Write-back & Planning for Power BI. Period

Power ON Visual Planner

Data Entry Matrix Documentation & Troubleshooting Guide

Contents

This document describes PowerON's **write back** capable tool: **Data Entry Matrix (DEMX)**. The product is a custom visual developed for Microsoft's Power BI, that enhances the user experience by enabling end users to make permanent changes on data – aka write back – by targeting measures, DAX expressions. On the following pages you will see a **detailed** overview of the matrix, you will learn how to use it in your reports together with some common use-case implementations, how to configure it properly and what are the pitfalls you might encounter and how to overcome them.

The **audience** of this document are both technicians (developers, DBAs, BI professionals) how who are aware the depths of SQL Server, SSAS Tabular models, Azure Services and Power BI, but also users who are focusing mostly on building and preparing reports using Power BI Desktop. Some sections are focusing more on technical specific subjects. If you find something very distant for you, ask for support either from your local IT or from PowerON by submitting a ticket on our site: https://support.poweronbi.com.

Overview

PowerON's Data Entry Matrix let users make changes on figures that are the results of DAX expressions defined aggregations. In traditional dimensional modelling, these aggregations are calculated on fact tables and the context of the calculation is defined by the dimension – or lookup – tables that are in relationship with the fact table. DEMX makes updating the result of your measures possible.

DEMX offers the following built-in features and components:

- Writing back values to the underlying data source
- Make changes on figures by using smart formulas or exact values
- Update values on totals, subtotals that cascades down to members
- Create comments on cells
- Supports copy and paste selected cells to / from the visual

It is worth to describe in high level **how write back works** for Tabular models. Depending on your data source (SSAS In-Memory, SSAS Direct Query or SQL only) PowerON's write back service performs the following operations:

- The service captures the modified value together with its **tuple** (intersection of dimensions on which the measure is calculated for the given cell) and the user context. After the service understands what to do, based on the SSAS model structure (relationships, table queries and measure definitions) it composes T-SQL statements that should be executed.
- Executes the compiled T-SQL statement against the underlying data source (fact table) to save the modifications
- In case of SSAS in-memory models it will reprocess the table. For performance optimization please consult the related chapter in this document.
- Initiates a refresh on the visual so that the changes will appear in the report

Support

Check out **PowerOn's Knowledge Base** articles at: https://support.poweronbi.com under Visual Planning for common use cases, tips, troubleshooting tools.

Please note, that in order to access the articles, you need to **register** on the site. It is advised as you will be able **to submit tickets** if you encounter any difficulties or issues, so that our support team can give you the best service. You can also **email** us at: vizsupport@poweronbi.com

Prerequisites

You must have a working and **configured write-back service** (PPWebService) installed in your environment. If that is set, in order to build a report, you first must **import the dataEntryMatrix v3 pbiviz** file into your Power BI Desktop instance. The visual can be found either in your **installation folder under Resources\PPWebService** in the **VPDemo.zip** archive file.

Also, DEMX can be converted from other Power BI Visuals, like other custom visuals in Power BI.

Before you begin make sure that you have a properly configured write back service (PPWebService) that is accessible from the point of the report server.

For more information about configuring the PPWebService check out PowerOn's Knowledge Base articles at: https://support.poweronbi.com under Visual Planning – Write Back Service for more details.

Please note, that in order to access the articles, you need to **register** on the site. It is advised as you will be able to submit tickets if you encounter any difficulties or issues, so that our support team can give you the best service.

In case of on-premises installations or virtual machines hosted in the cloud make sure that your Power BI Report Server can access the host IIS machine of the PPWebService. In case of Azure, the Web Applications are accessible by default. The webservice must have a dedicated connection string configured for the data source in the web.config connectionStrings section that is pointing to the used SSAS Cube.

Connections to the data sources should be configured by your IT team. Please refer to the following article in our knowledge base on how to add ones at:

https://support.poweronbi.com/portal/kb/articles/how-to-add-a-new-data-source-for-write-back

You can find examples of valid connection strings for different data sources at: https://www.connectionstrings.com/

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

- administration right on the cube, plus data reader and data write roles on the cube's underlying data source database
- if you intend to use the Commenting feature, the SQL permissions must be elevated to dbo, otherwise datareader and datawriter roles are sufficient.

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

- data reader membership on the SSAS cube

In case of impersonation enabled, then the end users must have data writer role on the underlying SQL databases.

Now let's focus on more specific details. The most important one is that you must have an **SSAS tabular** model created. Without an Analysis Services Tabular semantic model, write back on measures cannot be done.

The followings conditions must meet regarding your SSAS Tabular Model:

- **Power Query** data sources for tables are supported but only **without ANY transformation**. If you have an ETL-like logic in your table definitions leveraging Power Query:
    - consider moving that logic back to the SQL Server level (with writable views)
    - or make the transformation in your data flow (SSIS) so that you have fully prepared data in your data warehouse
    - or you can augment your model and can create separate tables for write back.
    - Best practice is to use native SQL query definitions for your SSAS entities.
- **Calculated tables** cannot be used for or during the write back. You can have calculated tables in your model, but you cannot use that table in the DEMX visual (neither on rows, nor on columns), and you cannot use that calculated table as a filtering element (like slicers, page/report level filters). If you have a calculated table in your model, you will need to turn of strict model structure checking – described in Models with calculated tables and columns chapter – otherwise you will receive an error. Best practice is to create a separate perspective in your model where you have only the supported entities.
- **Calculated columns** cannot be used for or during write back. You can have calculated columns in your model, but you cannot write back on them, neither you can use these columns as dimension members on the DEMX's rows or columns.
- **Relationships** based on calculated columns are not supported. The relationships defined in your SSAS model must be able to be mapped to SQL tables and join operators. Every column which is part of a relationship cannot be hidden in the model.
- Write back on **simple and additive measures** are supported by default, these are:
    - SUM(), AVERAGE()
    - And if only one column is used in the expression: SUMX(), AVERAGEX()
    - Datatype of numeric column which is included in a measure must be float
- To support write back on **complex measures** you need to fill in the **Description** field of your measure – in Visual Studio or Tabular Editor – that will tell the service which table and column it should target when committing the operation.
- **Complex measures** which:
    - Are containing multiple different columns in the expressions
    - Are modifying the filter context (CALCULATE, FILTER, ALL, etc)
    - Are using working tables (CALCULATETABLE, ADDCOLUMNS, SUMMARIZE, etc)
- Example measure: SUMX(Sales; Sales[Quantity] * Sales[UnitPrice])
    - In this case if you want the Quantity field to be changed put Quantity in the description of your measure.

Limitations

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

- **Renaming 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 SSAS models, the names should be equal to the names of the entities (visible in your model or in your report Fields sections).
- The characters '(' and ')' (**parenthesis**) are **not supported** in measure names
- The characters '[' and ']' (**brackets**) are **not supported** in measure names
- The character '.' (**dot**) is **not supported in table names**, but they can exist in schema names. If you have unsupported characters in your table name, we recommend creating a view on top over 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 5000 records can be displayed at one time. In case of the DEMX visual this means **5000 cells**. If the product of the number of the rows and columns (dimension members) used in your matrix exceeds this, consider using slicers in your report so that you will edit only a subset of the data and have a seamless workflow.
- **DAX Expression COUNTROWS(Table)** is not supported for write back.
- Non-writable **views** which serve as a data source for the fact table are not supported for write back. Either materialize your view into a physical table, reduce the complexity of the view or you can create INSTEAD OF INSERT | UPDATE triggers to handle the operation. Typically, non-writeable views are that contains complex SQL queries with multiple joins, CASE statements in WHERE clauses, CTEs (Common Table Expressions), aggregations. You can test your view by duplicating it under a different name and executing an INSERT statement against it.
- In case of on-premises **Power BI Report Server** at least January 2019 version is required together with Power BI Desktop 2019 January or more recent versions.
- If you are using **slicers**, page-report-or visual level filters you must use the **SmartFilter** helper visual as well. This is described in the Importance of SmartFilter chapter

Setup and configuration of the visual

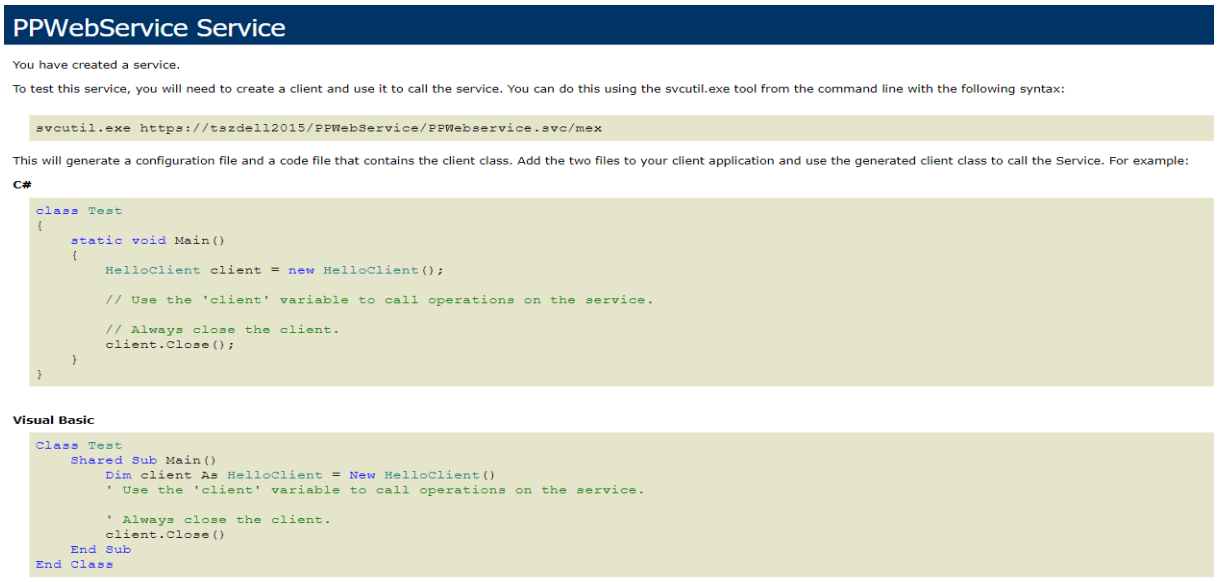
This main section describes the available configuration options for the Data Entry Matrix. Also, the following pages describes a **short** step by step instructions for building a **simple** report using DEMX. You will find the details of the configuration elements later in this document.

Validate the write back service

After installation you might want to be sure that the write back service is installed properly. You can validate that by navigating to your deployed url, which is in the following format:

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

If you see the following page after the page load, then 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#

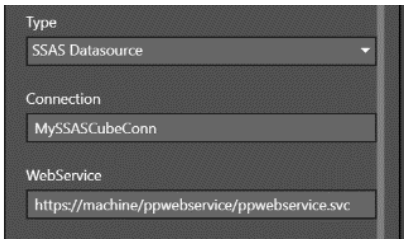
```
class Test
{
    static void Main()
    {
        HelloClient client = new HelloClient();

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

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

Visual Basic

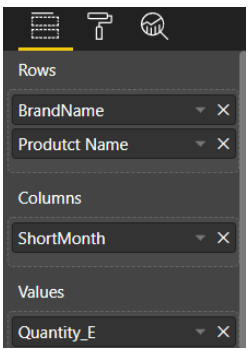
```
Class Test
    Shared Sub Main()
        Dim client As HelloClient = New HelloClient()
        ' Use the 'client' variable to call operations on the service.

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

Setting up the basics

After you launched Power BI Desktop, connected to a data source and imported the visual into your report, the first step is to tell the visual where can it find the write back service and which connection should it use during the process. The configuration options will appear in the Format section of the Visualizations **after** you dropped in the first field to the visual. You must complete the following steps in order to have a working report for the write back functionality.



Starting with Data Entry, you must set the **WebService**, **Connection** and **Type** properties (see referring chapters) to make the write back service working.



Last but not least please make sure you use the matrix properly, meaning adding measure(s) to the Values and multiple dimensions to Rows and Columns. A table supports two dimensions; however, matrix makes easier to display data across multiple dimensions. Dimensions are usually text or date(time) type of values, the measures are numeric values.

If you add a numeric type of column to the Rows/ Columns, you will not be able to modify and save the changes in the Data Entry Matrix.

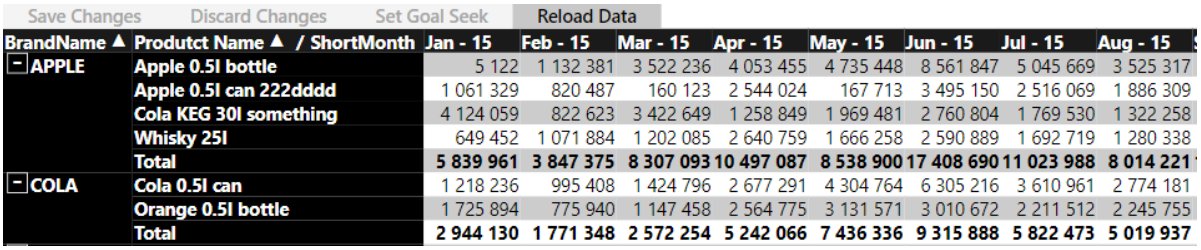
Add dimensions and measures

Simply add some fields into the Rows and Columns sections of the visual and add a write back compatible measure.

If you want to add some slicers, please refer to the Importance of SmartFilter Helper visual chapter.

Applying style

Navigate to the Column Headers and Row headers configuration and set a desired font size and color. You can create different theme for the totals and subtotals if you go to the related configuration section. For alternating colors, go the Values configuration and set different background colors for the Primary and secondary parameters. You might come up with a result below.



Save Changes | Discard Changes | Set Goal Seek | Reload Data

BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15	Mar - 15	Apr - 15	May - 15	Jun - 15	Jul - 15	Aug - 15
⊟APPLE	Apple 0.5l bottle	5 122	1 132 381	3 522 236	4 053 455	4 735 448	8 561 847	5 045 669	3 525 317
	Apple 0.5l can 222dddd	1 061 329	820 487	160 123	2 544 024	167 713	3 495 150	2 516 069	1 886 309
	Cola KEG 30l something	4 124 059	822 623	3 422 649	1 258 849	1 969 481	2 760 804	1 769 530	1 322 258
	Whisky 25l	649 452	1 071 884	1 202 085	2 640 759	1 666 258	2 590 889	1 692 719	1 280 338
	Total	**5 839 961**	**3 847 375**	**8 307 093**	**10 497 087**	**8 538 900**	**17 408 690**	**11 023 988**	**8 014 221**
⊟COLA	Cola 0.5l can	1 218 236	995 408	1 424 796	2 677 291	4 304 764	6 305 216	3 610 961	2 774 181
	Orange 0.5l bottle	1 725 894	775 940	1 147 458	2 564 775	3 131 571	3 010 672	2 211 512	2 245 755
	Total	**2 944 130**	**1 771 348**	**2 572 254**	**5 242 066**	**7 436 336**	**9 315 888**	**5 822 473**	**5 019 937**

Publishing and testing the report

When you finished with your report, publish it to either your on-premises Power BI Report server, or to PowerBI.com. The **write back will only work** reliable when your report is **published**, so you cannot test the full functionality in Power BI Desktop while you are authoring it. After publishing your report try to modify a cell value and click on save changes. Your modification should be visible in your report.

Completing the above steps, you should have a working, simple, write back capable report using Data Entry Matrix visual.

The following pages cover more detailed configuration options as well as typical use-cases that you might find useful for your needs.

Configuration

This chapter contains detailed description of the available configuration options for the DEMX visual.

Data Entry

This section contains the configurations regarding the connectivity with the write back service and some basic behavior of the DEMX visual.

Direct Query

Legacy option, the type of the SSAS model (if present) is determined automatically and will be removed in future releases.

Type

This option is referring to the connection type that the write back service will use. The setting is required.

- *SQL*: Set the connection string directly in the Connection property (leave the connection blank if using the default "**SQLConnection**" in set **in the web.config** of the write back service)
- *SQL Datasource*: The connection is set in the PPWebService web.config and you will have to refer it by its' name in the Connection property. Note: without an SSAS semantic model on top of your SQL database, SQL connections are only supported, if the relationships defined in your PowerBI report exists in your database with foreign key constraints, and only simple measures are used.
- *SSAS*: Set the connection string directly in the Connection property (leave the connection blank if using the default "SSASConnection" in set in the web.config of the webservice)
- *SSAS Datasource*: Set a connection name in the Connection property defined in the web.config of the webservice

Connection

Specify Name of the SQL or SSAS connection defined in the web.config file of the webservice, or the connection string directly, or **leave blank** depending on the Type setting.

WebService

Here you must set the write back service url 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.

Customer

Customer name provided by us along with the License server license key. If already specified in the web.config file (generally, you don't have to set this at it is done by the setup), **leave blank.**

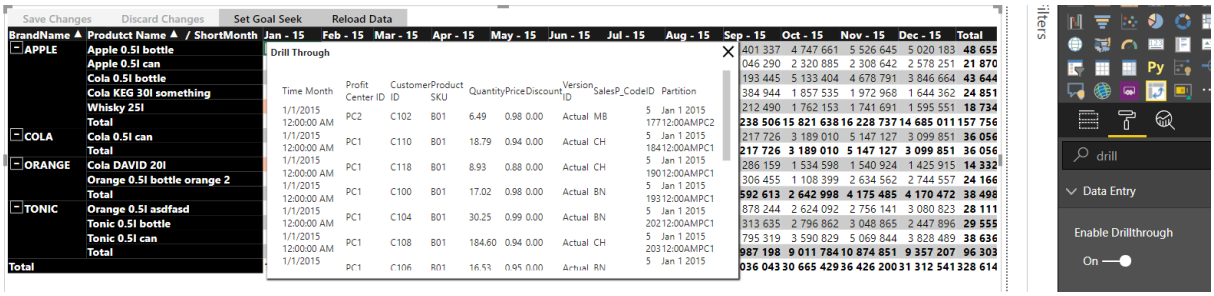
Domain

Set it to one of the following. Generally, this should **be left blank** as it is configured in the write back service web.config file during setup.

- the fixed the name of the internal domain used (e.g. DOMAIN)
- *auto*: use the domain part (part after @) of the powerbi.com user
- *auto-short*: use the short domain part (part after @ and before the .) of the powerbi.com user
- *azure*: use the full powerbi.com username (e.g. user@domain.com) for Azure AD authentication
- leave empty for no domain (e.g. for a SQL authentication user)
- *SQL=…,SSAS=…* set domain options for SQL and SSAS separately
- Custom user mapping of powerbi.com users to SQL and SSAS users can be set up in the UserMapping table in the SQL database (should contain User, SQLUser, SSASUser varchar columns) For more information please visit: https://support.poweronbi.com/portal/kb/articles/custom-user-mapping-when-using-powerbi-com-service

Enable drillthrough

If you enable this setting, by double clicking on a cell a popup will be rendered in which the related records of the fact table will be shown. Those rows will be displayed that are in the intersection of the dimensions determined by the cell. For example if you clicked on the cell of January for Apple 0.5l bottle product, an SQL query will be executed against the underlying database with a WHERE predicate containing the aforementioned dimension members, so the result will be those rows from the Sales fact table that belong to January and the Apple product. Note that this query will be executed on the SQL database even if you have an in-memory SSAS tabular model.



Support renaming

Obsolete, will be removed.

Input culture

If end users' browsers' or computer's culture or regional setting is **different** than the SSAS model property, it might happen that the numeric figures entered will not be saved correctly. It can happen when the decimal separator in the end user's machine is different (due to regional settings) then the model's setting. Generally, you will not need to set this, but if you experience issues during write back that is related to formatting, set the input culture here to match with the SSAS model's setting. Best practice is to set it to en-US in this case.

Read only

If turned on the matrix will become read only so end users will not be able to change values and commit write back.

Highlight enabled

Obsolete, will be removed.

Apply write security measures to totals

If enabled, the write security measure - used to control the writability of the values - will be applied for the totals as well. Make sure that the DAX formula used to check the conditions for cells can handle sub totals and totals as well-.

Initial refresh

Only affects how the data is retrieved when the visual is first rendered on the report page. If turned off the matrix will use the PowerBI cache –if available – to retrieve data. If turned on, the matrix will reach out to the SSAS model to fetch the most up to date data. The background is that PowerBI caches cell data for faster report rendering. It is mostly utilized when you have multiple report pages and you navigate between them. Therefore by default the DEMX has this setting enabled, so that when the visual rendered it will always connect to the underlying SSAS model to get data, and it will not use the cache. This is for concurrent workloads, as it might happen that after you made changes someone else also did, and if you navigated away and back, you might not see the change that was committed by the other user.

Windows Authentication

If you are in an on-premises environment using Power BI Report Server this setting has to be turned on, so that the Windows credentials can be passed back to the underlying data source. If you are completely in the cloud - your reports are published to PowerBI.com service - and you are using Azure AD, the setting should be turned off, so that AD credentials will be used when accessing the data source.

In case of Gateway by turning on Windows Authentication will have the following effects: Instead of setting the PowerBI.com credentials in the write back service request, the visual posts the windows login context. This value (e.g domain\user instead of username@domain.com) will be set if you use USERNAME() in computed and/or default value columns (see later in this document) as well in SQL context variables. Also, this makes possible to use impersonation, and it is **necessary to be turned on if Windows Authentication** is the required option set for authentication in **IIS** for the write back service.

Column Formatting

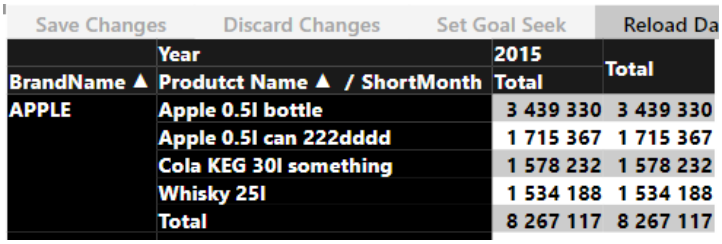
With these settings you can control the behavior and appearance of the columns in the visual.

Column alignment

You can set the text alignment for each column header with this setting.

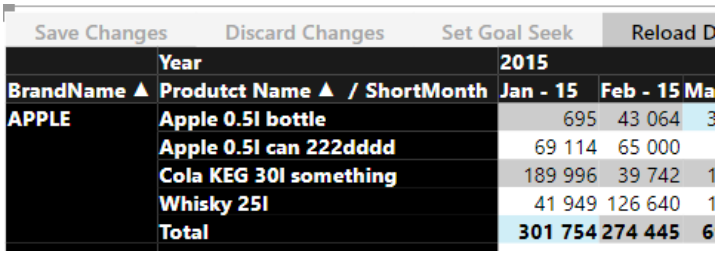
Column collapse

If you have multiple fields added to the column collection, you can turn on collapse on the top level one so that the matrix will be rendered as the group is collapsed by default.



	Year	2015	Total
BrandName ▲	Produtct Name ▲ / ShortMonth	Total	
APPLE	Apple 0.5l bottle	3 439 330	3 439 330
	Apple 0.5l can 222dddd	1 715 367	1 715 367
	Cola KEG 30l something	1 578 232	1 578 232
	Whisky 25l	1 534 188	1 534 188
	Total	8 267 117	8 267 117

If collapse is turned off, the visual will be rendered as all dimension members are expanded by default



	Year	2015		
BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15	Ma
APPLE	Apple 0.5l bottle	695	43 064	3
	Apple 0.5l can 222dddd	69 114	65 000	
	Cola KEG 30l something	189 996	39 742	1
	Whisky 25l	41 949	126 640	1
	Total	301 754	274 445	6

You can control how collapse is working by configuring the General -> Collapse method setting.

Column headers

You can customize the appearance of the column headers within this setting group, such as:

- Font and background colors
- Font Family and size
- Turn on or off the word wrap for longer strings
- Outline: obsolete, will be removed

Currently you cannot specify width individually for each of the columns. You can set the width for all the columns or you can turn on auto column width as well in the General settings.

Column sorting

You can control the ordering of the dimension members to be ascending or descending. Please also check the Sorting section of the Use Cases.

Column totals

Rendering column totals can be turned on or off within this setting group.

Row Formatting

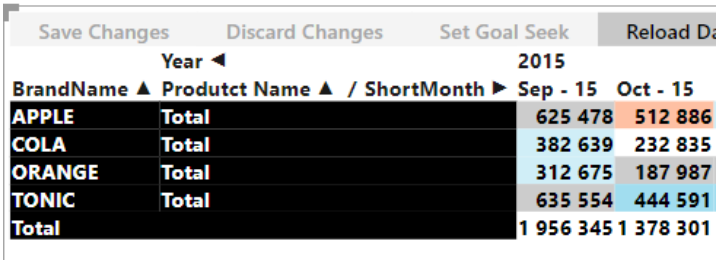
With these settings you can control the behavior and appearance of the rows in the visual.

Row alignment

You can set the text alignment for each row header with this setting.

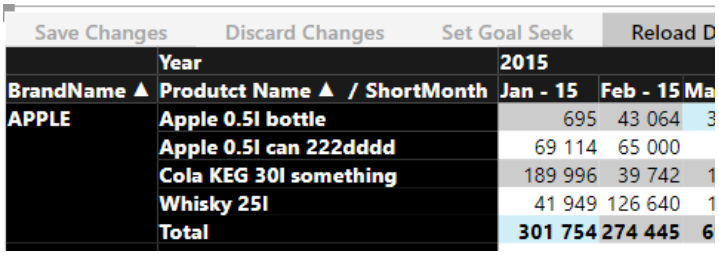
Row collapse

If you have multiple fields added to the row collection, you can turn on collapse on the top level one so that the matrix will be rendered as the group is collapsed by default.



	Year ◄	2015	
BrandName ▲	Produtct Name ▲ / ShortMonth ►	Sep - 15	Oct - 15
APPLE	Total	625 478	512 886
COLA	Total	382 639	232 835
ORANGE	Total	312 675	187 987
TONIC	Total	635 554	444 591
Total		1 956 345	1 378 301

If collapse is turned off, the visual will be rendered as all dimension members are expanded by default



	Year	2015		
BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15	Ma
APPLE	Apple 0.5l bottle	695	43 064	3
	Apple 0.5l can 222dddd	69 114	65 000	
	Cola KEG 30l something	189 996	39 742	1
	Whisky 25l	41 949	126 640	1
	Total	301 754	274 445	6

You can control how collapse is working by configuring the General -> Collapse method setting.

Row headers

You can customize the appearance of the row headers within this setting group, such as:

- Font and background colors
- Font Family and size
- Turn on or off the word wrap for longer strings
- Width of the row portion of the matrix. This can be very useful if you have members in the dimension that has a very long text
- Outline: obsolete, will be removed

Row sorting

You can control the ordering of the dimension members to be ascending or descending. Please also check the Sorting section of the Use Cases.

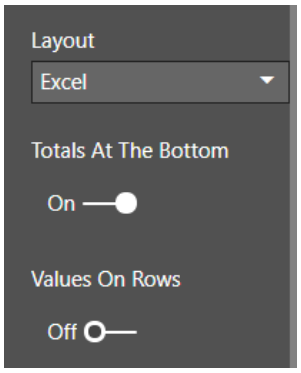
Row totals

Rendering row totals can be turned on or off within this setting group.

General settings

Under general settings you can control the behavior of the visual as well as how it is rendered.

Layout



By default, the DEMX is in the classic mode. If you change this to Excel, then the visual will be rendered differently as shown below and you will have to option to place the values on rows and show totals at the bottom. Turning on values on rows can be useful when you are using more measures in the matrix.

Classic rendering will result like this:

Save Changes | Discard Changes | Set Goal Seek | Reload Data

BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15	Mar - 15	Apr - 15	May - 15	Jun - 15
APPLE	Apple 0.5l bottle	695	43 064	348 711	428 818	25 000	589 654
	Apple 0.5l can 222dddd	69 114	65 000	18 695	241 333	25 000	234 749
	Cola KEG 30l something	189 996	39 742	176 932	128 520	125 000	188 375
	Whisky 25l	41 949	126 640	146 051	245 820	75 000	173 612
	Total	**301 754**	**274 445**	**690 389**	**1 044 491**	**250 000**	**1 186 389**

Excel rendering like this:

BrandName ▲ Produtct Name ▲	ShortMonth	Jan - 15	Feb - 15	Mar - 15	Apr - 15	May - 15	Jun - 15
APPLE							
Apple 0.5l bottle		695	43 064	348 711	428 818	25 000	589 654
Apple 0.5l can 222dddd		69 114	65 000	18 695	241 333	25 000	234 749
Cola KEG 30l something		189 996	39 742	176 932	128 520	125 000	188 375
Whisky 25l		41 949	126 640	146 051	245 820	75 000	173 612
Total		301 754	274 445	690 389	1 044 491	250 000	1 186 389

Format string

You can overwrite the measure formatting defined in the SSAS model by setting this property. If you have multiple measures this setting will influence all of it.

Auto size column width

If you turn this setting on, the visual will set the width for all your columns based on the widest one. If the setting is turned on, you can specify exactly how many pixels you want your columns to consume. This will influence all your columns. If turned on, consider using the stretch columns property so that columns will take up all the place that is available determined by the width of the matrix itself.

Horizontal cell padding

You can control how many pixels space you want to have between the columns with this setting.

Vertical cell padding

You can control how many pixels space you want to have between the rows with this setting.

Text size

You can define a general text size for the matrix. This will be overridden if you set up different sizes for your headers, values or totals respectively.

Auto-resize value threshold

When there are more value cells then the given one in this setting, the visual will use a faster but less sophisticated way to resize the rows or columns when a row or column is collapsed.

Collapse method

By default, the double click method is used which will require the end user to double click on the grouping member to expand. You can turn off this feature completely.

BrandName ▲ Produtct Name ▲	ShortMonth	Jan - 15	F
⊞ APPLE			
Total		301 754	2
⊟ COLA			
Cola 0.5l can		70 553	
Orange 0.5l bottle		135 394	
Total		205 947	

You can render a +/- icons in front of the members if you like, so that end users will need to click on those to collapse or expand the members.

Hover highlight

If enabled if you hoover your mouse over a cell, it will be highlighted by setting the cell's background color to black and the font color to white.

Stretch columns and rows

If your matrix has a width that is wider compared to how much space the total width of the rows or columns would consume, you can force the visual to stretch them to take all available space.

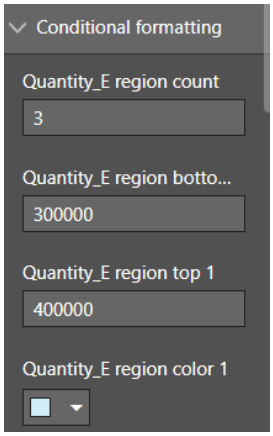
Selection border color and width

You can control the appearance of the selected cell by modifying the border color and width settings.

Position and size

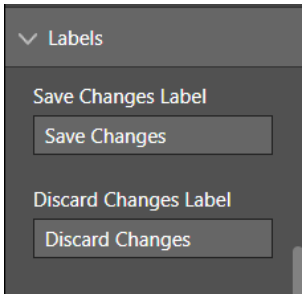
As for all visuals you can control the size of the matrix by setting height and width as well as the X and Y position. Setting the width and height have an effect on how stretching can be applied.

Conditional formatting



You can set up custom background colors for your measures depending on the values. First you need to define how many different colorings you want use by setting the region count property. If it set for 3, you will need to determine the boundaries for each by setting the bottom and top values. In this example, the first color – that is a light blue – will be applied as a background for those cells that have values between 300 000 and 400 000.

Labels



If you don't like the original text that are used for Save Changes, Discard changes and so on, you can change those by navigating to the Labels configuration and modifying the properties there.

Formatting Totals

In the Totals setting group you can control:

- Whether to show totals or not
- Show sub-and grand totals on rows and on the columns as well
- The primary and secondary font and background colors – for alternating
- Outline: obsolete, will be removed.

Formatting Values

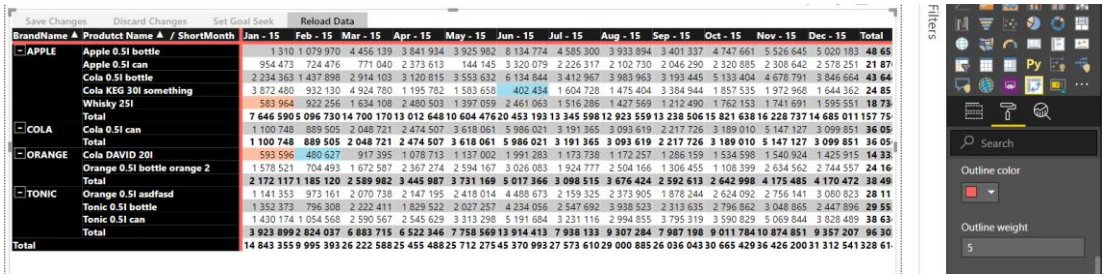
In the Values setting group you can control:

- Font family and size of the rendered measures values
- The primary and secondary font and background colors – for alternating
- The primary and secondary font and background colors for disabled (read only cells)
- The primary and secondary font and background colors for those cells where the values have been changed
- The maximum decimal places shown when you edit the cell value

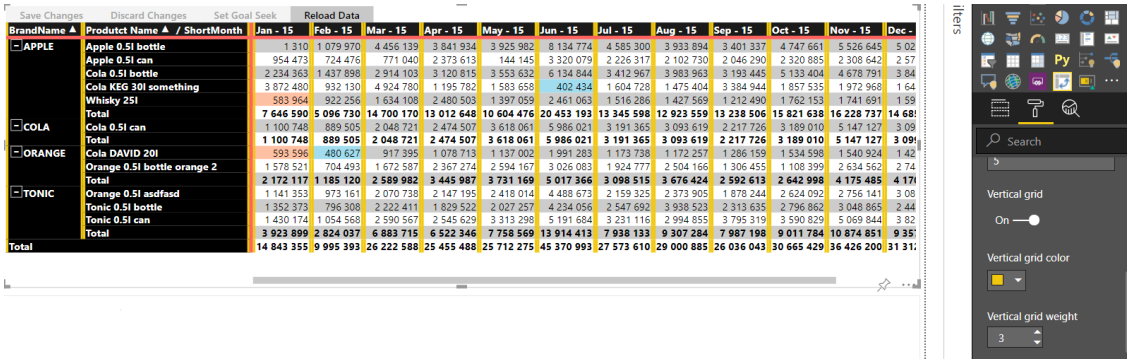
Grid settings

You can further customize the appearance of the matrix by specifying the parameters under this configuration collection.

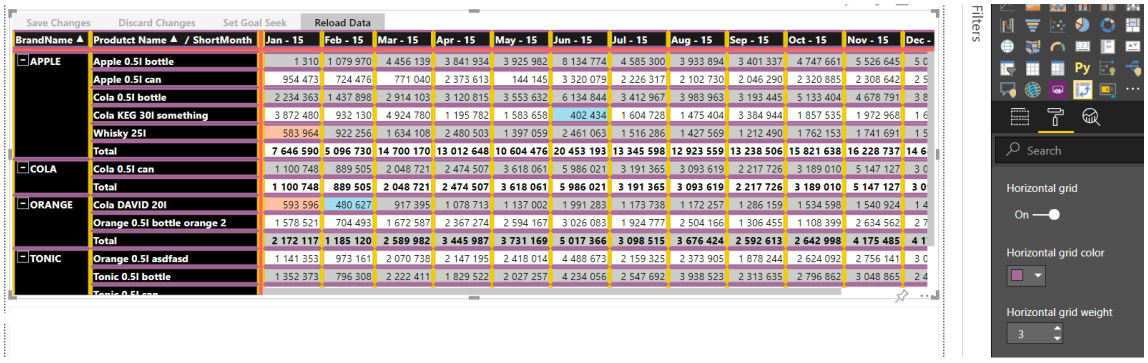
Setting the outline color and width will influence the borders between the row/column headers and the cells



Setting the vertical grid properties, you can modify the borders between columns



Horizontal grid settings will have an effect on the borders between rows



Importance of SmartFilter helper visual

It is a general functionality in PowerBI that when you are using **slicers**, page-report or visual level filters the visual itself is not aware that it is receiving a filtered dataset. For the writeback to work properly if you are using any of these filtering capable elements, you need **place SmartFilter** helper visuals in your report, so that the DEMX will be aware of these, and the write back engine will take these into consideration when it composes the SQL statements. Without it, slicer selection will be left out of the tuple that is being sent back to the service, it will be missing from the SQL statement, hence the saved result will be wrong. **It is important that you use the visual that is shipped with the setup kit**, as you can find a visual with the similar name in the Microsoft Visual Store but that does not have this functionality.

This visual is not-visible to the end users as it has no background no borders, and the values are not shown.

General rules:

If you have a slicer or filter for a field, you usually need to add a SmartFilter helper visual to the report with the same field, with the following **exceptions**:

- If the field or a lower granularity field from the same table is already on the field list of the writeback visual (e.g. If you have a slicer or filter for Product.Brand, and Product.Brand or Product.SKU is already in a SmartFilter or on the writeback visuals field list, you don't need to add a SmartFilter for it)
- If the field has lower granularity, then data in your writeback table. In this case you only need to add a SmartFilter with a field that corresponds to the granularity of the writeback data (e.g. if the granularity of your writeback data is based on Product.Brand, you don't need to add a SmartFilter for Product.SKU only for Product.Brand)
- If the field is already determined by the combination of one or more SmartFilters or writeback visual fields (e.g. If you have a slicer for Time.Quarter, and Time.Year and Time.Month is already in a SmartFilter or on the writeback visuals field list, you don't need to add a SmartFilter for it. Or if you have a slicer of filter for Account.Name, and Account.ID is already in a SmartFilter or on the writeback visuals field list, you don't need to add a SmartFilter for it.)

Performance wise it's better to use multiple SmartFilters with higher granularity fields, then one SmartFilter with a low granularity field. (i.e. If there are slicers or filters on Product.Brand and Product.Size, it's more performant to add SmartFilters for these instead of adding a SmartFilter for Product.SKU)

Smart formulas

You can invoke smart formulas on cells. Instead of typing exact values, you can use the following formulas:

- inc# : increases the value by the given # number
- inc%# : increases the value by the given # percentage
- dec# : decreases the value by the given # number
- dec%# : decreases the value by the given # percentage
- mul# : multiplies the value by the given # number
- div#: divides the value by the given # number
- <# : copies the entered # value all to the left to all columns in the given row

- #> : copies the entered # value all to the right to all columns in the given row
- v#: copies the entered # value until next total down in the given column
- ^#: copies the entered # value until next total up in the given column
- E# : only usable on total or sub totals. Will evenly distribute the # number between the detail rows in the aggregation group instead of proportional distribution. If there are different number of fact rows behind each cell, the cells will have different values after save.

Writeback on totals

The Data Entry Matrix has a special feature when it comes to writing back on totals or subtotals. Consider the following example where we have four products in the APPLE brand

Save Changes | Discard Changes | Set Goal Seek | Reload

BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15
APPLE	Apple 0.5l bottle	695	43 064
	Apple 0.5l can 222dddd	69 114	65 000
	Cola KEG 30l something	189 996	39 742
	Whisky 25l	41 949	126 640
	Total	301 754	274 445

As you can see the 274k total is a sum of the four products. There is a rate between these products in terms of how much they contribute to the total. Whisky gives the most, as 126 640 / 274 445 is 46% of the total, and Cola KEG contributes the least with its 14%. If you write back on a total or subtotal, by default the service will try to distribute the modified value honoring the original ratio between the products. When you change the subtotal value to 300 000 for example, 46% will be allocated for the Whisky product.

Tip: if you want to distribute the total value by a predetermined ratio here is a trick that you can apply. Before modifying the total, set a percentage figure for each of the elements inside that aggregation group like below and save changes

Save Changes | Discard Changes | Set Goal Seek | Reload

BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15
APPLE	Apple 0.5l bottle	695	10
	Apple 0.5l can 222dddd	69 114	10
	Cola KEG 30l something	189 996	50
	Whisky 25l	41 949	30
	Total	301 754	274 445

Then you can set a value for your total. In this case 10% will be allocated for Apple 0.5L bottle, and 30% for the Whisky product. Note: this technique will not work on complex measures.

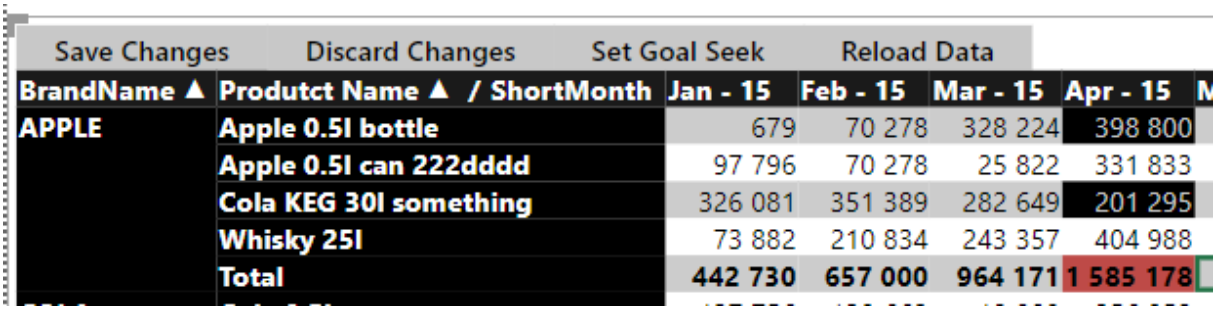
You can force the service to use even distribution by using the smart formula E#. If you enter E400000 in this example, 100 000 will be allocated for each of the products, as there are 4 items in this aggregation group.

Note: if you use a separate write back table with default values displayed if there is no written back record on that tuple, writing back on totals will only influence those cells where there are saved records for that tuple in your write back table.

Goal seek

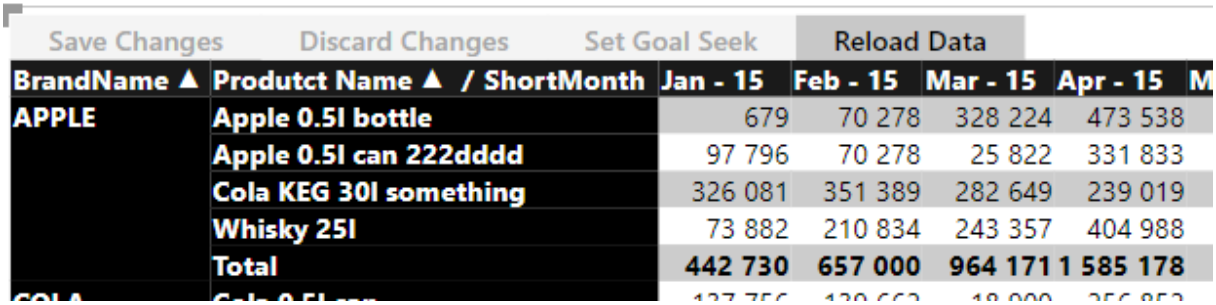
You can use the goal seek functionality to only modify selected cells to achieve a target goal.

Consider the example below, where we want to increase of total sales by 10% for the APPLE brand for April month, but by only modifying the Apple 0.5l bottle and Cola KEG products. You can do the by selecting these products cells and marking them for goal seek by clicking on Set Goal Seek followed by an inc10% smart formula applied on the sub total row. When you click on save changes, that 10% increase will be distributed amongst these two products.



Save Changes | Discard Changes | Set Goal Seek | Reload Data

BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15	Mar - 15	Apr - 15	M
APPLE	Apple 0.5l bottle	679	70 278	328 224	398 800	
	Apple 0.5l can 222dddd	97 796	70 278	25 822	331 833	
	Cola KEG 30l something	326 081	351 389	282 649	201 295	
	Whisky 25l	73 882	210 834	243 357	404 988	
	Total	442 730	657 000	964 171	1 585 178	



Save Changes | Discard Changes | Set Goal Seek | Reload Data

BrandName ▲	Produtct Name ▲ / ShortMonth	Jan - 15	Feb - 15	Mar - 15	Apr - 15	M
APPLE	Apple 0.5l bottle	679	70 278	328 224	473 538	
	Apple 0.5l can 222dddd	97 796	70 278	25 822	331 833	
	Cola KEG 30l something	326 081	351 389	282 649	239 019	
	Whisky 25l	73 882	210 834	243 357	404 988	
	Total	442 730	657 000	964 171	1 585 178	

You can turn of this functionality in the Goal seek settings collection. You can also set the coloring theme for the cells that are marked to participate in the operation.

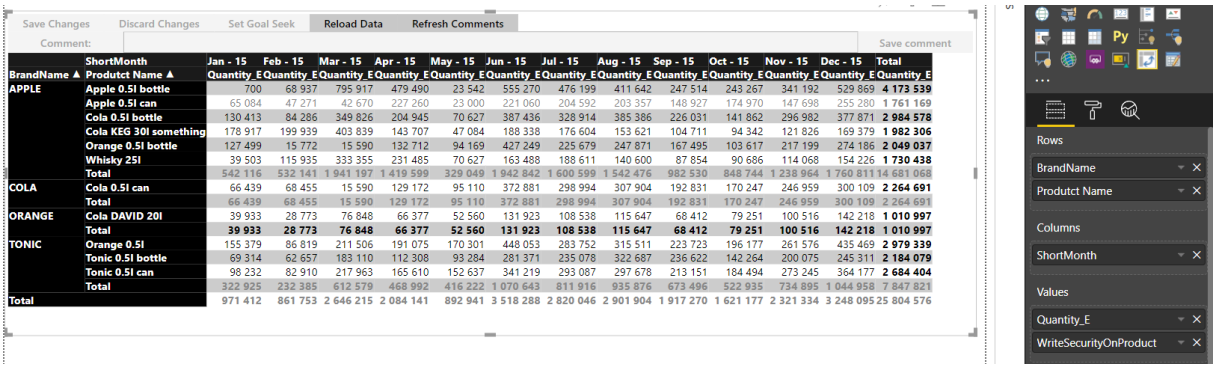
Important note: you can only use goal seek feature on those measures and cells which are in relation to each other and the operation is arithmetically possible. For example if you set goal seek on a cell which is not included in the measure of the target (the cell you modify) then the operation will not complete.

Securing cells

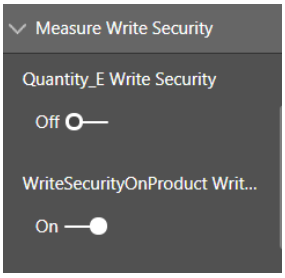
Measure Write Security

You can control which cells are editable and which not in the visual by creating a measure that will be evaluated on the cells, but its result will be solely used for this security purpose. This measure can be deployed into your SSAS model, or it can be an embedded measure in the report itself.

So at the end you will have two at least two measures in your matrix as below



BrandName	ShortMonth Productct Name	Jan - 15 Quantity_E	Feb - 15 Quantity_E	Mar - 15 Quantity_E	Apr - 15 Quantity_E	May - 15 Quantity_E	Jun - 15 Quantity_E	Jul - 15 Quantity_E	Aug - 15 Quantity_E	Sep - 15 Quantity_E	Oct - 15 Quantity_E	Nov - 15 Quantity_E	Dec - 15 Quantity_E	Total Quantity_E
APPLE	Apple 0.5l bottle	700	68 937	795 917	479 490	23 542	555 270	476 199	411 642	247 514	243 267	341 192	529 869	**4 173 539**
	Apple 0.5l can	65 084	47 271	42 670	227 260	23 000	221 060	204 592	203 357	148 927	174 970	147 698	255 280	**1 761 169**
	Cola 0.5l bottle	130 413	84 286	349 826	204 945	70 627	387 436	328 914	385 386	226 031	141 862	296 982	377 871	**2 984 578**
	Cola KEG 30l something	178 917	199 939	403 839	143 707	47 084	188 338	176 604	153 621	104 711	94 342	121 826	169 379	**1 982 306**
	Orange 0.5l bottle	127 499	15 772	15 590	132 712	94 169	427 249	225 679	247 871	167 495	103 617	217 199	274 186	**2 049 037**
	Whisky 25l	39 503	115 935	333 355	231 485	70 627	163 488	188 611	140 600	87 854	90 686	114 068	154 226	**1 730 438**
	Total	542 116	532 141	1 941 197	1 419 599	329 049	1 942 842	1 600 599	1 542 476	982 530	848 744	1 238 964	1 760 811	14 681 068
COLA	Cola 0.5l can	66 439	68 455	15 590	129 172	95 110	372 881	298 994	307 904	192 831	170 247	246 959	300 109	**2 264 691**
	Total	66 439	68 455	15 590	129 172	95 110	372 881	298 994	307 904	192 831	170 247	246 959	300 109	2 264 691
ORANGE	Cola DAVID 20l	39 933	28 773	76 848	66 377	52 560	131 923	108 538	115 647	68 412	79 251	100 516	142 218	**1 010 997**
	Total	**39 933**	**28 773**	**76 848**	**66 377**	**52 560**	**131 923**	**108 538**	**115 647**	**68 412**	**79 251**	**100 516**	**142 218**	**1 010 997**
TONIC	Orange 0.5l	155 379	86 819	211 506	191 075	170 301	448 053	283 752	315 511	223 723	196 177	261 576	435 469	**2 979 339**
	Tonic 0.5l bottle	69 314	62 657	183 110	112 308	93 284	281 371	235 078	322 687	236 622	142 264	200 075	245 311	**2 184 079**
	Tonic 0.5l can	98 232	82 910	217 963	165 610	152 637	341 219	293 087	297 678	213 151	184 494	273 245	364 177	**2 684 404**
	Total	322 925	232 385	612 579	468 992	416 222	1 070 643	811 916	935 876	673 496	522 935	734 895	1 044 958	7 847 821
Total		971 412	861 753	2 646 215	2 084 141	892 941	3 518 288	2 820 046	2 901 904	1 917 270	1 621 177	2 321 334	3 248 095	25 804 576



The Quantity_E measure is shown and can be written back, and the result of the WriteSecurityOnProduct measure is used to control which cell is read only. If you turn the Write Security setting on for this measure, it will be hidden and will be used to determine the security.

The logic of this security measure should be the following:

If the DAX expression returns an empty string (“”) or BLANK() then the visual will consider this cell to be read only. (Be careful with BLANK() as the visual might filter out those rows, if the displayed measure also does not have a value. Look for show members with no values setting in your report. Best practice is to use empty string.). If it returns anything else – like 1 or TRUE() it will consider this cell to be writeable.

Consider this example:

```
CanWrite :=
MAXX (
    Sales;
    IF (
        AND (
            OR (
                SELECTEDVALUE ( 'Time'[ShortMonth] ) = "Mar - 15";
                SELECTEDVALUE ( 'Time'[ShortMonth] ) = "Jun - 15"
            );
            SELECTEDVALUE ( 'Product'[SKU] ) = "H02"
        );
        "";
        1
    )
)
```

It will prohibit modification on cells that belong to the H02 Product and contains data for March and June in 2015.

You can implement more complex securities with this technique such as control permissions by user – entity mapping (which user can edit which product). Please refer to the following article that contains a detailed example on how to implement such a use case:

https://support.poweronbi.com/portal/kb/articles/implement-complex-write-security-per-entity

Read Only Values

You can completely turn of write back capability on a given measure if you turn this setting on for it.

SQL Row Level Security Policies

You can implement RLS policies on your fact tables to further tighten the security. Please refer to the following article in our Knowledge Base:

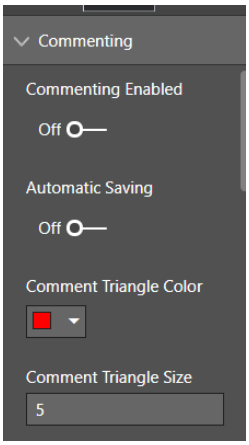
https://support.poweronbi.com/portal/kb/articles/control-write-back-permissions-on-back-end-sql-server

Commenting

You can turn on the commenting feature for your SSAS model. If enabled, you can create comments on each cells that will be saved to the underlying SQL database. Comment records will contain:

- Tuple (on which cell the comment was created)
- Date
- User
- Text

This comment table will be created by the write back service inside your SQL database with the name dbo.Comments. You can configure the following settings:



Automatic Saving: If enabled, comments are automatically saved after changes.
Comment Triangle Color: The color of the comment symbol appearing on cells.
Comment Triangle Size: Size of triangle in pixels
Auto Refresh Comments: If enabled, comments automatically refresh without the need of pressing the 'Refresh Comments' button.
Show comments on cell select: If enabled when a cell selected and there is comment, the popup will automatically be rendered.
Show Date: If enabled, displays the date of the comment's creation/last update.

By default, the last comment will be persisted on a cell if you overwrite it. However, by leveraging SQL triggers you can preserve the comment history. Please refer to this article on how can you implement it:
https://support.poweronbi.com/portal/kb/articles/preserve-comments-in-the-same-cell-aka-saving-comments-history

Use cases

This chapter describes typical uses cases that you can implement in your environment to enrich Table Editor feature leveraging native database objects. Chapter references Microsoft's SQL Server's features.

Versioning

Please refer to the following articles in our knowledge base regarding versioning:

- https://support.poweronbi.com/portal/kb/articles/implement-versioning
- https://support.poweronbi.com/portal/kb/articles/implement-version-management

Custom validation

You can create complex validation logics by leveraging SQL server features, more precisely **triggers** to check certain conditions and send back message to the client in case of violation of a business rule. The core logic of the trigger can contain anything you like, only the return method of your message needs to be in a certain format. The following example shows how to return a custom message in a trigger:

```
IF (@YourCondition)
   THROW 50001, N'<SQLError>Cannot save modifications due to violation of business
rules.</SQLError>', 1
```

Auditing

This chapter will provide you a guide how to capture the username who is modifying the current measure. Please refer to the following article in our Knowledge Base:

https://support.poweronbi.com/portal/kb/articles/get-user-name-on-back-end-during-write-back-sql-server

For example, if you modify a cell that is an aggregation of multiple records of the Fact table, you can create a FactAudit table with the same structure as the table in question with columns that can contain extra information about the operation, like:

- ModifiedBy – the user's name
- ModifiedAt – the date
- OldValue
- NewValue

The following trigger will capture the actual user name and the current date, and saves it to an audit table:

```
CREATE TRIGGER [dbo].[trg_FactAudit] on [dbo].[Fact]
AFTER INSERT, UPDATE
AS
BEGIN

	DECLARE @USERNAME VARCHAR(255)

	SET  @USERNAME = CAST(SESSION_CONTEXT(N'user_name') as varchar(255))

	INSERT INTO [dbo].[ProductAudit]
	SELECT
	[SKU]
	,@USERNAME
	,GETDATE()
	,yourFactTableColumns
	FROM INSERTED

END
```

Please note, that writing back to a measure can have an effect on multiple rows in the underlying fact table. If you implement this solution your audit table can grow quickly, depending on the granularity of your fact table. Consider using the Tracing feature or enabling the SaveWriteBackHistory setting as described in the Advanced Settings chapter

Performance optimization

In case of large fact tables reprocessing in-memory SSAS models can take some time. The write back service can determine which portion of your data should be processed. For that you can implement special partitioning. Please refer to the following article in our Knowledge Base for examples, walk throughs and tips:

- https://support.poweronbi.com/portal/kb/articles/performance-tips-for-direct-query-mode
- https://support.poweronbi.com/portal/kb/articles/performance-optimization-tips
- https://support.poweronbi.com/portal/kb/articles/implement-ssas-partitioning-for-improved-write-back-performance
- https://support.poweronbi.com/portal/kb/articles/dedicated-table-for-write-back-using-ssas-partitioning

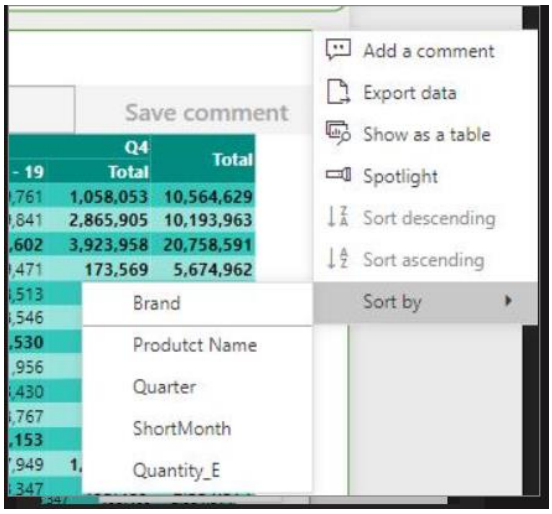
Models with calculated tables and columns

As described in the Limitations chapter, you cannot write back on calculated tables and columns. You can have these types of objects inside your SSAS model, but by default the write back service always do a model metadata check before saving changes, and if it finds such structures will prohibit the process. If you are certain that you are not trying to write back on measures that are using calculated columns, and you are not using calculated tables in your matrix (neither on rows nor on columns), also these objects are not used in any kind of filters (slicers, visual level filters, etc), you can turn off this strict checking by using Advanced Configurations.

Sorting

If you do not achieve the desired result when it comes to the built-in sorting functionality, you can still try out the followings:

You can use PowerBI built in sorting functionality:



Or you can set up a Sort By Column parameter in your SSAS model in Visual Studio or Tabular Editor.

Advanced configuration

You can control the behavior of the write back service for each SSAS model. For this you need to create a dedicated SQL table in a strict format. It is important to know, that if you have multiple data sources in your SSAS model, you need to tell the write back engine that in which data source should it look for this setting table. The way to do it is to add the WriteBack – case sensitive string in the name of your data source where you will be creating this helper table. So for instance if you have three data sources defined (ds_A, ds_B and ds_C) for three different SQL databases (Database_1, Database_2, Database_3) and you want to create this settings table in Database_2 – which is used in ds_B, then rename it to be ds_B_WriteBack for instance.

The structure of the table is:

```
CREATE TABLE [dbo].[AdvancedSettings]
(
    [Setting] [varchar](255) NOT NULL,
    [Value] [sql_variant] NULL,
CONSTRAINT [pk_AdvancedSettings] PRIMARY KEY CLUSTERED( [Setting] ASC )
)
```

You cannot rename the table or use other schema then dbo at the moment. Be aware, that the [Value] column is an sql_variant datatype, so different settings will have different types (like bit, varchar, numeric, etc) It means that when you try to copy this table by the Generate Script tasks in SQL Management Studio it will not recognize the correct data types and might result in a non-compatible setting. If you need to move/copy this table – or the entire database – always re-create this helper table with the appropriate data types by using the correct insert statements.

Below you find the most important settings, for the complete list please refer to this article:

https://support.poweronbi.com/portal/kb/articles/advanced-configuration

Important settings:

1. Turn off strict metadata checking: in case of you have calculated tables, relationships by default the service will prohibit the operation. If you turn this off, the operation will not be blocked. Be careful that you will not use unsupported objects during the write back (for example you do not use calculated column as a row / column member in your matrix or as slicers, etc)
2. Enable tracing: if you enable this a Trace.Txt file be created in the folder at the location of the webservice which will log all operations during write back.
3. Save write back history: if you enable this, a dbo.WriteBackHistory table will be created in your database (in case of multiple data sources, it will use the connection with the WriteBack tag) that will contain the following information for each cell modification:
    a. ChangeDate – time of change
    b. ChangedBy – user name who committed the change
    c. Tuple – the cell's tuple or coordinate
    d. OldValue – original value of the cell
    e. NewValue the updated value of the cell

For point 1:

```
INSERT [dbo].[AdvancedSettings] ([Setting], [Value]) VALUES (N'SkipInvalidMeasures',
CAST(1 as bit)) --Don't throw error on non-parsable DAX expressions
INSERT [dbo].[AdvancedSettings] ([Setting], [Value]) VALUES
(N'SkipCalculatedRelationships', CAST(1 as bit)) --Don't throw error if calculated column
used in relation
```

For point 2:

```
INSERT [dbo].[AdvancedSettings] ([Setting], [Value]) VALUES (N'EnableTracing', CAST(1 as
bit)) -- creates Trace.txt inside the web service's folder and log all operations
```

For point 3:

```
INSERT [dbo].[AdvancedSettings] ([Setting], [Value]) VALUES (N'SaveWritebackHistory',
CAST(1 as bit)) -- creates dbo.WritebackHistory table and logs cell changes
```

Troubleshooting

For all cases, please visit our Knowledge base at: https://support.poweronbi.com/ to find a solution for your problem. Below you can find common cases. If your issue cannot be solved by the provided materials, please open a ticket on our support site, and PowerOn will assist you.

Network Error

Symptom: You receive Network Error message when you try to save data.

Cause: You may get this error typically in the following cases:

- 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 was specified
- the license service is stopped unexpectedly
- in case of on-premises or Gateway installations the Windows Authentication setting is turned accordingly in the visual under Data Entry based on the IIS authentication settings
- 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: Make sure that the write back service is up and running and reachable (firewall not interfering) as it is described earlier in this document. Verify that the referenced connection exists in the write back service configuration and the connection type is selected correctly.

A particular error can indicate a license service failure. Please refer to this article:

https://support.poweronbi.com/portal/kb/articles/error-the-communication-object-system-servicemodel-channels-servicechannel-cannot-be-used-for-communication-because-it-is-in-the-faulted-state

If you encounter CORS issue, please check the web.config file of the write back service for typo-s, and also missing dll-s and config files inside the web service folder. As a last resort, try updating the web service file following these articles:

- https://support.poweronbi.com/portal/kb/articles/how-to-update-the-service-manually-azure-cloud
- https://support.poweronbi.com/portal/kb/articles/how-to-update

If you encounter SSL error, make sure that the certificate is issued by a trusted authority for the full qualified domain name of the IIS server, or that the certificate is set to ignore by the client's browser in case of self-signed certificate, or if the certificate is issued internally by your organization and you try to reach the report outside of the organization domain.

If you are using Power BI Reporting Services on-premise, make sure that the report server url and the web service url format matches. Either both have to reference the machine name or the full qualified domain name, and both have to be http or https. Also, if Windows Authentication is set in the IIS configuration make sure that Windows Authentication is enabled in your visual under Data Entry settings.

In case of on-premises installation in a domain with Windows Authentication, make sure that Service Principal Names are created for your SQL and SSAS servers and the correct domain users are used for the services. An SPN is also needed for the service account that is running the PPWebService on the IIS machine as the Application Pool user. Make sure that delegation is enabled in your Active Directory from the App Pool's user (which has to be a trusted user) to the SQL/SSAS services. If these are not set correctly, Kerberos authentication issues can occur, which might result in HTTP 403 or 404 errors or appearing log-in popups. Please refer to the following article and contact your internal IT team: https://support.poweronbi.com/portal/kb/articles/configure-iis-for-kerberos-authentication

The visual is not working in Power BI Desktop or the settings are not shown.

Symptom: The visual is not rendered, or Data Entry Matrix specific setting are not displayed (like Data Entry, General, etc)

Cause: Generally, there can be two reasons for this:

- your machine is running out of memory, therefore Power BI Desktop cannot render the elements properly
- your Power BI Desktop Cache is outdated.

Solution: Free up memory on your computer by closing other applications. For clearing the Power BI Desktop cache, please refer to the article in our Knowledge Base at:

https://support.poweronbi.com/portal/kb/articles/power-bi-desktop-clear-cache

Save Failed

Symptom: You receive Save Failed message when you try to write back to the selected table.

Cause: You may get this error if there is a configuration error in your visual, the write back service is not configured properly, or in the following cases:

- there is 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

Wrong figures were saved

Symptom: Writeback was successful, but bad number is shown after the visual is refreshed.

Cause: You may get this error if:

- You are using slicers page-report or visual level filters and you forgot to add SmartFilter helper visuals properly
- You have an issue with the relationships in your SSAS model
- You have data quality issues in your underlying database, like duplicate members, incompatible data types, etc
- You have a logic error in your DAX measure formula

Solution:

- Add SmartFilter helper visuals
- Check your SSAS model relationships
- Verify your data integrity in your underlying SQL database
- Check your DAX measure

As there can be various reasons for this error, review the paragraphs in this chapter and please also visit our Knowledge Base to find the solution at: https://support.poweronbi.com

If you were not able to overcome your issue, please submit a ticket on our support site or write as at vizsupport@poweronbi.com and we will contact you shortly to help you investigate and fix the problem.