

Power BI – Model Documenter

#PowerBler



DaxStudio - 2.5.0

File **Home** **Help**

Run **Cancel** **Clear Cache** **Output** **Edit** **DAX** **Format Query** **A To Upper** **Comment** **To Lower** **Uncomment** **Swap Delimiters** **Merge XML** **Find** **Query Plan** **Traces** **Scan** **Right Layout** **Cache** **Bottom Layout** **Internal** **Server Timings** **Connect** **Refresh Metadata**

Query1.dax*

```

10 EVALUATE
11 ADDCOLUMNS (
12     SUMMARIZE (
13         Sales,
14         'Date'[Calendar Year],
15         'Product'[Category Name] ),
16         "sales", [Sales Amount],
17         "Margin", [Margin]
18 )
  
```

Total SE CPU Line Subclass Duration CPU Rows KB Query

550 ms	0 ms	x0.0	1 SQL	301	0	SELECT TOP (1000001) [SELECT
			2 SQL	231	0	SELECT TOP (1000001) [TOP (1000001) [t0].[Calendar Year
						Name],SUM([t1].[Line Margin])	Name],SUM([t1].[Line Margin])]

FE SE

ALM Toolkit for Power BI

File **Home** **Help**

Compare **Select** **Validate** **Update** **Generate** **Script** **Options** **Report** **Differences**

Source: <https://southcentralus.azure.windows.net/chwade003/AdventureWorks2> Target: <https://southcentralus.azure.windows.net/chwade003/AdventureWorks3>

Metadata **Functions**

Ready

DimCustomer

Type	Source Name	Status	Target Name	Action
Data Source	SQLlocalhost(SPI)AdventureWorksDW	Different Definitions	SQLlocalhost(SPI)AdventureWorksDW	Update
Relationship	'DimCustomer'[GeographyKey]>DimGeography[GeographyKey]	Missing in Target		Skip
Table	RangeEnd	Missing in Target		Skip
Table	RangeStart	Same Definition	DimCustomer	Skip
Table	DimCustomer	Same Definition	'DimCustomer'[GeographyKey]>DimGeography[GeographyKey]	Skip
Table	DimDate	Same Definition	DimDate	Skip
Table	DaysCurrentQuarterToDate	Same Definition	DaysCurrentQuarterToDate	Skip
Table	DaysInCurrentQuarter	Same Definition	DaysInCurrentQuarter	Skip
Table	DimEmployee	Same Definition	DimEmployee	Skip
Table	DimGeography	Same Definition	DimGeography	Skip
Table	DimProduct	Same Definition	DimProduct	Skip
Table	DimProductSubcategory	Same Definition	'DimProduct'[ProductSubcategoryKey]>DimProductSubcategory[ProductSubcategoryKey]	Skip
Table	DimProductSubcategory	Same Definition	DimProductSubcategory	Skip
Table	DimProductSubcategory	Same Definition	'DimProductSubcategory'[ProductCategoryKey]>DimProductCategory[ProductCategoryKey]	Skip
Table	FactInternetSales	Same Definition	FactInternetSales	Skip
Table	FactInternetSales	Missing in Target	FactInternetSales[CustomerKey]>DimCustomer[CustomerKey]	Skip
Table	FactInternetSales	Missing in Target	FactInternetSales[DueDateKey]>DimDate[DateKey]	Skip
Table	FactInternetSales	Missing in Target	FactInternetSales[OrderDateKey]>DimDate[DateKey]	Skip
Table	FactInternetSales	Missing in Target	FactInternetSales[ProductKey]>DimProduct[ProductKey]	Skip
Table	FactInternetSales	Missing in Target	FactInternetSales[ShipDateKey]>DimDate[DateKey]	Skip
Table	InternetCurrentQuarterMargin	Missing in Target		Skip
Table	InternetCurrentQuarterSales	Missing in Target		Skip
Table	InternetDistinctCountSalesOrder	Missing in Target		Skip
Table	InternetOrderLinesCount	Missing in Target		Skip
Table	InternetPreviousQuarterMargin	Missing in Target		Skip

```

1 {
2   "name": "DimProduct",
3   "columns": [
4     {
5       "name": "Class",
6       "dataType": "string",
7       "sourceColumn": "Class"
8     },
  
```

```

1 {
2   "name": "DimProduct",
3   "columns": [
4     {
5       "name": "Class",
6       "dataType": "string",
7       "sourceColumn": "Class"
8     },
  
```

C:\Projects\AdventureWorks\Model.bim - Tabular Editor 2.1

File **Edit** **View** **Model**

Perspective: (All objects) Translation: (No translation)

DAX Editor **Advanced Scripting**

```

[Reseller Current Quarter Sales] :=
TOTALQTD([Reseller Total Sales], 'Date'[Date])
  
```

Basic

- Description: Sales
- Display Folder: False
- Hidden: Reseller Current Quarter Sales
- Name: Measure

Metadata

- DAX identifier: [Reseller Current Quarter Sales]
- Format String: '\$#.00;(\$#.00);\\$#.00'

Options

Other

Translations and Perspectives

- Captions: 0 empty, 0 translated, 0 default
- Descriptions: 0 empty, 0 translated, 0 default
- Display Folders: 0 empty, 0 translated, 0 default
- Perspectives: Shown in 1 out of 3 perspectives

Perspectives

- Internet Operation: False
- Inventory: False
- Reseller Operation: True

Metadata



Marc Lelijveld

Data & AI consultant
Macaw Netherlands



✉ Marc.Lelijveld@outlook.com

🐦 @MarcLelijveld

linkedin.com/in/MarcLelijveld

🌐 Data-Marc.com



Our Partners

Gold



dbWatch

Silver



Enterprise Training Center

Global



Please sign in!

We have a little delay with
Teams live, but we will
make it happen!

Join with the Kahoot app
or via Kahoot.it



Our agenda for today

- External Tools, what is it?
- Analysis Services in memory
- Building External Tools
- I want to hear from you!



Documentation

Documentation is overrated?



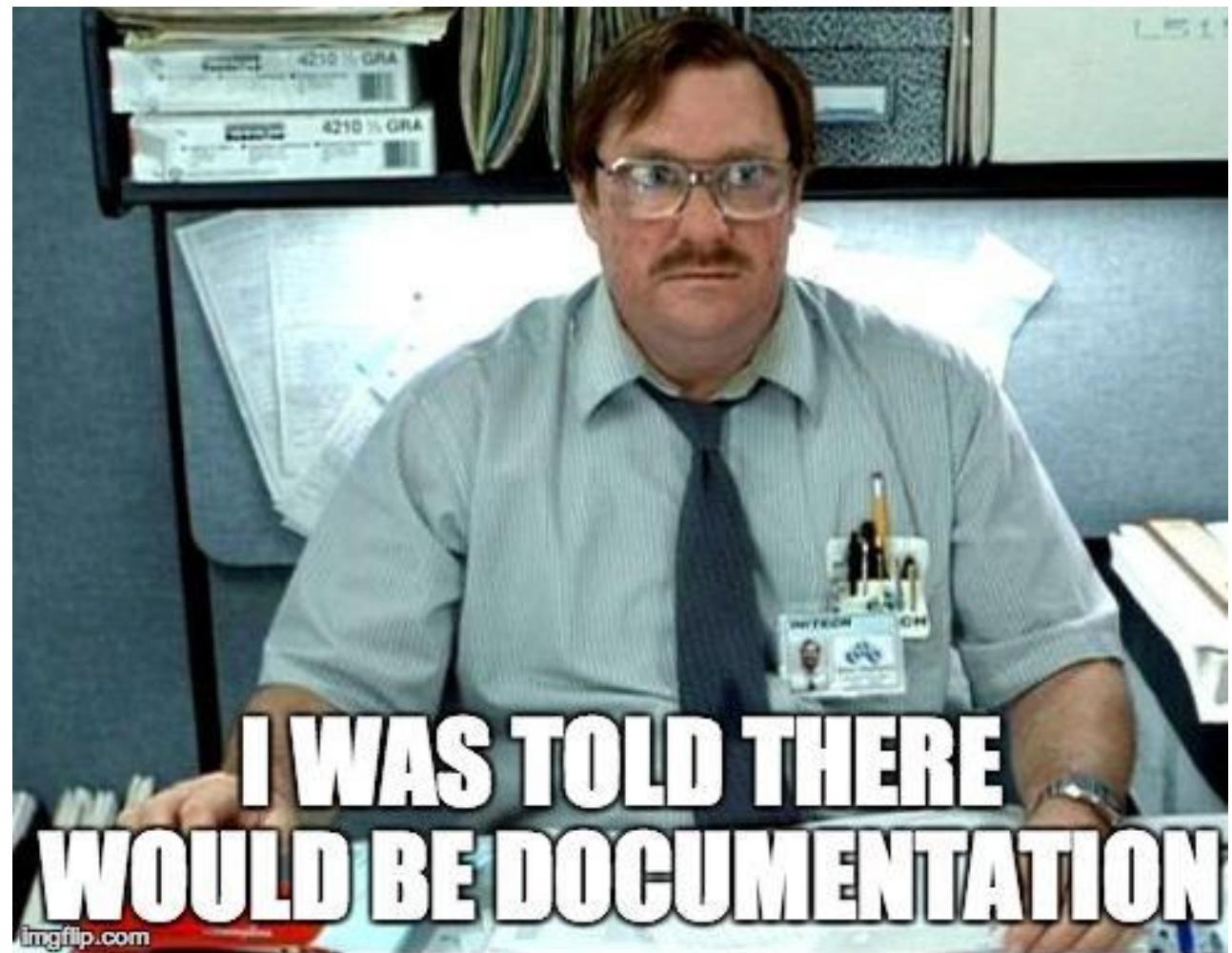
▲ Yes!

◆ No

Documentation is overrated?

▲ Yes!

◆ No



imgflip.com

Documentation?



Marc Lelijveld
@MarcLelijveld

If you deliver or use a #PowerBI solution which uses a shared dataset. Does it come with proper documentation?

Yes, nicely shared in PBI

18.3%

Yes, but too much text!

5%

If I've time left

30%

No, what is documentation

46.7%



Ásgeir Gunnarsson
@bidgeir

...

Als antwoord op [@MarcLelijveld](#)

You're missing an option. If it's prioritised. I always ask for it to be part of the task but too often it's not prioritised. Even though clients pay me for advice, documentation is sadly one of those they most often ignore

Do you currently document your Power BI solutions?



▲ Always! I describe every detail!

◆ I generate something and that's it

● I manually put something together

■ Documentation? What's that?

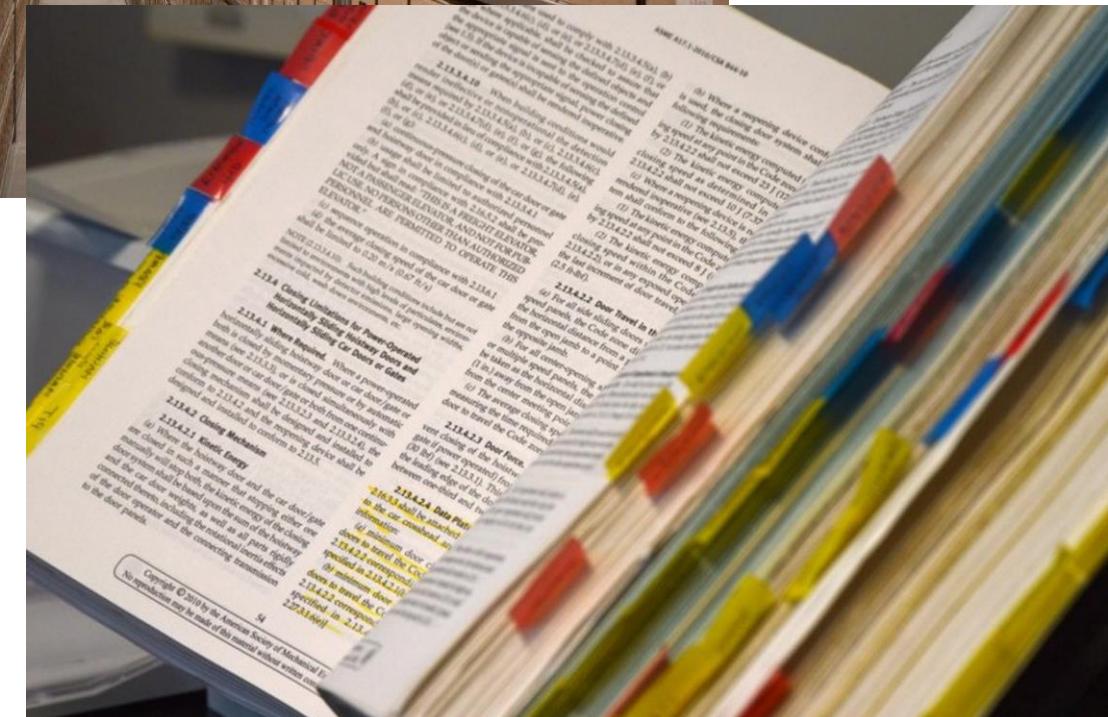
Do you currently document your Power BI solutions?

▲ Always! I describe every detail!

● I manually put something together

◆ I generate something and that's it

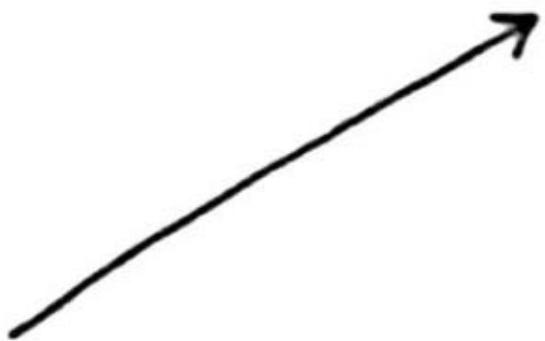
■ Documentation? What's that?



Handover projects

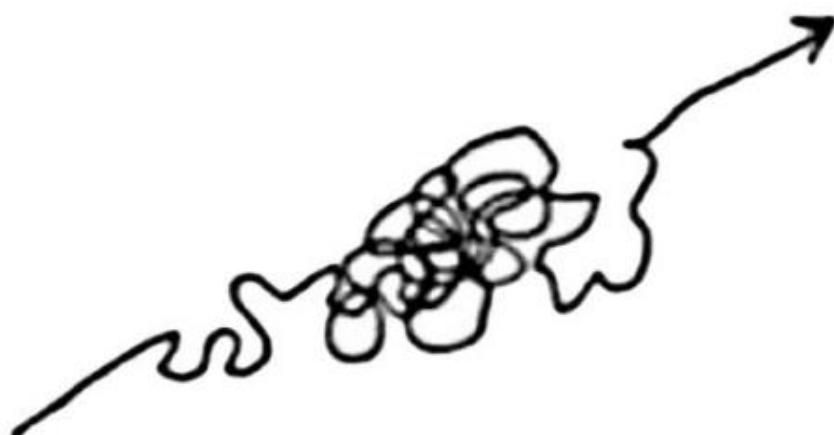


Success



what people think
it looks like

Success



what it really
looks like



Data-Marc.com

I want to document the following items:



▲ Power Query

◆ DAX expressions

● Relationships and model

■ Column and table properties

What do you want to document?

I want to document the following items:

▲ Power Query

● Relationships and model

◆ DAX expressions

■ Column and table properties

The screenshot shows the Power BI Advanced Editor window with the title "SalesLT SalesOrderHeader". The code pane contains the following DAX script:

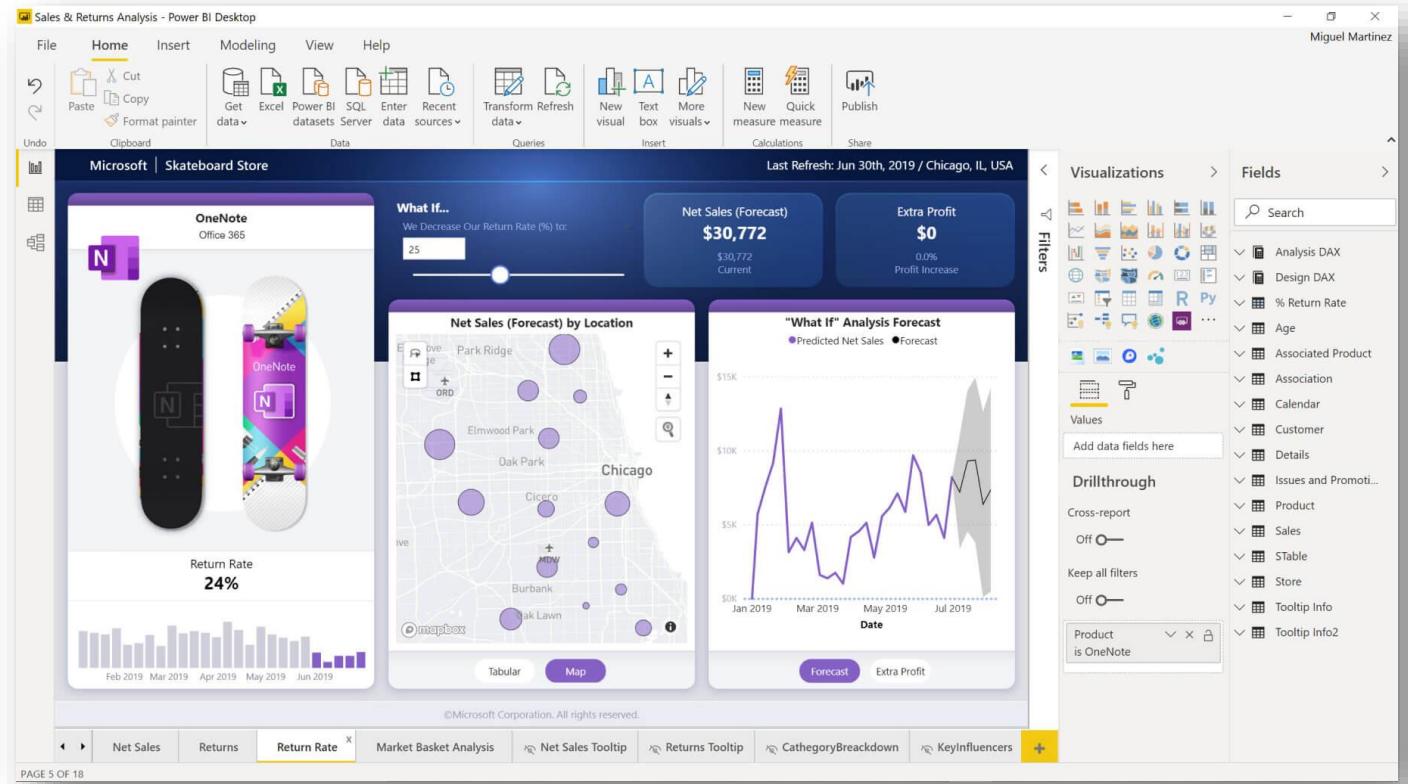
```
let
    Source = Sql.Databases("sqlserver-new.database.windows.net"),
    PowerBIcheatSheet = Source[[Name="PowerBIcheatSheet"]][Data],
    SalesLT_SalesOrderHeader = PowerBIcheatSheet[[Schema="SalesLT",Item="SalesOrderHeader"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(SalesLT_SalesOrderHeader,{{"SalesOrderID", type text}, {"CustomerID", type text}, {"OrderDate", type date}}),
    #"Filtered Rows1" = Table.SelectRows(#"Changed Type", each true),
    #"Replaced Value" = Table.ReplaceValue(#"Filtered Rows1",0,9,Replacer.ReplaceValue,{"Status"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Replaced Value",{{"DueDate", type date}}),
    #"Filtered Rows" = Table.SelectRows(#"Changed Type1", each true)
in
    #"Filtered Rows"
```

Below the code pane is the Power Query Editor interface, showing the "SalesLT SalesOrderHeader" query. The preview pane displays a table with columns: SalesOrderID, Rev, OrderDate, Status, DueDate. A status bar at the bottom indicates "28 COLUMNS, 32 ROWS - Column profiling based on top 1000 rows".

But what about the rest?

- Visuals
- Bookmarks
- Page navigation

The report in general...



Documentation

Model Documentation

26 # Tables 16 # Tables to be deleted 253 # Columns 138 # Columns Hidden 96 # DAX Calculated

Table Type	Table Name	Table Type	Column Name	Encoding
All	SalesLT SalesOrderHeader	PQ	AccountNumber	HASH
			BillToAddressID	HASH
			Comment	HASH
			CreditCardApprovalCode	HASH
			CustomerID	HASH
			DueDate	HASH
			Freight	VALUE
			ModifiedDate	HASH
			OnlineOrderFlag	HASH
			OrderDate	HASH
			PurchaseOrderNumber	HASH
			RevisionNumber	HASH
			rowguid	HASH
			RowNumber-26629798-1795-4f74-8f37-6a1ba8059b61	VALUE
			SalesOrderID	HASH
			SalesOrderNumber	HASH
			ShipDate	HASH
			ShipMethod	HASH
			ShipToAddressID	HASH
			Status	HASH
			SubTotal	VALUE
			TaxAmt	VALUE
			TotalDue	VALUE
	SalesLT ProductModelProductDescription	PQ	Culture	HASH
			ModifiedDate	HASH

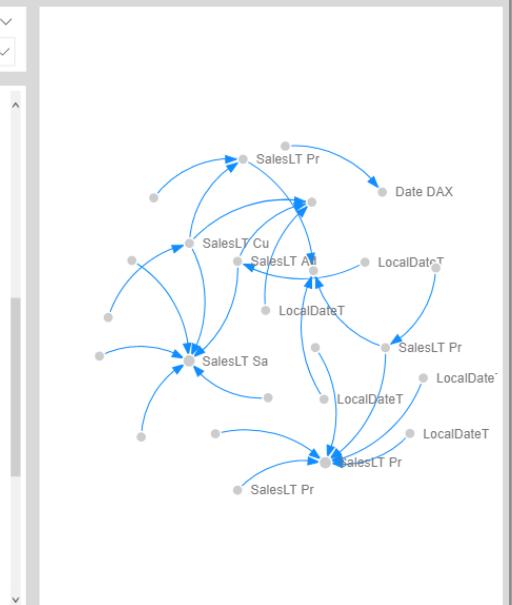
Selected Source Expression

```
let
Source = Sql.Databases("sqlserveraw.database.windows.net"),
PowerBI_CheatSheet = Source[[Name="PowerBI_CheatSheet"]][Data],
SalesLT_SalesOrderHeader =
PowerBI_CheatSheet[[Schema="SalesLT",Item="SalesOrderHeader"]][Data],
#"Changed Type" = Table.TransformColumnTypes(SalesLT_SalesOrderHeader,
{("SalesOrderID", type text), ("CustomerID", type text), ("OrderDate", type date)}),
#"Filtered Rows1" = Table.SelectRows(#"Changed Type", each true),
#"Replaced Value" = Table.ReplaceValue(#"Filtered Rows1",0.9,Replacer.ReplaceValue,{"Status"}),
#"Changed Type1" = Table.TransformColumnTypes(#"Replaced Value",{"(DueDate", type date)}),
#"Filtered Rows" = Table.SelectRows(#"Changed Type1", each true)
in
#"Filtered Rows"
```

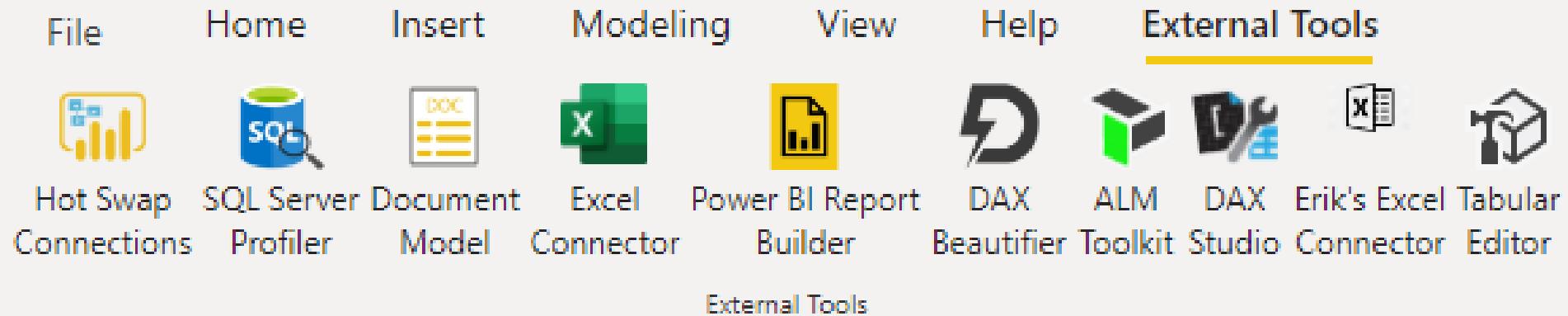
Relationships

25 # Relationships 22 # Active 3 # Inactive 6 # Bi-Directional 2 # Need attention

Relationship	Cardinality	Relation To	
9788-5077-e31a8	One > Many	SalesLT Product - SalesLT Product	
SalesLT Category	One <> Many	SalesLT Product - SalesLT Product	
Model	One > Many	SalesLT Product - SalesLT Product	
83b8-576d-2244e	One > Many	SalesLT ProductDescription - SalesLT ProductDescription	
Customer	One <> One	SalesLT ProductDescription - SalesLT ProductDescription	
a74e-23d1-88e8-4066-a74e-30976a78e82b	One > Many	SalesLT ProductModel - SalesLT ProductModel	
True	LocalDateTable_5f6068d5-8854-4772-b45f-7eedee302ec - LocalDateTable_5f6068d5-8854-4772-b45f-7eedee302ec	One > Many	SalesLT ProductModelProductDescription - SalesLT ProductModelProductDescription
True	SalesLT ProductDescription - SalesLT ProductDescription	One <> One	SalesLT ProductModelProductDescription - SalesLT ProductModelProductDescription
True	SalesLT ProductModel - SalesLT ProductModel	Many > Many	SalesLT ProductModelProductDescription - SalesLT ProductModelProductDescription
True	LocalDateTable_1f72160e-4147-4e47-8a96-23a31f53f17e - LocalDateTable_1f72160e-4147-4e47-8a96-23a31f53f17e	One > Many	SalesLT SalesOrderHeader - SalesLT SalesOrderHeader



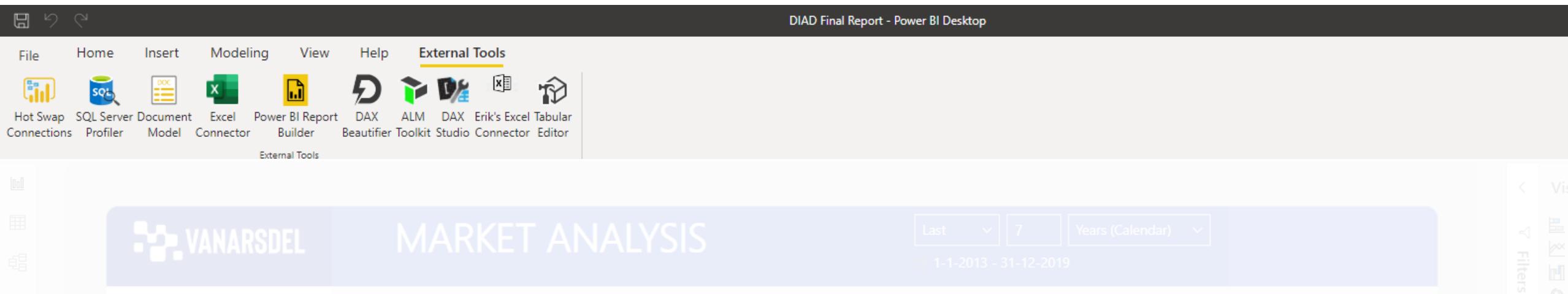
Powered by  Data-Marc.com



External Tools, what is it?

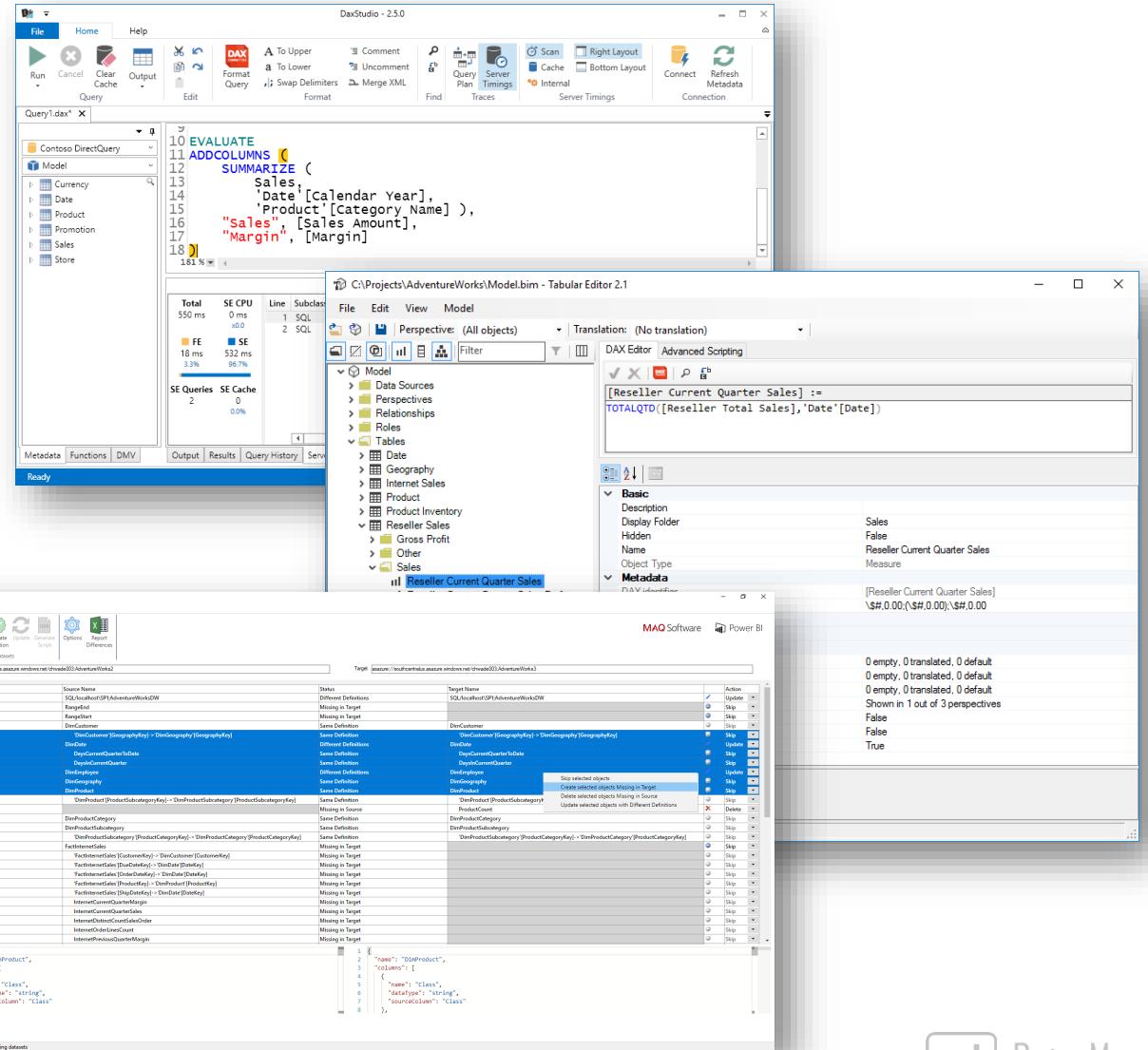
What are external tools in Power BI?

- GA since September release!!
- 3rd party tooling
- Depending on new metadata
- Read + Write to TOM



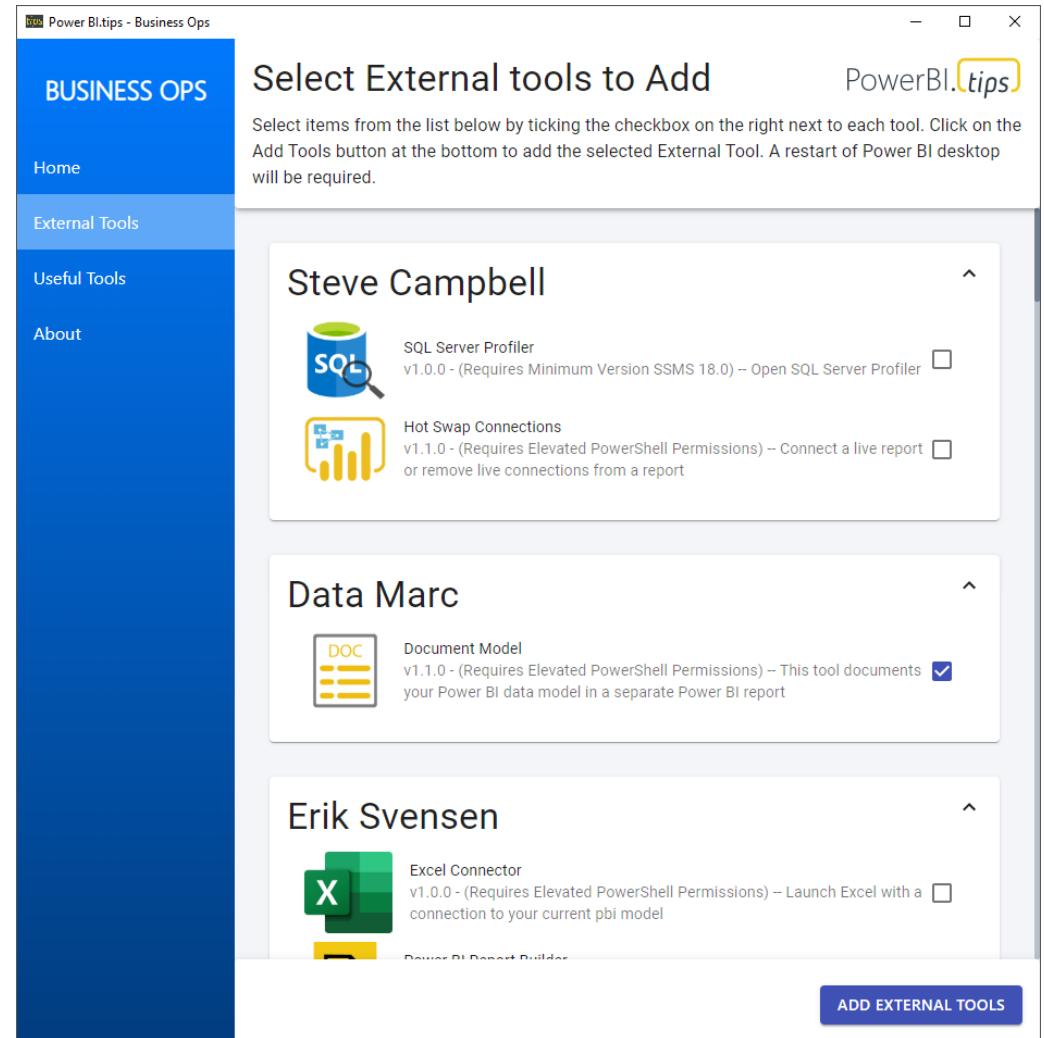
Connect with 3rd party tools

- DAX Studio
- Tabular Editor
- ALM Toolkit
- ...
- Your tool?



Install additional External Tools?

- Requires Admin permissions
- Download them 1-by-1
- Use BusinessOps by PowerBI.tips



What is out there now?

Erik Svensen



Excel Connector
v1.0.0 - (Requires Elevated PowerShell Permissions) -- Launch Excel with a connection to your current pbi model



Power BI Report Builder
v1.0.0 - (Requires Install of Power BI Report Builder) -- Open a template report in Power BI Report Builder



Open In Tableau
v1.0.0 -

PBI.tips - Themes Gallery



Themes Gallery
v1.0.0 - (Opens in Google Chrome) Pick colors from palettes



Themes Gallery
v1.0.0 - (Opens in Edge) Pick colors from thousands of color palettes

Davis Zhang



DAX Beautifier
v1.0.2 - (Requires install of AMO library) -- Use this tool to format all DAX code with one click!

David Eldersveld



Open Python
v1.0.0 - (Requires Python3 All Users Install) -- Opens a connection to a Python Script

Steve Campbell



SQL Server Profiler
v1.0.0 - (Requires Minimum Version SSMS 18.0) -- Open SQL Server Profiler



Hot Swap Connections
v1.1.0 - (Requires Elevated PowerShell Permissions) -- Connect a live report or remove live connections from a report

SQL BI



Analyze in Excel
v1.1.2 - Analyze in Excel for Power BI Desktop

A model documentation tool should be:



- ▲ An application on it's own, like DAX studio, Tabular editor etc.
- ◆ Flexible and easy to customize
- The current version in Power BI is fine for me

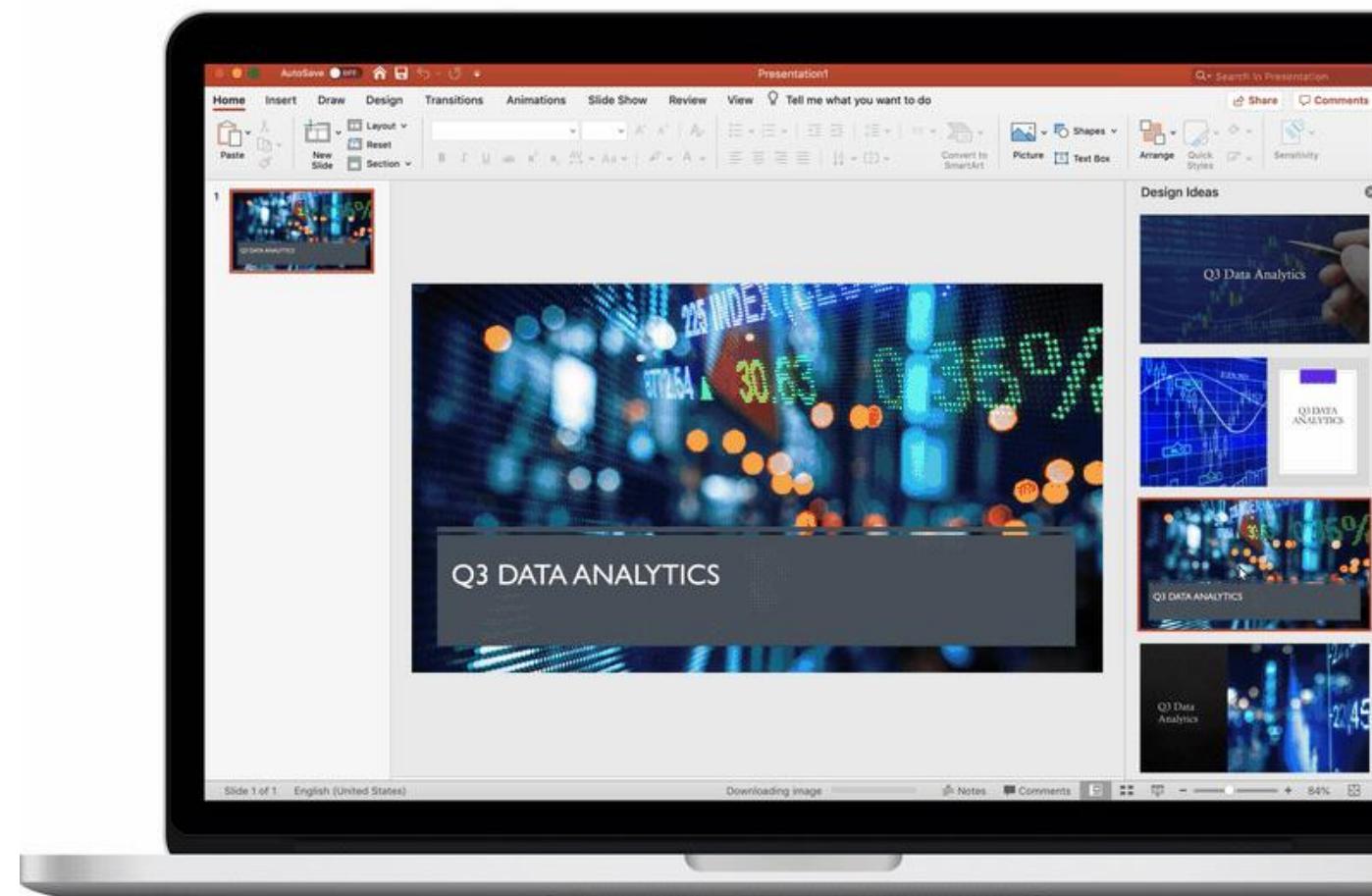
What should a documentation tool look like?

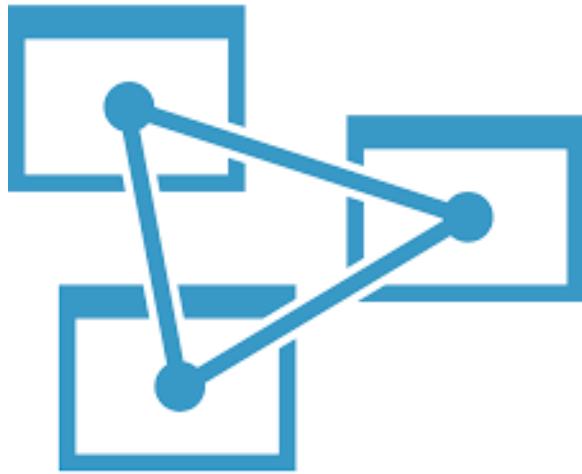
A model documentation tool should be:

▲ An application on its own, like DAX studio, Tabular editor etc.

● The current version in Power BI is fine for me

◆ Flexible and easy to customize





Analysis Services in memory

Analysis services in memory

Power BI

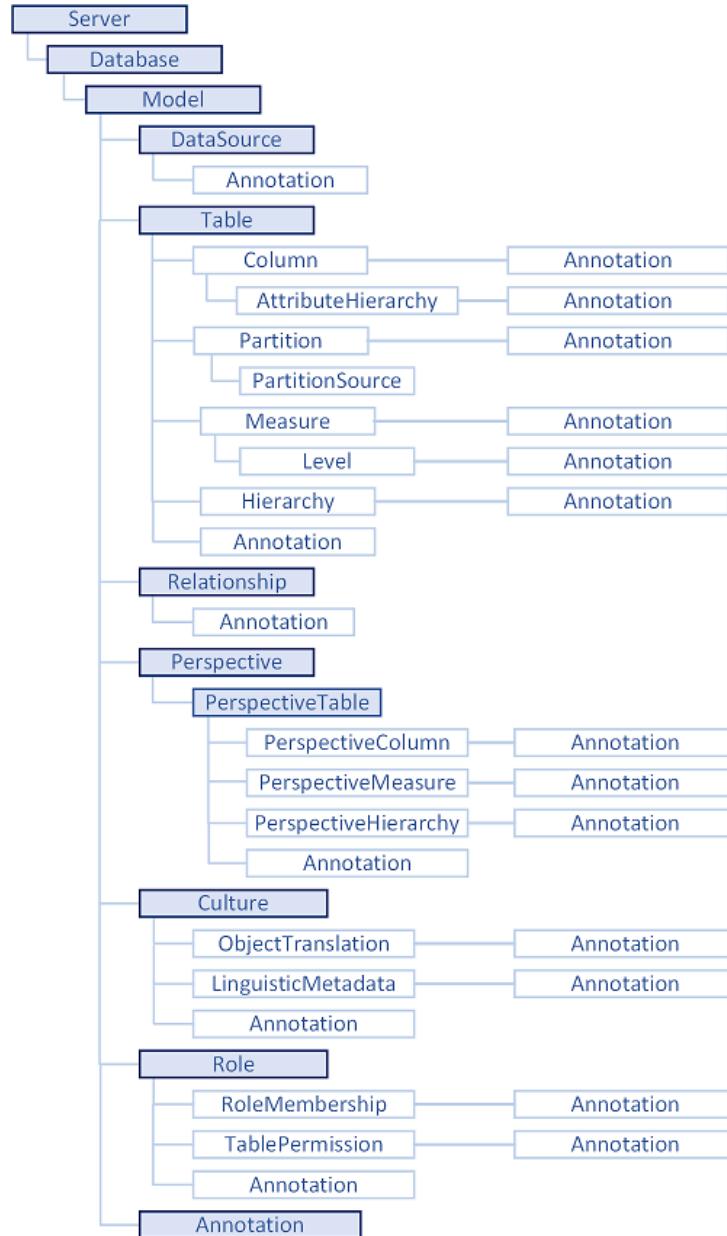


Analysis Services
under the hood

Name	Status	12% CPU	67% Memory	0% Disk	0% Network	4% GPU	GPU engine
> Microsoft Edge (38)		0%	1.501,2 MB	0 MB/s	0,1 Mbps	0%	GPU 0 - 3D
Microsoft Power BI Desktop (9)		0,1%	939,6 MB	0 MB/s	0 Mbps	0%	
CefSharp.BrowserSubprocess		0%	4,0 MB	0 MB/s	0 Mbps	0%	
CefSharp.BrowserSubprocess		0%	17,0 MB	0 MB/s	0 Mbps	0%	
CefSharp.BrowserSubprocess		0%	178,4 MB	0 MB/s	0 Mbps	0%	
CefSharp.BrowserSubprocess		0%	64,3 MB	0 MB/s	0 Mbps	0%	
CefSharp.BrowserSubprocess		0,1%	119,7 MB	0 MB/s	0 Mbps	0%	
CefSharp.BrowserSubprocess		0%	6,8 MB	0 MB/s	0 Mbps	0%	
Console Window Host		0%	5,8 MB	0 MB/s	0 Mbps	0%	
Microsoft SQL Server Analysis Services		0%	251,2 MB	0 MB/s	0 Mbps	0%	
Untitled - Power BI Desktop		0,1%	292,3 MB	0 MB/s	0 Mbps	0%	
> Spotify (32 bit) (3)		0,4%	134,3 MB	0,1 MB/s	0 Mbps	1,1%	GPU 0 - 3D
> Microsoft Teams (7)		2,1%	875,4 MB	0 MB/s	0,1 Mbps	2,2%	GPU 0 - 3D
> Notepad++ : a free (GNU) source code editor		0%	4,7 MB	0 MB/s	0 Mbps	0%	
> OneNote for Windows 10 (2)		0%	1,5 MB	0 MB/s	0 Mbps	0%	
> Task Manager		0,8%	26,3 MB	0 MB/s	0 Mbps	0%	
> Windows Explorer (2)		0,1%	85,2 MB	0 MB/s	0 Mbps	0%	
Background processes (80)							
> 1Password for Windows desktop (32 bit)		0%	2,6 MB	0 MB/s	0 Mbps	0%	
> Adobe Acrobat Update Service (32 bit)		0%	0,2 MB	0 MB/s	0 Mbps	0%	
> Antimalware Service Executable		0,1%	119,6 MB	0 MB/s	0 Mbps	0%	
> cav.exe (32 bit)		0%	2,1 MB	0 MB/s	0 Mbps	0%	

New metadata

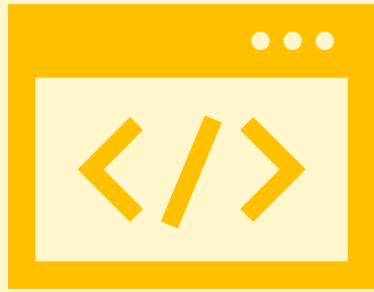
- General Available since September 2020 release!
- Matches Analysis Services metadata
- Model.bim
- Tabular Object Model (TOM)
- Open format (json)



```
1  
2  
3  
4  
5 -{  
6     "version": "1.1.0",  
7     "name": "Document Model",  
8     "description": "This tool documents your Power BI data model in a separate Power BI report.",  
9     "path": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",  
10    "arguments": "C:\\temp\\Data-Marc_WriteConnectionDetailsToFile.ps1 \"%server%\" \"%database%\"",  
11    "iconData": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAANUAAAD7CAMAAADKOCH3AAAAAXNSR0IABJZG5dAAAAAElFTkSuQmCC",  
12 }  
13  
14  
15  
16
```

Building External Tools

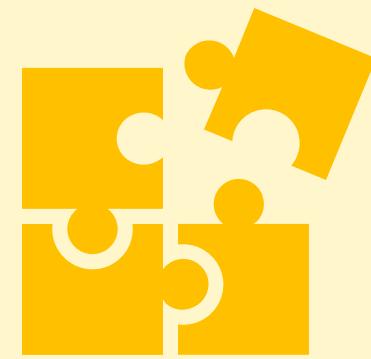
How to setup your own External Tool?



**Build your
application**

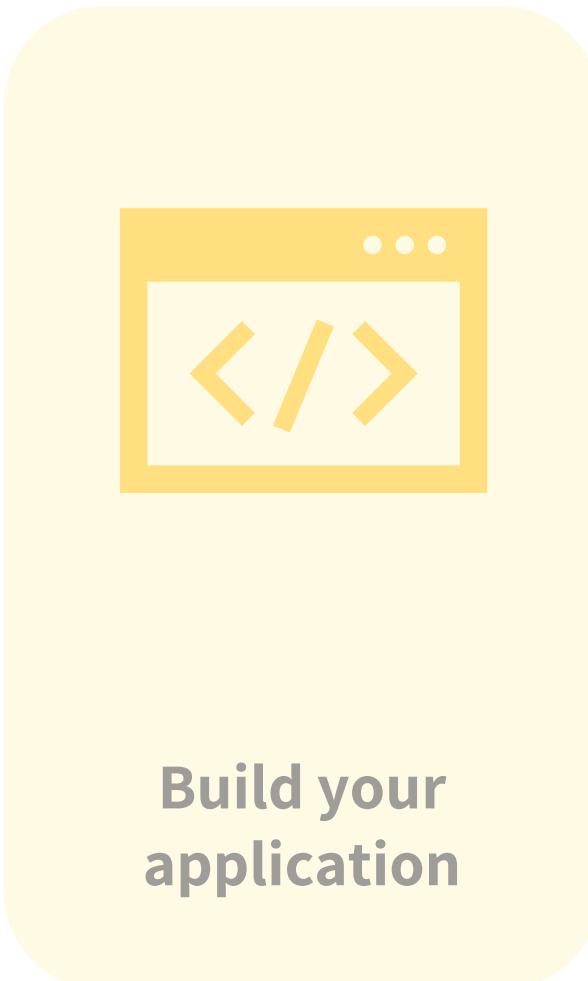


Create your icon



**Integrate in
Power BI Desktop**

How to setup your own External Tool?

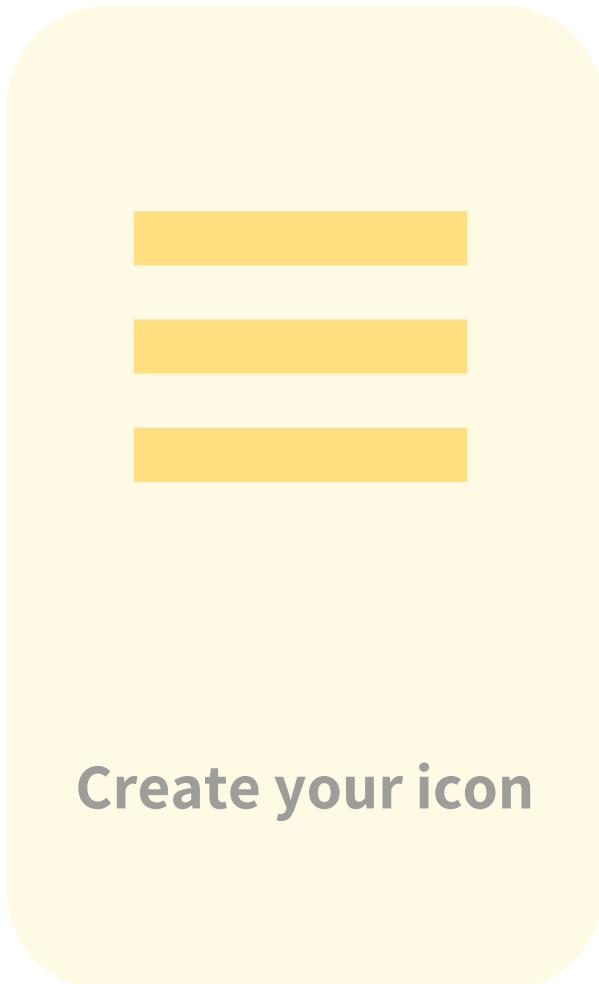


Build your application

A screenshot of the Windows PowerShell ISE interface. The main window displays a PowerShell script named "Data-Marc_WriteConnectionDetailsToFile.ps1". The script is used to write connection details to a file. It defines variables for installation location, default location, and final location, and then writes these to a log file. It also includes a function to download files from a GitHub repository. The right side of the interface shows a "Commands" palette with a list of available cmdlets, and the bottom shows a terminal window with the command "PS D:\OneDrive Macaw\OneDrive - Macaw\External Tools - Model Documentation>" and the status "Ln 20 Col 54 | 115%".

```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Data-Marc_WriteConnectionDetailsToFile.ps1
10
11 # Below you can define your personal preference for file saving and reading.
12 # The default location can be changed and will be leverages throughout the entire script.
13 # InstallerLocation only applies to installation via PowerBI.tips Business Ops.
14 $InstallerLocation = "$TOOL_INSTALL_DIR\"
15 $DefaultLocation = 'C:\BusinessOpsTemp'
16 $finalLocation = if($InstallerLocation -like "*TOOL_INSTALL_DIR*")
17 {$DefaultLocation} else {$InstallerLocation}
18
19 # Write out file locations
20 Write-Host 'installer location ' + $InstallerLocation
21 Write-Host 'default location ' + $DefaultLocation
22 Write-Host 'final location ' + $finalLocation
23
24 #This part starts tracing to catch unfortunate errors and defines where to write the file.
25 $LogFile = $finalLocation + 'PBI_DocumentModel_Logfile.txt'
26 Start-Transcript -Path $LogFile
27
28 # Function to automatically download the pbit file if it cannot be found on the defined location.
29 # Function based on https://gist.github.com/chrisbrownie/f20cb4508975fb5da145d3d38024a
30 function DownloadfilesFromRepo {
31     Param(
32         $Owner = 'marcelijleveld',
33         $Repository = 'External-Tools-Model-Documentation',
34         $Path = 'ModelDocumentationTemplate.pbit',
35         $DestinationPath = 'C:\BusinessOpsTemp'
36     )
37
38     $baseUri = "https://api.github.com/"
39     $uriPath = "repos/$Owner/$Repository/contents/$Path"
40     $wr = Invoke-WebRequest -Uri $($baseUri+$uriPath)
41     $objects = $wr.Content | ConvertFrom-Json
42     $files = $objects | Where-Object {$_.type -eq "file"} | Select-Object -exp download_url
43     $directories = $objects | Where-Object {$_.type -eq "dir"}
44
45     $directories | ForEach-Object {
46         DownloadFilesFromRepo -Owner $Owner -Repository $Repository -Path $_.path -DestinationPath $($DestinationPath)
47     }
48
49     if (-not (Test-Path $DestinationPath)) {
50         # Create directory if it does not exist
51     }
}
PS D:\OneDrive Macaw\OneDrive - Macaw\External Tools - Model Documentation>
```

How to setup your own External Tool?



Create your icon

The screenshot shows a web application for converting images to Base64. At the top right, the text "CONVERT YOUR IMAGES TO BASE64" is displayed above a dashed box containing "DRAG & DROP IMAGES ANYWHERE" and "OR CLICK HERE". In the center, there is a cartoon character named "MR. BASE" wearing a white shirt and tie, holding a briefcase. Below the character are three sections: "File Formats", "Image optimization", and "Browser Support".

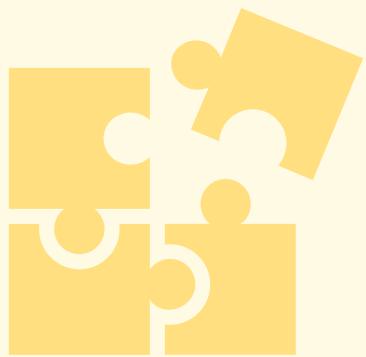
File Formats
You can upload up to 20 images (max. 1.00 MB each) as JPG, PNG, GIF, WebP, SVG or BMP.
Please note that Internet Explorer 8 has a limit of 32 KB for [data URI](#). Versions below have no support.

Image optimization
We can optimize your JPEG & PNG images, using [jpegoptim](#) and [optipng](#). This will reduce the file without any visible impact (in most cases).
Optimization status:

Browser Support
The [encoded results](#) are [compatible](#) with all of the following browsers and versions. Please send me a note if there are issues.

Chrome 4+	Safari 3.1+
Firefox 2+	Opera 9+
Edge	IE 8+

How to setup your own External Tool?



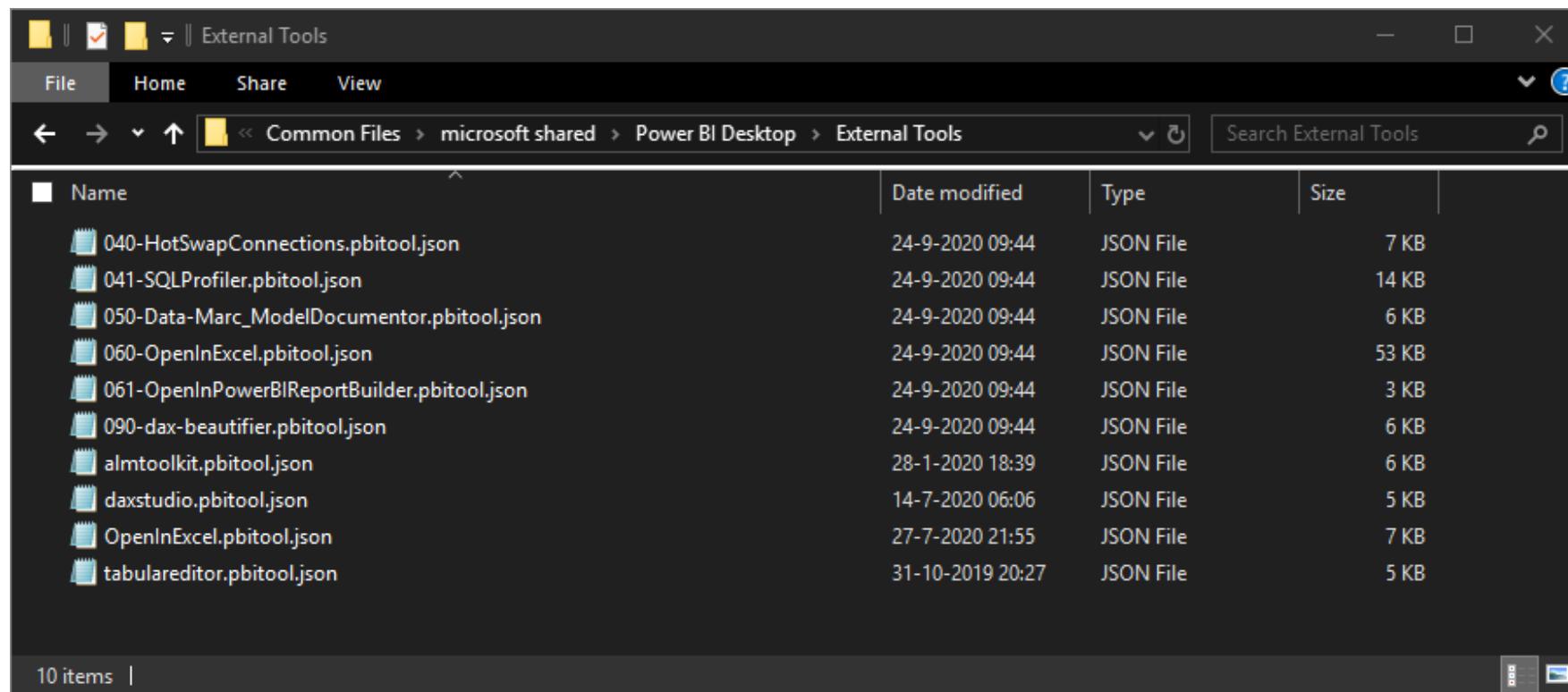
Integrate in
Power BI Desktop

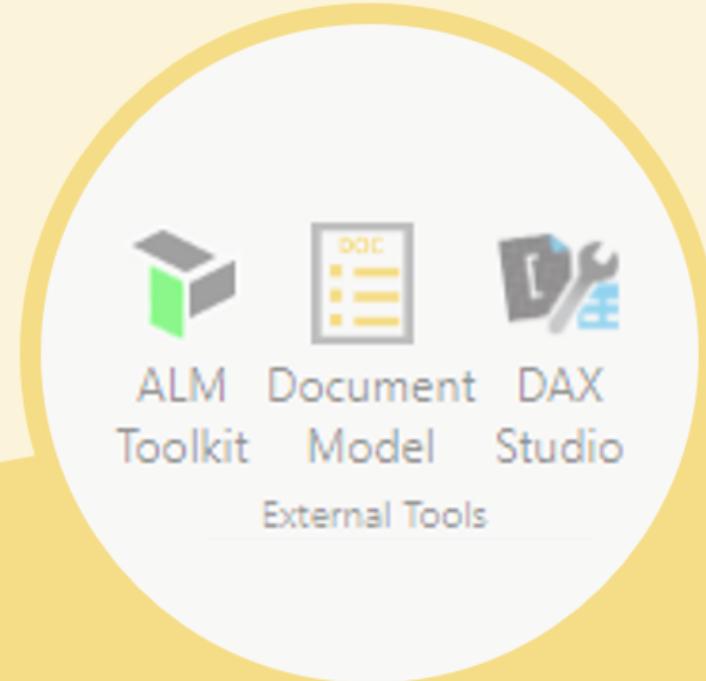
- *.pbitool.json file
- \%\server%\
- \%\database%\

```
{  
  "version": "1.1.0",  
  "name": "Document Model",  
  "description": "This tool documents your Power BI data model in a separate Power BI report.",  
  "path": "C:\\Windows\\\\System32\\\\WindowsPowerShell\\\\v1.0\\\\powershell.exe",  
  "arguments": "C:\\\\temp\\\\Data-Marc_WriteConnectionDetailsToFile.ps1 \\"%server%\" \\"%database%\"",  
  "iconData": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAANUAAAD7CMAAADKOCH3AAAAAXNSR0I\"}
```

Drop the file in the right location

C:\Program Files (x86)\Common Files\Microsoft Shared\Power BI Desktop\External Tools





Document your
model

Power BI Model Documenter

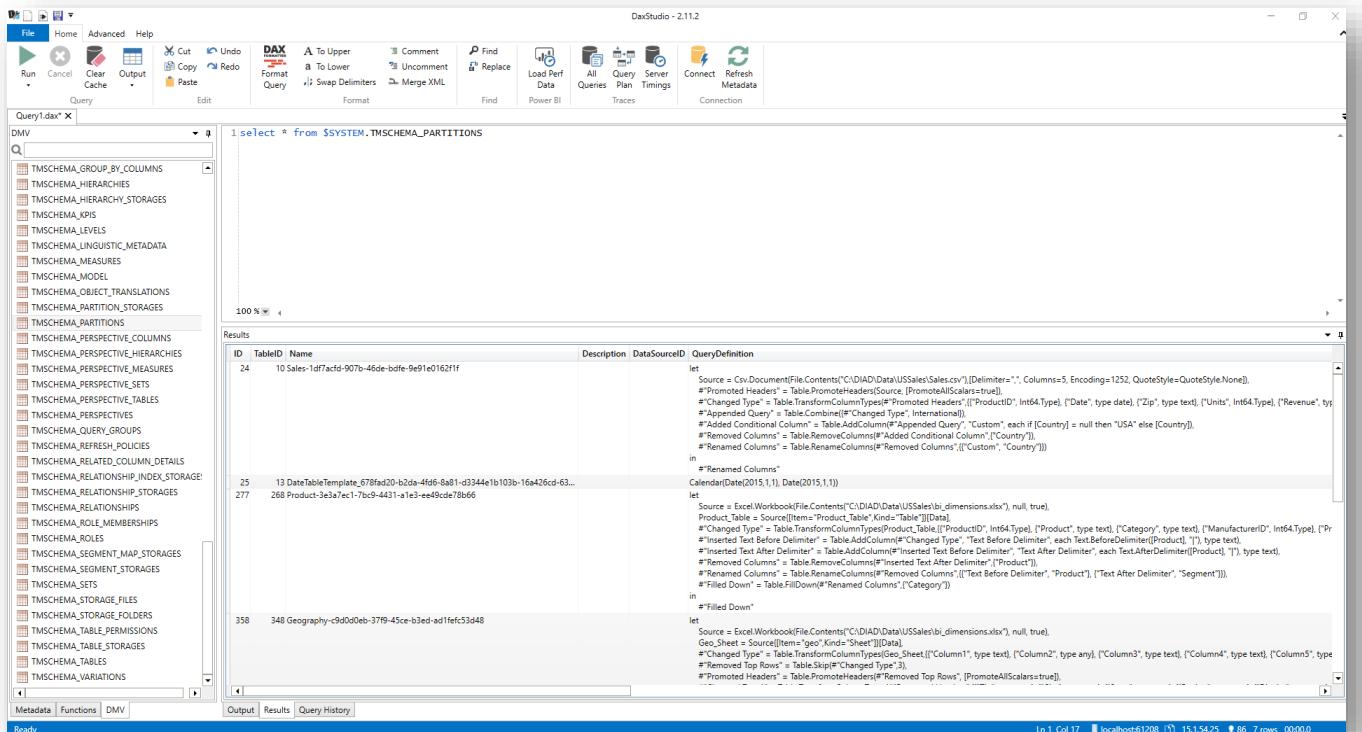
Dynamic Management Views

Analysis Services Dynamic Management Views (DMVs) are queries that return information about model objects, server operations, and server health.

- DB Schema = Database model
- DISCOVER = Operations & Sessions
- TM Schema = Tabular = Power BI / AAS
- MD Schema = MDX = Multidimensional

Dynamic Management Views

- Analysis Services metadata
 - Tables
 - Columns
 - Measures
 - Perspectives
 - Partitions
 - ...
- Query via DAX Studio



The screenshot shows the DAX Studio interface with a query window titled "Query1.dax". The query is:

```
1 select * from $SYSTEM.TMSCHEMA_PARTITIONS
```

The results pane displays several partitions with their IDs, TableIDs, names, descriptions, and data source IDs. The first partition is for Sales, the second for DateTableTemplate, and the third for Geography. The "QueryDefinition" pane shows the DAX code used to generate these results.

```
let
    Source = Csv.Document(File.Contents("C:\DAX\Datasets\Sales.csv"),[Delimiter=",", Columns=5, Encoding=1252, QuoteStyle=QuoteStyleNone]),
    #>Promoted Headers? = Table.PromoteHeaders(Source,[PromoteAllScalars=true]),
    #>Changed Type? = Table.TransformColumnTypes(#>Promoted Headers,[{"ProductID": Int64.Type}, {"Date": type date}, {"Zip": type text}, {"Units": Int64.Type}, {"Revenue": ty
    #>Appended Query? = Table.Combine(#>Changed Type, International),
    #>Added Conditional Column? = Table.AddColumn(#>Appended Query, "Custom", each if [Country] = null then "USA" else [Country]),
    #>Removed Columns? = Table.RemoveColumns(#>Added Conditional Column,[{"Country"}]),
    #>Renamed Columns? = Table.RenameColumns(#>Removed Columns,[{"Custom", "Country"}])
in
#>Renamed Columns",
CalendarDate(2015,1), Date(2015,1))
let
    Source = Excel.Workbook(File.Contents("C:\DAX\Datasets\bi_dimensions.xlsx"), null, true),
    Product = Source[[Product],Table Kind=Table],
    #>Changed Type? = Table.TransformColumnTypes(Product,Table[{"ProductID": Int64.Type}, {"Product": type text}, {"Category": type text}, {"ManufacturerID": Int64.Type}, {"Pr
    #>Inserted Text Before Delimiter? = Table.AddColumn(#>Changed Type, "Text Before Delimiter", each Text.BeforeDelimiter([Product]), "Text"),
    #>Inserted Text After Delimiter? = Table.AddColumn(#>Inserted Text Before Delimiter, "Text After Delimiter", each Text.AfterDelimiter([Product]), "Text"),
    #>Removed Columns? = Table.RemoveColumns(#>Inserted Text After Delimiter,[{"Product"}]),
    #>Renamed Columns? = Table.RenameColumns(#>Removed Columns,[{"Text Before Delimiter", "Product"}, {"Text After Delimiter", "Segment"}]),
    #>Filled Down? = Table.FillDown(#>Renamed Columns,[{"Category"}])
in
#>Filled Down",
let
    Source = Excel.Workbook(File.Contents("C:\DAX\Datasets\bi_dimensions.xlsx"), null, true),
    Geo_Sheet = Source[[Geo],Table Kind=Table],
    #>Changed Type? = Table.TransformColumnTypes(Geo_Sheet,[{"Column1": type text}, {"Column2": type any}, {"Column3": type text}, {"Column4": type text}, {"Column5": type
    #>Removed Top Rows? = Table.Skip(#>Changed Type, 3),
    #>Promoted Headers? = Table.PromoteHeaders(#>Removed Top Rows,[PromoteAllScalars=true]),
    #>Filled Down? = Table.FillDown(#>Promoted Headers,[{"Category"}])
in
#>Filled Down"
```

At the bottom of the interface, there are tabs for "Ready", "Output", "Results", and "Query History".

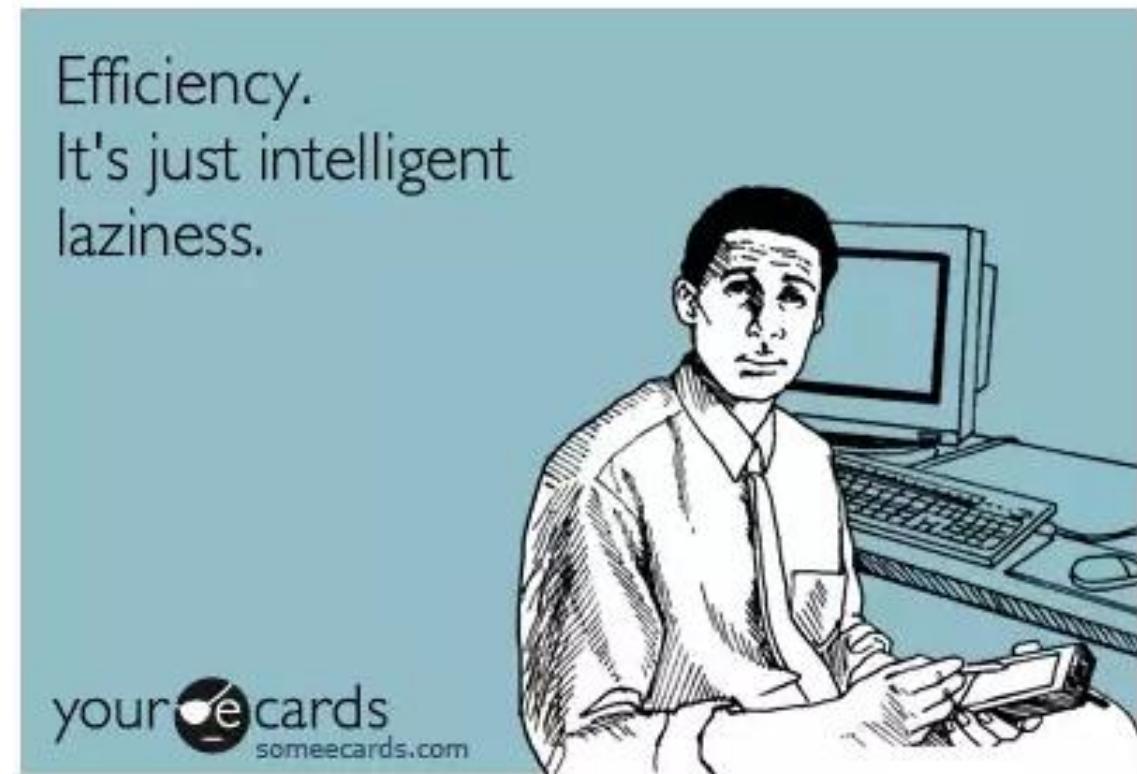
DEMO!

- Show all tables in model
 - Table Expressions
- Show all measures
- Show Perspectives
 - Measures in perspectives
 - Columns in perspectives
- Show columns
- Roles
 - Role expressions



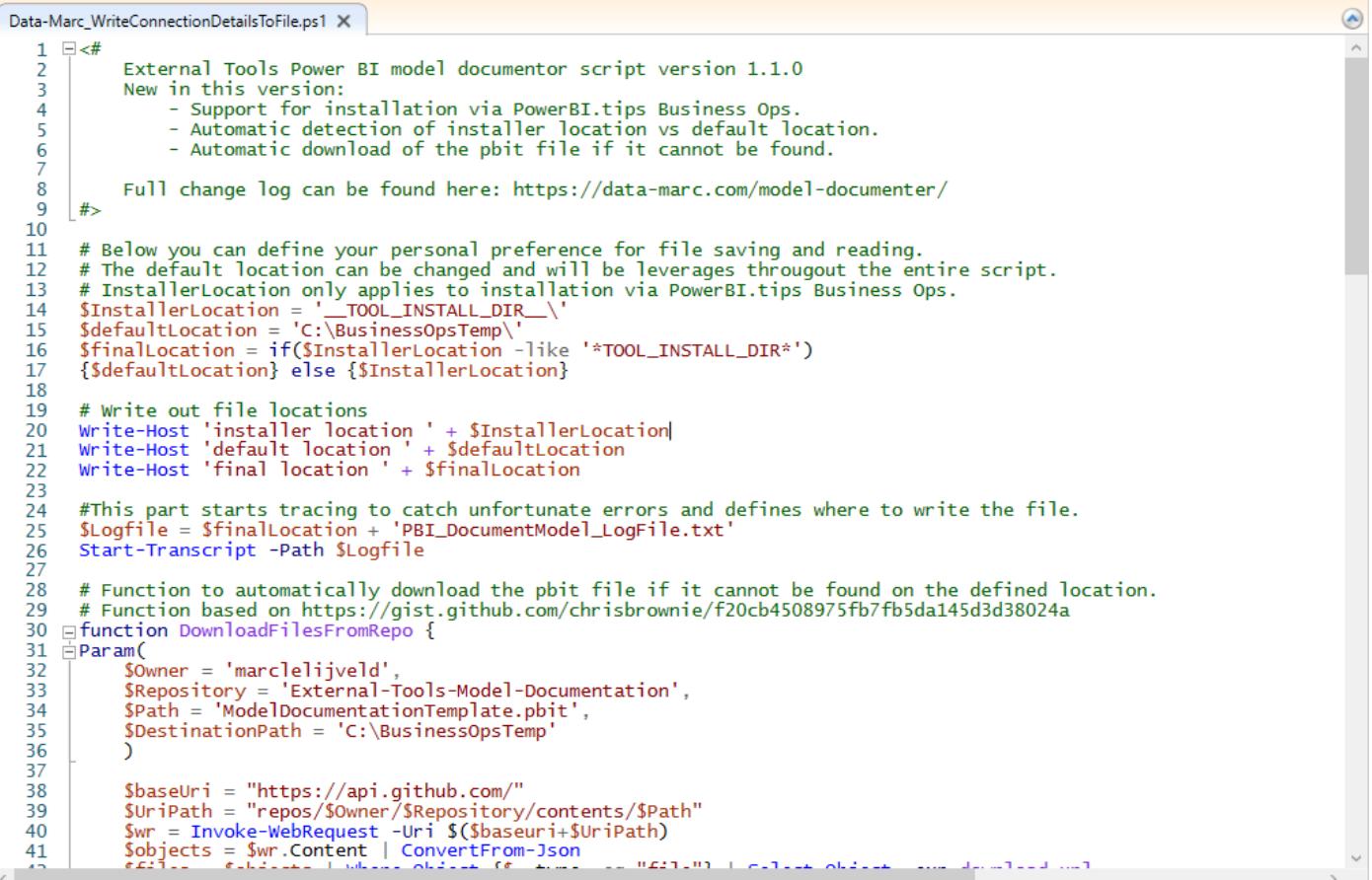
Automation of documentation

- Need for something to execute tasks automatically.
- Something that runs on every computer



PowerShell

- Retrieve Server and Database information
- Dump connection file in json format
- Download template from GitHub
- Opens Power BI template



The screenshot shows a code editor window titled "Data-Marc_WriteConnectionDetailsToFile.ps1". The script is a PowerShell function designed to write connection details to a file. It includes comments explaining its purpose and how it handles different installation paths. The code uses Write-Host to output locations, Start-Transcript to log errors, and Invoke-WebRequest to download a template file from GitHub.

```
1 #<#
2     External Tools Power BI model documentor script version 1.1.0
3     New in this version:
4         - Support for installation via PowerBI.tips Business Ops.
5         - Automatic detection of installer location vs default location.
6         - Automatic download of the pbit file if it cannot be found.
7
8     Full change log can be found here: https://data-marc.com/model-documenter/
9 #>
10
11 # Below you can define your personal preference for file saving and reading.
12 # The default location can be changed and will be leverages throughout the entire script.
13 # InstallerLocation only applies to installation via PowerBI.tips Business Ops.
14 $InstallerLocation = '__TOOL_INSTALL_DIR__'
15 $defaultLocation = 'C:\BusinessOpsTemp\'
16 $finalLocation = if($InstallerLocation -like '*TOOL_INSTALL_DIR*') { $defaultLocation } else { $InstallerLocation }
17
18 # Write out file locations
19 Write-Host 'installer location ' + $InstallerLocation
20 Write-Host 'default location ' + $defaultLocation
21 Write-Host 'final location ' + $finalLocation
22
23 #This part starts tracing to catch unfortunate errors and defines where to write the file.
24 $LogFile = $finalLocation + 'PBI_DocumentModel_LogFile.txt'
25 Start-Transcript -Path $LogFile
26
27 # Function to automatically download the pbit file if it cannot be found on the defined location.
28 # Function based on https://gist.github.com/chrisbrownie/f20cb4508975fb7fb5da145d3d38024a
29 function DownloadfilesFromRepo {
30     Param(
31         $Owner = 'marcelijeveld',
32         $Repository = 'External-Tools-Model-Documentation',
33         $Path = 'ModelDocumentationTemplate.pbit',
34         $DestinationPath = 'C:\BusinessOpsTemp'
35     )
36
37     $baseUri = "https://api.github.com/"
38     $uriPath = "repos/$Owner/$Repository/contents/$Path"
39     $wr = Invoke-WebRequest -Uri $($baseUri+$uriPath)
40     $objects = $wr.Content | ConvertFrom-Json
41     $objects | Where-Object { $_.type -eq "file" } | Select-Object name, content | Out-File -Encoding utf8 $destinationPath
42 }
```

Learnings

- Capturing server and database parameters
- Double \\ in path took a while to figure out
- PowerShell in general + debugging
- On-demand editing *.pbit file not possible
- Dump connection json file as “workaround”

Learn from others!

What's new in v1.2.0

- Better error handling
- Folder creation if not exist yet
- Menu at startup
- Support two export types
 - Power BI Template
 - Excel Template

The screenshot shows the Model Documentation feature in both Power BI Desktop and Excel.

Power BI Desktop: A ribbon menu bar with tabs like File, Home, Insert, Modeling, View, Help, External Tools, and others. Below it is a "Model Documentation" card showing metrics: 8 Measures, 1 Hidden, 4 In displayfolder, 6 Need description, and 1 Need attention. It lists tables like SalesLT.SalesOrderHeader, SalesLT.Customer, and SalesLT.Product with their respective measures and descriptions.

Excel: An open Excel spreadsheet titled "ModelDocumentationTemplate1 - Excel". The first sheet contains a table titled "Model Documentation" with columns for FromTable, FromColumn, ToTable, Active, and Relationship. The table lists various relationships between tables such as Date, SalesLT.Address, SalesLT.Customer, SalesLT.CustomerAddress, SalesLT.Product, SalesLT.ProductCategory, SalesLT.ProductModel, SalesLT.SellEndDate, SalesLT.SellStartDate, SalesLT.SalesOrderDetail, SalesLT.SalesOrderHeader, SalesLT.SalesOrderHeaderBillToAddress, SalesLT.SalesOrderHeaderCustomer, SalesLT.SalesOrderHeaderDelivery, SalesLT.SalesOrderHeaderModifiedDate, SalesLT.SalesOrderHeaderOrderDate, SalesLT.SalesOrderHeaderShipDate, and SalesLT.SalesOrderHeaderShipToAddressID. The "Relationship" column contains DAX formulas describing the relationship between tables and columns.

Right-hand pane: A sidebar with sections for Visualizations, Fields, PivotTable Fields, and Relationships, providing detailed metadata for each field and relationship.

DEMO!

Build new tool for PBIG

- New icon in PBI Desktop

Demo Model Documenter

- How to query DMVs in Power BI
- The tool itself

Nobody:

Bugs right before a demo:



It's showtime...

Known limitations (or irritations)

- PowerShell Execution Policies
- Power BI Privacy levels might block loading
- Native database queries
- Requires admin approval for installation
- Requires MSOLAP.8 Provider to be installed
- Excel template requires manual download

```
PS C:\Users\Marcl> Get-ExecutionPolicy -List
```

scope	ExecutionPolicy
MachinePolicy	Undefined
UserPolicy	Undefined
Process	Undefined
CurrentUser	Undefined
LocalMachine	Unrestricted

Privacy Levels

Always combine data according to your Privacy Level settings for each source

Combine data according to each file's Privacy Level settings

Always ignore Privacy Level settings ⓘ

Native Database Queries

Require user approval for new native database queries

What is currently missing and do you want to add?



Players, type your answer!

Planned enhancements

What is currently missing and do you want to add?



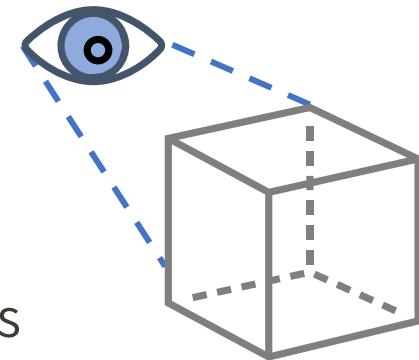
Soon

- Adding Roles + expressions
- Perspectives
- Auto download Excel template



Later

- Support for live connected models
- Next-next-finish installer



So, from now on...



Wrap up

External Tools...

...are depending on Analysis Service **metadata** format.

...allow you to develop with **3rd party tools**.

...opens tons op **opportunities to contribute** to Power BI.

LET'S
RECAP...

Model Documenter...

... allows me to be lazy (or efficient) by **generating** documentation.

... opens new opportunities to **hand-over** your solutions.

... is super powerful for **self-service purposes** and end-users.

For resources

[Https://data-marc.com/model-documenter/](https://data-marc.com/model-documenter/)



Marc Lelijveld

Data & AI consultant
Macaw Netherlands

- Marc.Lelijveld@outlook.com
- [@MarcLelijveld](https://twitter.com/MarcLelijveld)
- linkedin.com/in/MarcLelijveld
- Data-Marc.com



Document your
model

