper div {
display: none;
}
```

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	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

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	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	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

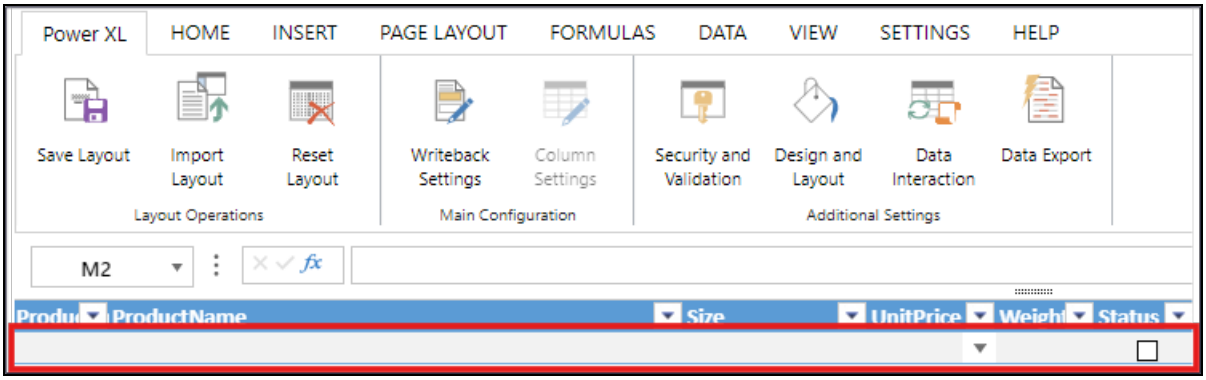
Data table after successful save:

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

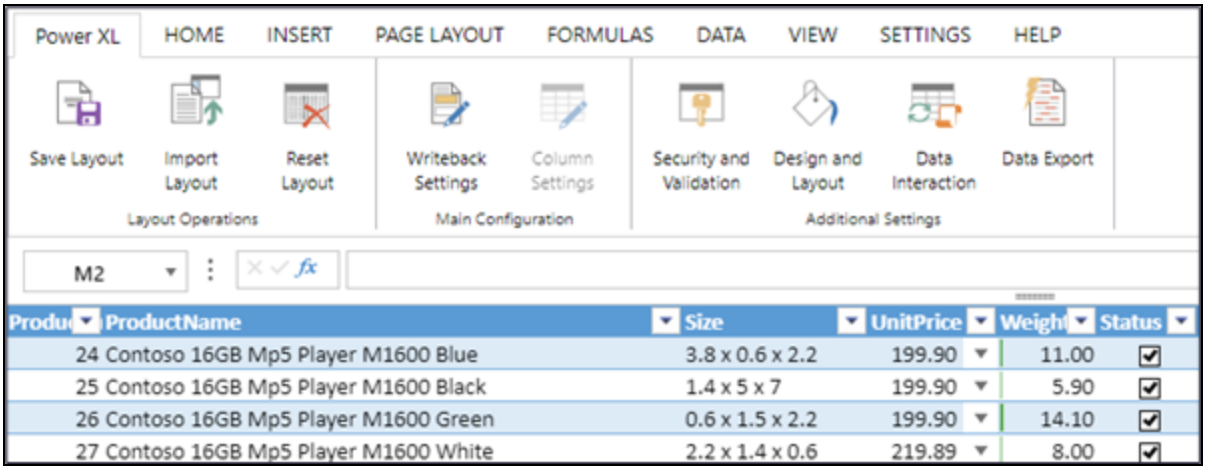
Always Show Insert Row

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

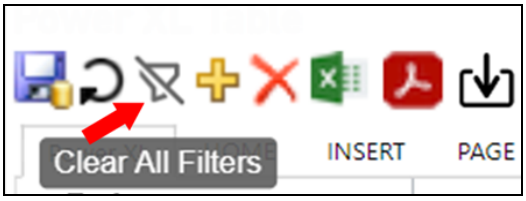
The **Clear All Filters** button enables the removal of all the filters that have been applied to the column headers in the visual.

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

Show Status Bar in Reading View

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

Turn ON this setting to display the timestamp of the last successful save in the grid's status bar when in reading mode. The status bar also displays the average, count, and summary values for the numbers selected in the grid.

Empty Row Count Below Table

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

This setting allows you to specify the number of additional rows to display beneath the data table based on your preference. The minimum value is 0 and the default value is 100.

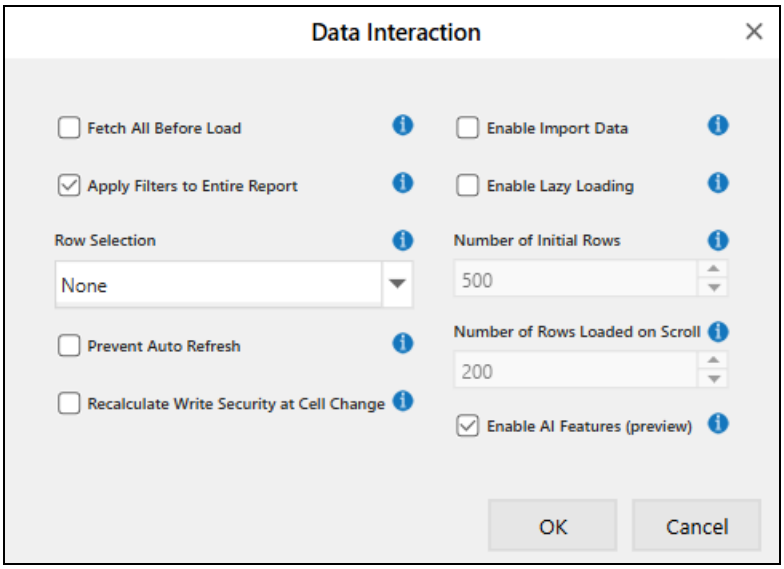
Region

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

This setting allows you to specify the input format for decimals, thousand separators, and dates, displaying the column values accordingly. Select **Auto Detect** from the drop-down menu for the visual to identify the browser language and set the local format, or select any language/region of your preference. The default value is **English (US)**.

Note that for date-type columns, you must delete the existing column in the grid and re-add it to apply the updated format.

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**
  - 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 15 and 30000. Trying to apply a value outside this interval will cause the value to return to either 15 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.

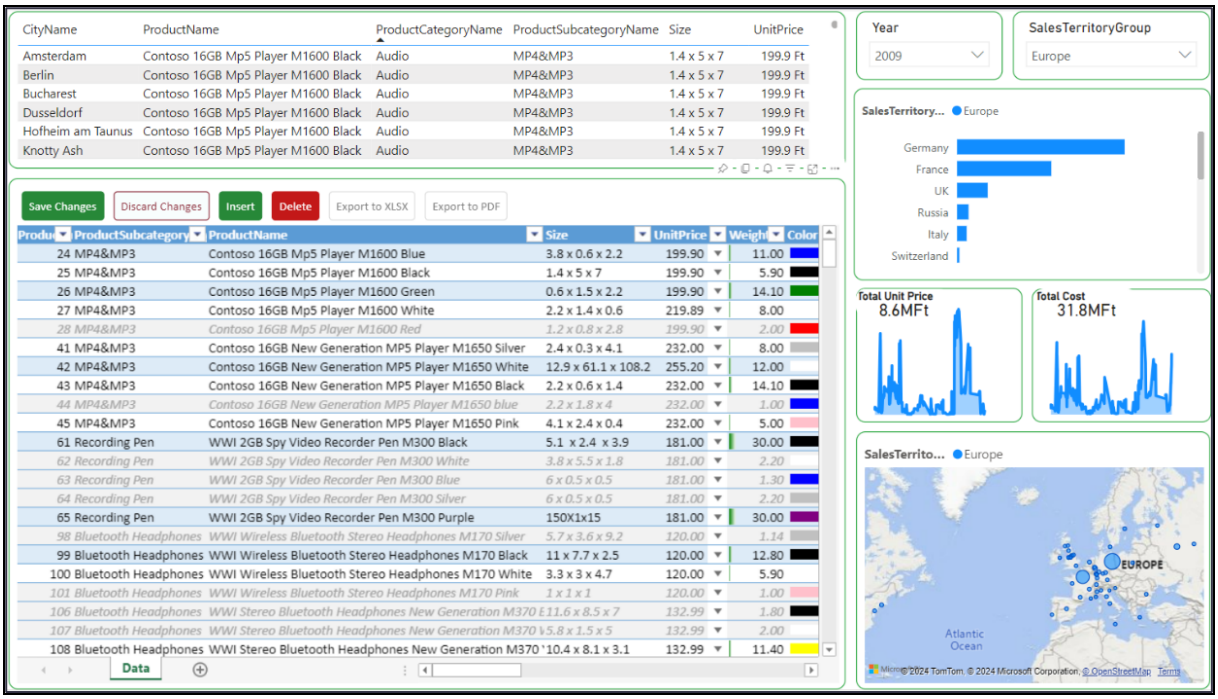
This setting allows the column filtering of the visual to be applied to all the other visuals on the same report page.

- 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.

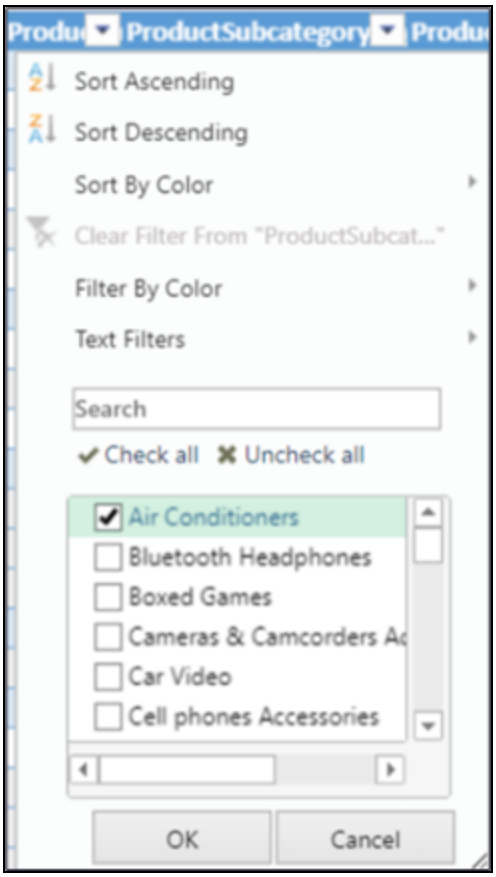
- When the setting is ON, if a filter is applied to a column header, the filter effect will be applied to the entire report, affecting all visuals.

The setting is demonstrated in the use case below.

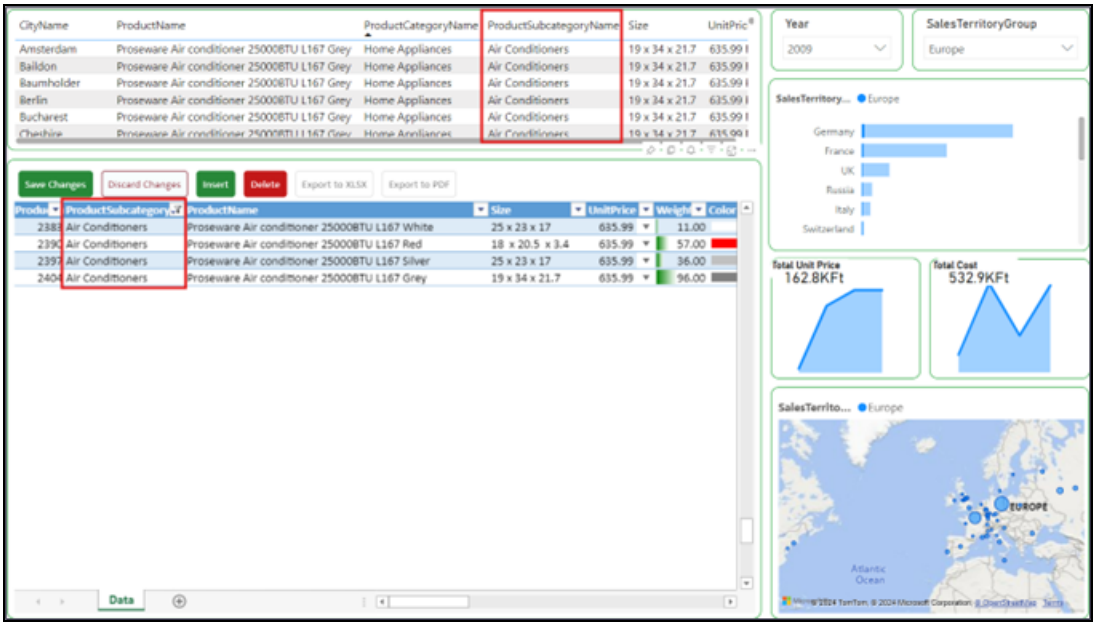
Before applying a Column Filter all records are visible in all the visuals of the report:



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.

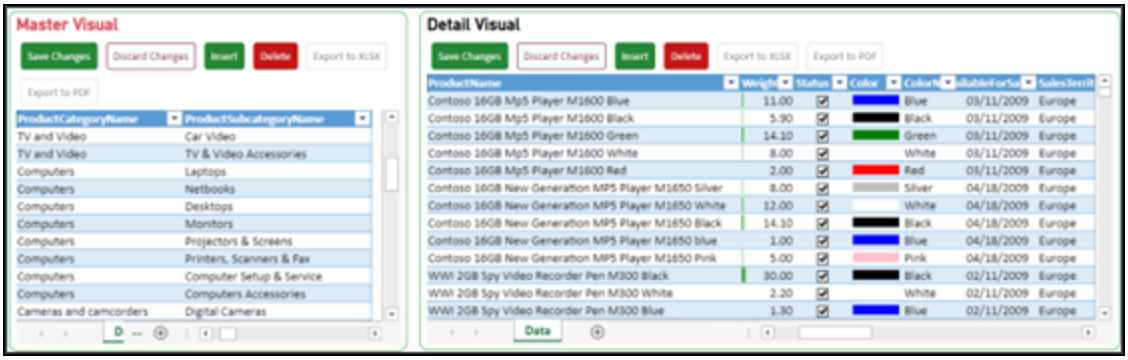
Row Selection allows filtering records through a Master visual. By selecting a row in the Master visual, the Detail visual will re-render with the records that fit the ID of the rows selected in the Master visual.

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.

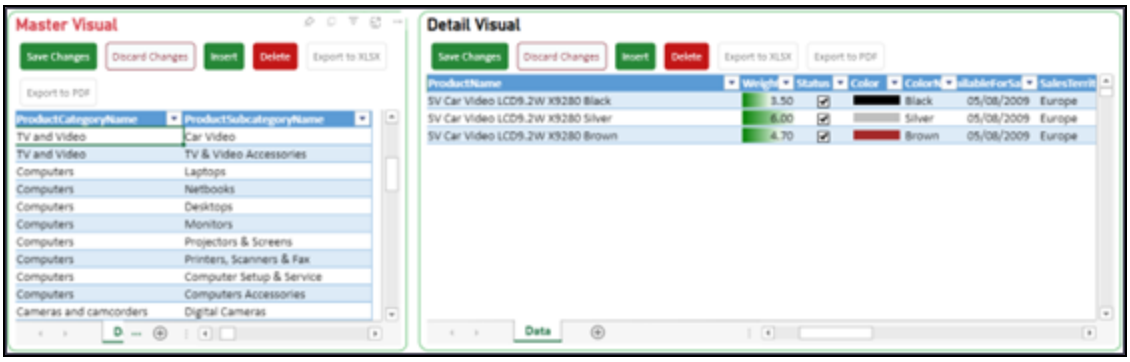
In order for this to work, the “Row Selection” setting must be set up in the Master visual.

The master visual, where the Row Selection setting is enabled, must include a Key Column (found in the Column Settings property group). Without a Key Column in the master visual's data table, the Row Selection feature will always filter out only the last row, regardless of the rows or sections the user intends to display.

Before Row Selection:



After Row Selection:



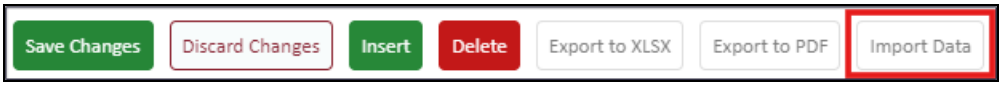
The **Row Selection** dropdown box has the following settings:

- **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

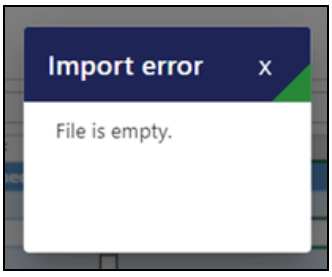
When merging data, consider the following scenarios:

- 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."

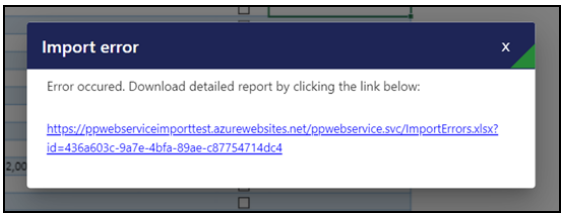
Error Handling

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.



Prevent Auto Refresh

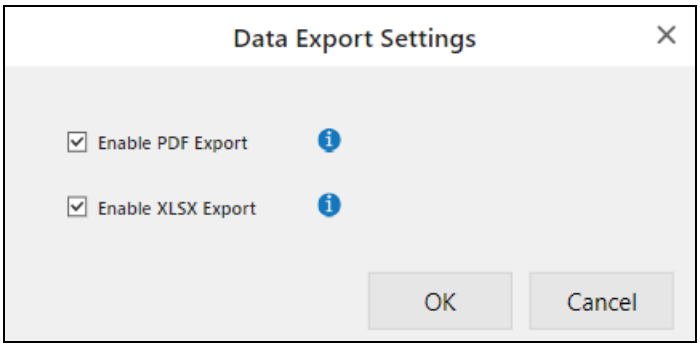
To access this setting, click the **Data Interaction** button on the **Power XL** tab.

This setting blocks grid updates when you have unsaved changes. It prompts you to save or discard your edits, preventing unintentional loss. Turn on this setting to create a block period for automatic refresh.

Enable AI Features

See Enable AI Features for detailed information.

Data Export Settings

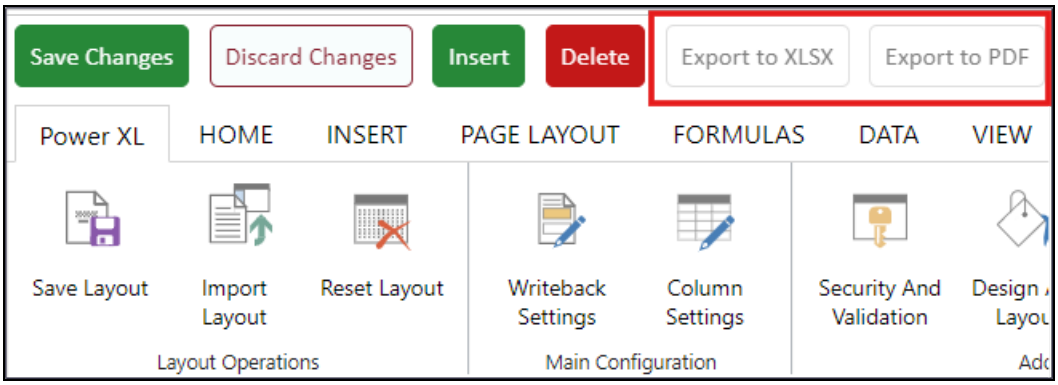


Enable XLSX Export / Enable PDF Export

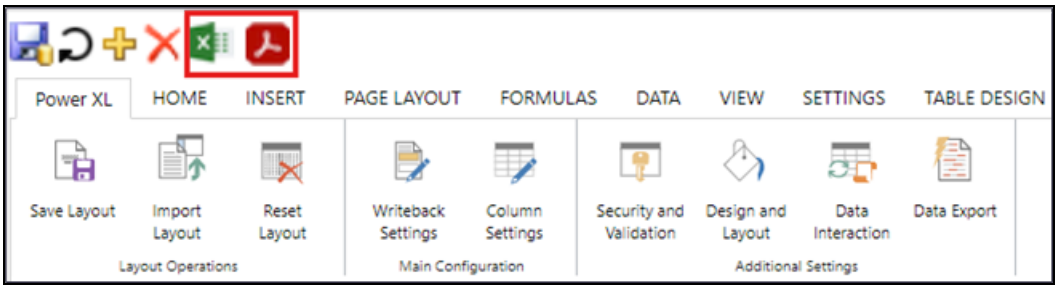
To access these settings, click the **Data Export Settings** button on the **Power XL** tab.

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.

Enable AI Features

This topic covers the AI-powered features introduced in Power XL Table v25.3. Power ON AI currently supports BYOM (Bring Your Own Model), which allows you to integrate your own Azure OpenAI instance to power AI capabilities within your Power ON environment.

Prerequisites

Before you can enable and use AI features, you must complete the following setup steps:

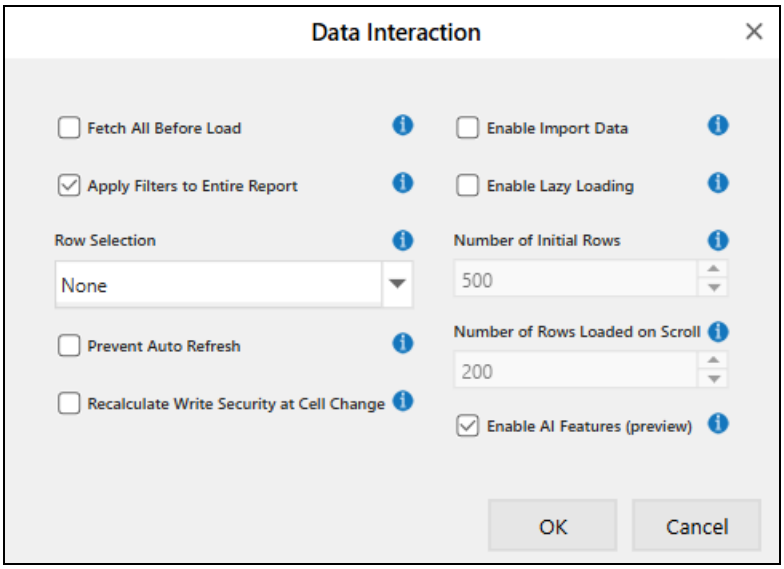
1. Create your own Azure OpenAI instance. This is currently the only API supported.
   a. Go to Azure AI Foundry OpenAI Models.
   b. Follow the setup instructions to create your instance.
2. Add the API key for your Azure OpenAI instance to the Azure Key Vault you are using for your Power ON backend.

   Note: Store the API key as a secret in your Azure Key Vault.
3. Create a model deployment in Azure AI Foundry.
   a. Access Azure AI Foundry (available after creating your OpenAI instance).
   b. Deploy your chosen AI model.
4. Configure the following settings in the Power ON backend web.config file:
   i. OpenAiServiceApiKeyName: The name of the secret in Azure Key Vault that contains your API key
   ii. OpenAiServiceUrl: The URL of your OpenAI deployment
   iii. OpenAiDeploymentName: The name of the AI model to be used and deployed (defaults to gpt-4.1 if not set)

Note: Refer to Microsoft's guide How-to: Create and deploy an Azure OpenAI in Microsoft Foundry Models resource - Azure OpenAI for complete setup instructions.

Enable AI Features

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

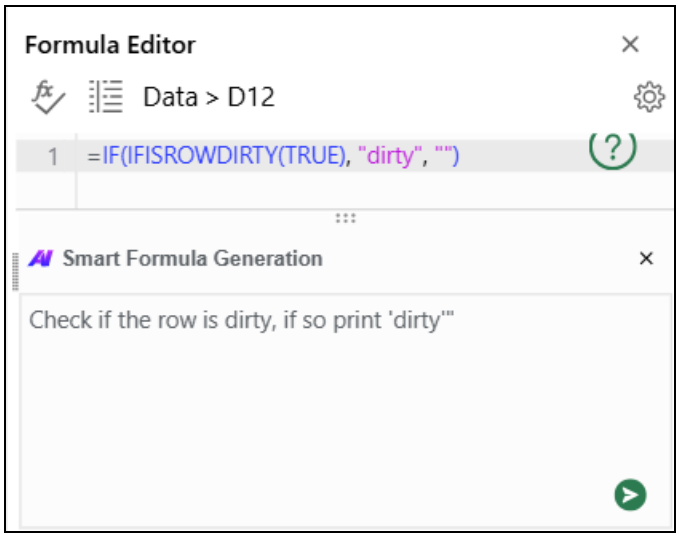


After you turn on the **Enable AI Features** setting, you can use the following capabilities.

Generate Formulas with AI

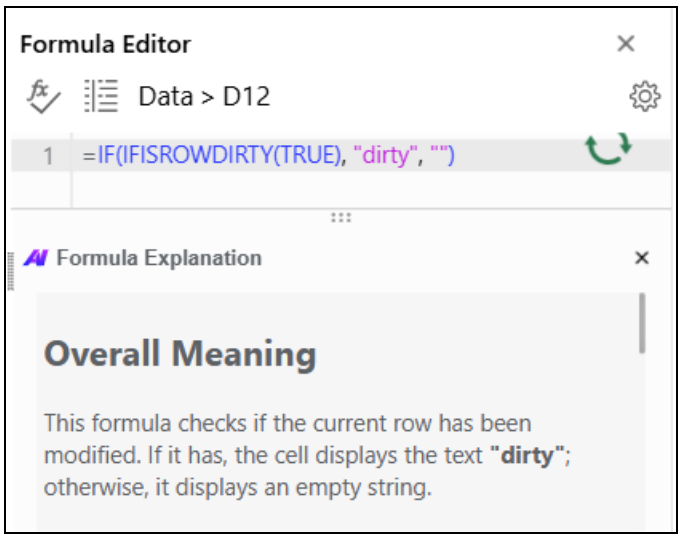
You can use AI to help you create formulas.

1. On the **FORMULAS** tab, select **Show Formula Editor**.
2. In the formula editor, select the **AI** button  .
3. In the **Smart Formulas Generation** field, describe the formula you need. For example, "Check if the row is dirty, if so print 'dirty'".
4. Submit your description.



5. After AI provides the formula, select the question mark icon to see an explanation of how the

formula works.



Create and Analyze Pivot Tables with AI

AI can help you set up and understand your pivot tables.

Smart Pivot Layout Generation

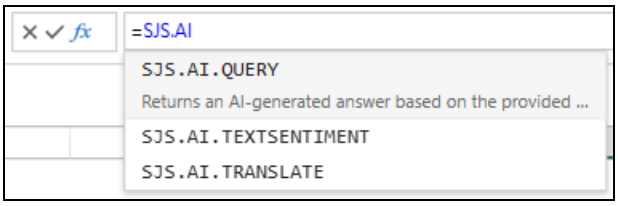
1. On the **INSERT** tab, go to **PivotTable**.
2. Select the **A** icon.
3. In the **Smart Pivot Layout Generation** field, describe what you want in your pivot table. For example, "Sum of sales by region".
4. Apply your description. The pivot table fields update automatically based on your requirements.

Intelligent Data Analysis

1. On the **INSERT** tab, go to **PivotTable**.
2. Select the **I** icon.
3. Ask your question in the Intelligent Data Analysis field. For example, "What is the sum of total sales?".
4. Submit your question. AI generates an answer for you.

Use AI Formulas

You can type AI-powered formulas directly in the formula bar. Start by typing =SJS.AI and then choose from these formulas:



SJS.AI.QUERY

Returns an AI-generated answer based on text content.

Syntax: =SJS.AI.QUERY("<prompt>", "<value>")

Example: =SJS.AI.QUERY("Summarize this feedback", A2) returns a summary of the text in cell A2.

SJS.AI.TEXTSENTIMENT

Classifies the sentiment of text as Positive, Negative, or Neutral.

Syntax: =SJS.AI.TEXTSENTIMENT("<value>", "positive, negative, neutral")

Example: =SJS.AI.TEXTSENTIMENT(B3, "positive, negative, neutral") analyzes the text in cell B3 and returns its sentiment classification.

SJS.AI.TRANSLATE

Translates text into a specified language.

Syntax: =SJS.AI.TRANSLATE("<value>", "<language>")

Example: =SJS.AI.TRANSLATE(C4, "Spanish") translates the text in cell C4 to Spanish.

When you turn off this setting, the AI icon is removed from the formula editor and pivot tables. Any AI formulas in your spreadsheet display the message "AI features are disabled in the settings."

Known Issue

AI-generated values are not cached, causing the AI engine to make new API calls each time the report reloads, which may impact report loading performance.

Power XL Table Rules and Limitations

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

- Naming Limitations
  - Editable columns and tables in Power BI reports must retain the same names as in the data source.
  - For Semantic models, names must match the entities in the model or report Fields sections.
  - For SQL data sources, entity names must match those in the SQL tables (for example, "Customer" in SQL should not be "Customers" in the report). This applies to field or column names as well.
  - Brackets ("[" and "]") are not supported in measure names.
  - Dots (".") are not supported in table names but can be used in schema names. If table names contain unsupported characters, create a compliant view and use it for writing back.

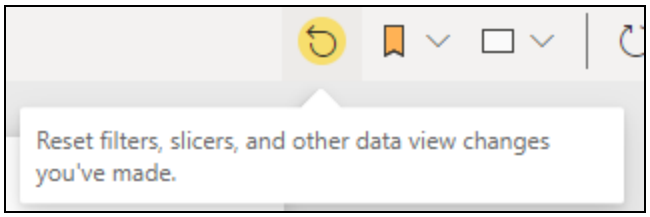
- A maximum of 30.000 records can be displayed at one time. If the underlying table contains more rows, consider using slicers in the report to edit a subset of the data and maintain a seamless workflow. For optimal performance, it is recommended to reduce the displayed table size using slicers.

- Write-Back Limitations
  - Write-back to measures is not supported; only row-level data can be modified. Use the Data Entry Matrix (DEMx) visual for modifying measure values.
  - Non-writable views as data sources for fact tables are not supported. To address this, materialize the view into a physical table, simplify the view, or create INSTEAD OF INSERT | UPDATE triggers. Non-writable views often include complex SQL queries with multiple joins, CASE statements in WHERE clauses, Common Table Expressions (CTEs), and aggregations. To test the view, duplicate it under a different name and execute an INSERT statement against it.

- On-premise Power BI Report Server requires at least the January 2019 version, along with Power BI Desktop for Report Server 2019 January or newer versions.

- Versions older than v1.2.0 may display custom columns instead of PBI columns after changing bookmarks. Save the original layout to resolve this (not the bookmarked one). Use the **Save Layout** button on the Designer Ribbon or the **Save** button in Edit mode.

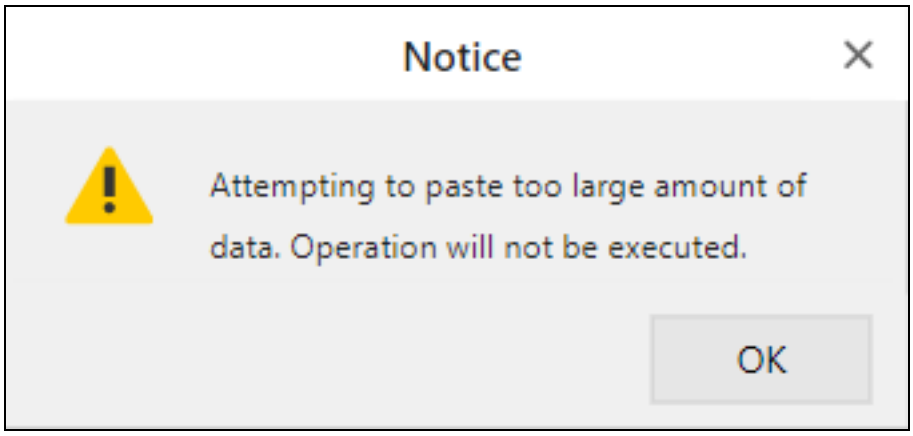
- The **Reset filters, slicers, and other data view changes you've made** yellow button indicates that the report has been modified. Select it before entering Edit mode to avoid retaining Reading view settings.



- Cutting Rules
  - Cutting the entire data table is allowed in Edit mode but not in Reading View.
  - Cutting individual columns is disabled in Edit mode regardless of column type.

- Copy-Paste Limitation - There are limitations on the number of cells that can be pasted at once in the visual, depending on the user's context and the type of sheet they are working on:
  - Non-Data Sheets (Edit Mode, Reading View, and other views): The limit is 40,000 cells.
  - Data Sheet in Reading View: The limit is 40,000 cells.
  - Edit Mode with Formulas in Datatable: The limit is 3,000 cells.
  - Edit Mode with =IFISROWDIRTY() Formula Outside Paste Range: The limit is 1,000 cells.

  If the user attempts to paste more cells than allowed, a pop-up message will notify them of the limit. This ensures optimal performance and prevents potential issues from pasting too many cells. By adhering to these limitations, users can maintain the functionality and performance of the visual.



Upgrade Visuals

This topic outlines the process of upgrading visuals from a previous version.

1. Download the latest version of the visual from the provided URL.
2. Open a visual-related report and download the .pbix file by selecting **File** > **Download the .pbix file** option.
3. Rename the downloaded report to preserve the original version and then open the file in PowerBI Desktop.
4. Click on the three dots in the **Visualizations** pane.
5. Select **Import a visual from a file** and then select **Import**.
6. Locate the downloaded visual file, likely saved in your Downloads folder in File Explorer. Click to select it and then click **Open**.
7. You will receive a notification once the import is successful. Click **OK**.
8. Right-click the visual icon and select **About** from the menu. A pop-up window will display details, including the version number.
9. Select the visual in your report and click the Power ON <visual name> icon (for the latest version). The visual in the report will update to the newer version while retaining your previous settings.

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