



A Power BI Documentation System

Not Just an Afterthought





Special thanks to Fabric and Power BI Team at



This Summit presented to you by





David Kofod Hanna



Senior Advisor, Data Storytelling @ Twoday
+200 courses as Academy Trainer and 10 years as consultant
Microsoft Data Platform MVP, Certified Trainer in Microsoft and Tabular Editor



Passionate about guiding self-service Power BI
developers for more enterprise manageable concepts in a consumable and
practical way



Born on beautiful “Sunshine island”: Bornholm
Lives in Denmark, Silkeborg with wife and 3 kids
Love football and running half-marathons





Who **loves** documentation?





Who **writes** documentation?





Who **updates** documentation?





Who reads documentation?





Who can **find** documentation?





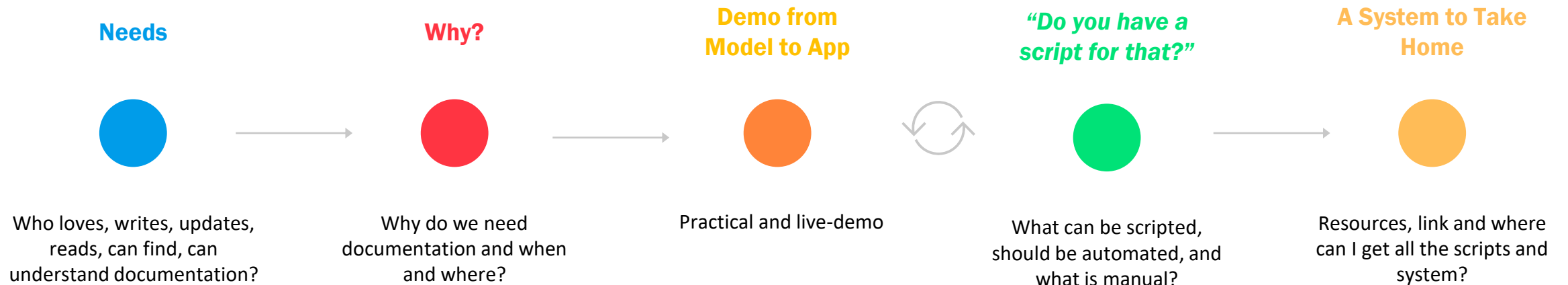
Who can understand documentation?





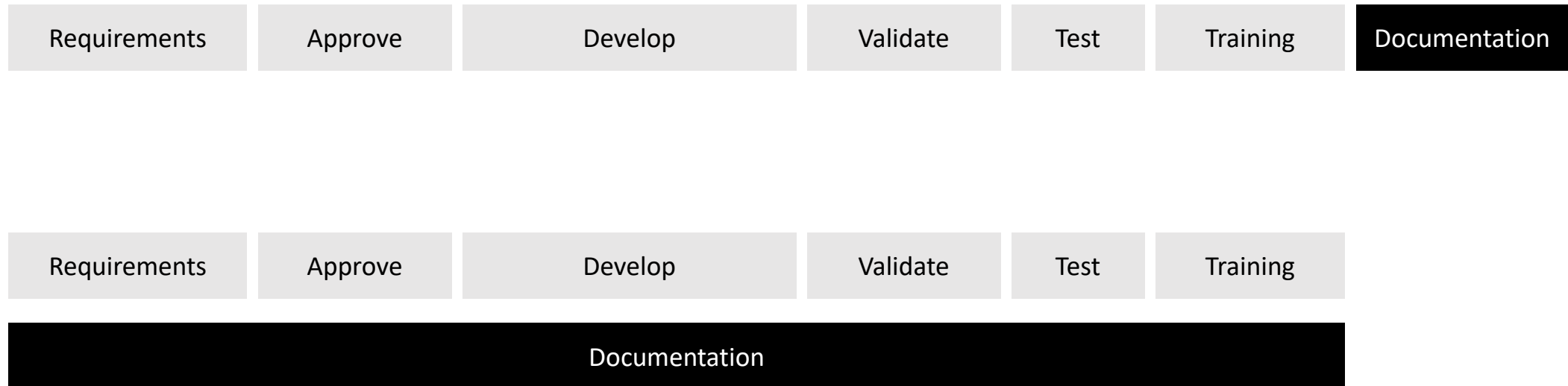
A Power BI Documentation System

– Not Just an Afterthought





Example of project





Why documentation?

Transparency

Reproducibility

Collaboration

Compliance

Scalability

Easier debugging







 Please write my
documentation





*“You do not rise to the level of your **goals**, you fall to the level of your **systems**.”*

- James Clear, Atomic Habits





Power BI Documentation System – from Model to App



Power Query
{M}



Semantic
Model



Power BI
Desktop



App



Purpose for docs:
*Clear, Concise,
Complete, Consistent,
Correct, Consumable*

Documentation and
transparency from
model to app

From report consumers
to report developers to
semantic model
developers and data
ingest

Best Practice Analyzer
Rules to highlight
violations and severity
of best practices

Tabular Editor 2/3 for
Semantic Model and
DAX scripts

Semantic Link Labs in
Fabric Notebooks for
automation like never
before





Power BI Documentation System – from Model to App

Automated or Script
 Manual
 Requires Fabric SKU



Power Query
{M}



Semantic Model



Power BI Desktop



App



Purpose for docs:

*Clear, Concise,
Complete, Consistent,
Correct, Consumable*

Documentation and transparency from model to app

From report consumers to report developers to semantic model developers and data ingest

Best Practice Analyzer Rules to highlight violations and severity of best practices

Tabular Editor 2/3 for Semantic Model and DAX scripts

Semantic Link Labs in Fabric Notebooks for automation like never before





Power BI Documentation System – from Model to App

Automated or Script
 Manual
 Requires Fabric SKU



Power Query {M}

Query Folders

Applied Steps
(Rename + Properties)

Table Properties

Parameters for easy switch

Last Refresh & Measure Table & Table.Profile

Powerqueryformatter.com



Semantic Model

Format DAX

DAX Expression as Description

INFO.VIEW Model Docs

Display Folders for Measures & Columns

Table Groups

Semantic Model Docs



Power BI Desktop

Model Layout View per Fact

DAX Query View + VertiPaq Analyzer

Unpin Ineffective Default Visuals

Report Theme JSON Docs

Report Visual elements in Selection Pane

Report Docs



App

Video and PDF in an App

Notebooks in an app

OneLake Catalog

All into a Lakehouse

External Tools & Copilot

End-User Docs

Purpose for docs:

Clear, Concise, Complete, Consistent, Correct, Consumable

Documentation and transparency from model to app

From report consumers to report developers to semantic model developers and data ingest

Best Practice Analyzer Rules to highlight violations and severity of best practices

Tabular Editor 2/3 for Semantic Model and DAX scripts

Semantic Link Labs in Fabric Notebooks for automation like never before



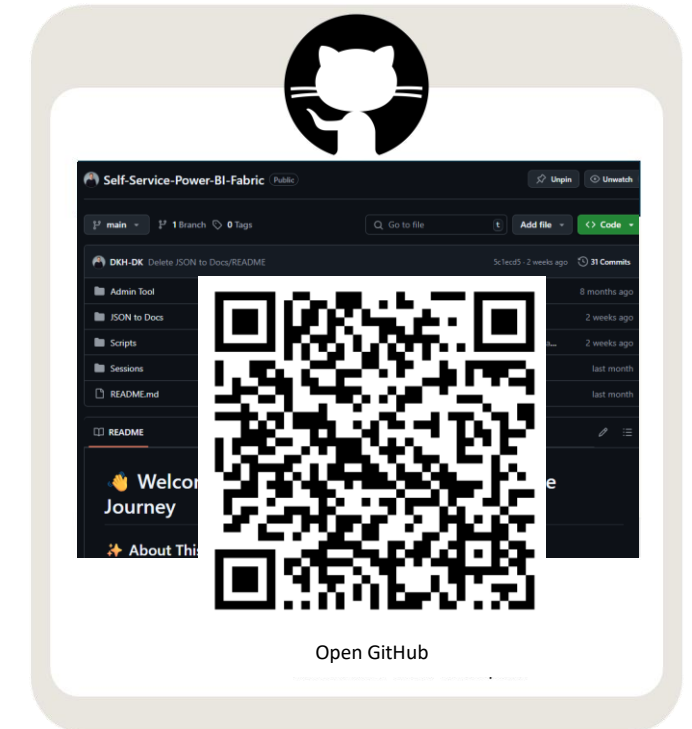
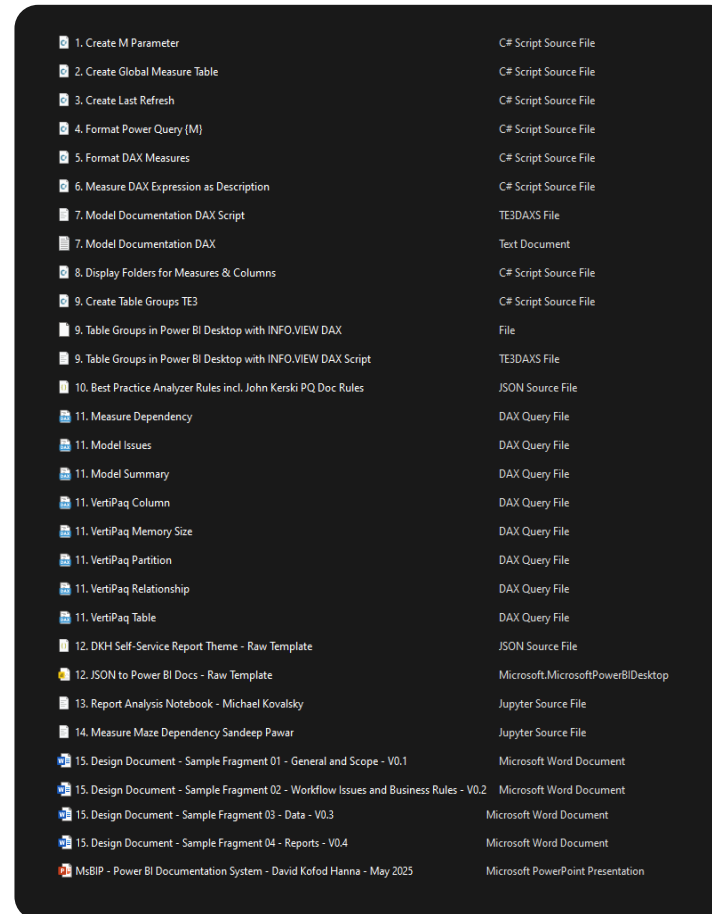


A Power BI Documentation System

– Not Just an Afterthought

**Released on
GitHub
and LinkedIn
article**

#DocumentationCanBeFun

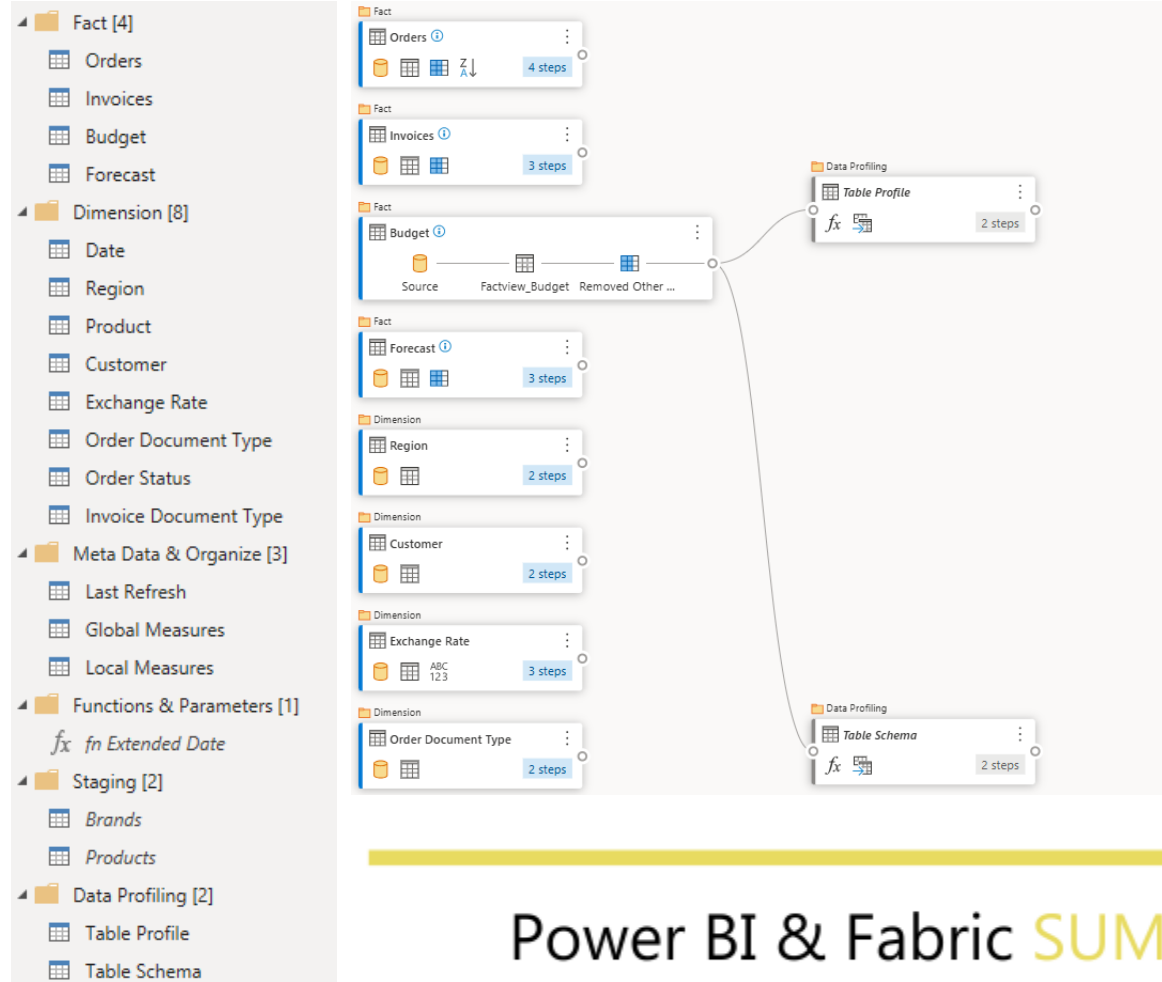




Query folders by type – and Diagram View in Dataflow Gen2



- Dimview Brands
- Dimview Budget Rate
- Dimview Customers
- Dimview Employees
- Dimview Exchange Rate
- Dimview Invoice Document Type
- Dimview Order Document Type
- Dimview Order Status
- Dimview Products
- Dimview Regions
- Factview Budget
- Factview Forecast
- Factview Invoices
- Factview Orders





Applied Steps & Properties in step



APPLIED STEPS

Source	✱
Navigation	✱
Promoted Headers	✱
Changed Type	
Added Conditional Column	✱
Changed Type1	
Added Conditional Column1	✱
Added Conditional Column2	✱
Added Conditional Column3	✱
Changed Type2	
Inserted Sum	✱
Renamed Columns	
Merged Queries	✱
Expanded Accuracy Master	✱
Added Conditional Column4	✱
Changed Type3	
Filtered Rows	✱



APPLIED STEPS

Source	✱
Navigation	✱
Promoted Headers	✱
Changed Type	
Filtered Rows on Nulls from Excel	✱
DQ Check Column: Firstname = Null	✱
DQ Check Column: Lastname = Null	✱
DQ Check Column: Mobilephone = Null	✱
DQ Check Column: e-Mail = Null	✱
DQ Check Column Sum of Nulls above	✱
Merged Queries Accuracy Master for c...	✱
Expanded Accuracy Master for clientna...	✱
DQ Check Column: Clientname = Client...	✱
Changed Type Wholenumber for DQ c...	

- View Native Query
- Diagnose
- Properties...

My Standard:

Filter as early as possible (rows and columns)

Combine similar steps into one – instead of Changed Type1, Changed Type2, Changed Type3

Rename steps that are important for yourself or colleagues to understand

- Added Conditional Column
- Merge Queries
- Append Queries

Add more documentation for tooltip info at steps by right-clicking step and choose **Properties**– if you are needing more space than in the step name.

The #"" in the steps can be avoided if you don't use space

Query Folding for SQL DB and OData connection and always **Roche's Maxim of Data Transformation**.

David Kofod Hanna • You
Senior Advisor | Academy | Speaker
3d ...

Not an excuse, but curious if the Power Query engine will optimize this in the engine call?

Like · Reply · 2 replies · 1,863 impressions

Chandeep Chhabra Author
Power BI Trainer and Consultant
3d ...

David Kofod Hanna Very hard to convincingly say - "Power Query engine will take care of it"





Table properties –
add it once and see it everywhere





Table properties – add it once and see it everywhere

Queries [22]

Dimension [8]

Customer

- Copy
- Paste
- Delete
- Rename
- Enable load
- Include in report refresh
- Duplicate
- Reference
- Move To Group
- Move Up
- Move Down
- Create Function...
- Convert To Parameter
- Advanced Editor
- Properties...

In Power Query Editor

Queries [22]

Dimension [8]

Product

Customer

Ex From CRM, ERP and ServiceNow - combined and validated by Data Governance team

Order Document type

In Power BI Desktop

Name Customer

Storage mode Import

Data refreshed 5/16/2025, 8:06:53 PM

Description From CRM, ERP and ServiceNow - combined and validated by Data Governance team

Data

- > Units
- > User Configuration
- > Customer
- > Date

In Dataflows in the service

From CRM, ERP and ServiceNow - combined and validated by Data Governance team

Customer

Source

Dimview_Custom... Marked key colu...

In TMDL

Apply Preview

```
1 createOrReplace
2
3
4
5
6
7
8
9
10
11
12
```

From CRM, ERP and ServiceNow - combined and validated by Data Governance team

table Customer

lineageTag: 726a20c4-ae07-450e-8d9f-e2b51dfa4e91

column 'Customer Key'

dataType: string

isHidden

isAvailableInMdx: false

displayFolder: Key

In INFO.VIEW DAX Functions

Model Documentation

by INFO.VIEW Functions

Filter by keyword

Location	Name	Type	Expression	Description
	Budget	Table		From IBM TMS
	Customer	Table		From CRM, ERP and ServiceNow - combined and validated by Data Governance team
	Date	Table		Derived from Melissa de Korte Extended Date Function
	Exchange Rate	Table		From purchased source from web
	Global Measures	Table		Store Global measures from Semantic Model Developer
	Invoice Document Type	Table		MS F&O
	Last Refresh	Table		Last refresh of semantic model - not the underlying job and pipelines
	Local Measures	Table		Store Local Report Level Measures
	Order Document Type	Table		MS F&O
	Order Status	Table		MS F&O

In OneLake catalog

Semantic model

Power BI Docs - System

Open

Overview Lineage Monitor Permissions

Location Power BI & Fabric Showcase

Refreshed 19/05/25, 13:35:28

Owner David Kofod Hanna

Tables

Name	Type	Description
> Customer	Table	From CRM, ERP and ServiceNow - combined and validated by ...
> Date	Table	Derived from Melissa de Korte Extended Date Function
> Exchange Rate	Table	From purchased source from web
> Global Measures	Table	Store Global measures from Semantic Model Developer





Model Docs DAX INFO.VIEW



"There's a script for that"

Model Documentation

by INFO.VIEW Functions

Filter by keyword



Column

Measure

Relationship

Table

Location	Name	Type	Expression	Description
Global Measures	Global Measures	Measure	"Locate them here my semantic model developer friend"	
Global Measures	Sum of Delivery Cost	Measure	SUM('Invoices'[Delivery Cost])	This measure is the sum of column 'Invoices'[Delivery Cost]
Global Measures	Sum of Forecast (EUR)	Measure	SUM('Forecast'[Forecast (EUR)])	This measure is the sum of column 'Forecast'[Forecast (EUR)]
Global Measures	Sum of Freight	Measure	SUM('Invoices'[Freight])	This measure is the sum of column 'Invoices'[Freight]
Global Measures	Sum of Late Delivery Penalti...	Measure	SUM('Invoices'[Late Delivery Penalties])	This measure is the sum of column 'Invoices'[Late Delivery Penalties]
Global Measures	Sum of Net Invoice COGS	Measure	SUM('Invoices'[Net Invoice COGS])	This measure is the sum of column 'Invoices'[Net Invoice COGS]
Global Measures	Sum of Net Invoice Cost	Measure	SUM('Invoices'[Net Invoice Cost])	This measure is the sum of column 'Invoices'[Net Invoice Cost]
Global Measures	Sum of Net Invoice Quantity	Measure	SUM('Invoices'[Net Invoice Quantity])	This measure is the sum of column 'Invoices'[Net Invoice Quantity]
Global Measures	Sum of Net Invoice Value	Measure	SUM('Invoices'[Net Invoice Value])	This measure is the sum of column 'Invoices'[Net Invoice Value]
Global Measures	Sum of Net Order Quantity	Measure	SUM('Orders'[Net Order Quantity])	This measure is the sum of column 'Orders'[Net Order Quantity]
Global Measures	Sum of Net Order Value	Measure	SUM('Orders'[Net Order Value])	This measure is the sum of column 'Orders'[Net Order Value]
Global Measures	Sum of Total Budget	Measure	SUM('Budget'[Total Budget])	This measure is the sum of column 'Budget'[Total Budget]
Local Measures	Local Measures	Measure	"Locate them here my report developer friend"	
Table Group DAX	Number of tables	Measure	COUNTROWS('Table Group DAX')	



INFO.VIEW DAX Functions make this possible

INFO.VIEW.COLUMNS()

INFO.VIEW.MEASURES()

INFO.VIEW.TABLES()

INFO.VIEW.RELATIONSHIPS()

```
1 Model Documentation =
2 VAR _columns =
3     SELECTCOLUMNS(
4         FILTER(
5             INFO.VIEW.COLUMNS( ),
6             [Table] <> "Model Documentation" && NOT ( [IsHidden] )
7         ),
8         "Type", "Column",
9         "Name", [Name],
10        "Description", [Description],
11        "Location", [Table],
12        "Expression", [Expression]
13    )
14 VAR _measures =
15     SELECTCOLUMNS(
16         FILTER(
17             INFO.VIEW.MEASURES( ),
18             [Table] <> "Model Documentation" && NOT ( [IsHidden] )
19         ),
20         "Type", "Measure",
21         "Name", [Name],
22         "Description", [Description],
23         "Location", [Table],
24         "Expression", [Expression]
25    )
26 VAR _tables =
27     SELECTCOLUMNS(
28         FILTER(
29             INFO.VIEW.TABLES( ),
30             [Name] <> "Model Documentation" && [Name] <> "Calculations"
31             && NOT ( [IsHidden] )
32         ),
33         "Type", "Table",
34         "Name", [Name],
35         "Description", [Description],
36         "Location", BLANK( ),
37         "Expression", [Expression]
38    )
39 VAR _relationships =
40     SELECTCOLUMNS(
41         INFO.VIEW.RELATIONSHIPS( ),
42         "Type", "Relationship",
43         "Name", [Relationship],
44         "Description", BLANK( ),
45         "Location", BLANK( ),
46         "Expression", [Relationship]
47    )
48 RETURN
49 UNION( _columns, _measures, _tables, _relationships )
```

Display folders for columns



"There's a script for that"

Factview Invoices

- Billing Date
- Σ Billing Document Line Item Number
- Σ Billing Document Number
- Billing Document Type Code
- Customer Key
- Σ Delivery Cost
- DWCreatedDate
- Σ Freight
- Σ Late Delivery Penalties
- Local Currency
- Σ Net Invoice COGS
- Σ Net Invoice Cost
- Σ Net Invoice Quantity
- Σ Net Invoice Value
- OTD Indicator
- Σ Overdue Payment Penalties
- Product Key
- Ship Date
- Σ Taxes & Commercial Fees

Invoices

- > Attributes
- > Dates
- > Flags
- > Key
- > Numeric

```
//Go through each table in the model
foreach(var table in Model.Tables)
{
    if(table.Name != "Date"){
        //First look at columns
        foreach( var column in table.Columns)
        {
            var keySuffix = "Key";
            var columnDataType = column.DataType.ToString();
            //DWCreatedDate column should be hidden in a separte folder
            if( column.Name == "DWCreatedDate")
            {
                column.DisplayFolder = "Attributes\\Metadata";
                column.IsHidden = true;
            }
            //Numeric columns should not be aggregated and float (double) data type should not be used
            if(column.DataType == DataType.Double || column.DataType == DataType.Decimal || column.DataType == DataType.Int64)
            {
                column.DisplayFolder = "Numeric";
                column.SummarizeBy = AggregateFunction.None;
                if(column.DataType == DataType.Double)
                {
                    column.DataType = DataType.Decimal;
                }
            }
            //Boolean data types into their own folder
            if(column.DataType == DataType.Boolean)
            {
                column.DisplayFolder = "Flags";
            }
            if(column.DataType == DataType.String)
            {
                column.DisplayFolder = "Attributes";
            }
            //Keys go into their own display folder, should not be aggregated and hidden.
            if(column.UsedInRelationships.Any())
            {
                column.DisplayFolder = "Key";
                column.SummarizeBy = AggregateFunction.None;
                column.IsHidden = true;
            }
            //Date keys get their own folder and other dates go in attributes
            if( column.DataType == "DateTime" && column.Name != "DWCreatedDate")
            {
                if(column.UsedInRelationships.Any())
                {
                    column.DisplayFolder = "Key";
                    column.IsHidden = true;
                }
                else{
                    column.DisplayFolder = "Dates";
                }
            }
        }
    }
}
```

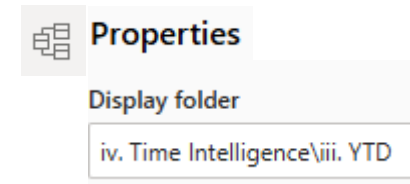


Display folders in Global and Local Measure Groups



- ✓ i. Base
 - > Budget
 - > Forecast
 - > Invoice
 - > Order
- > ii. KPI Targets
- > iii. Comparison
- ✓ iv. Time Intelligence
 - > i. MTD
 - > ii. QTD
 - > iii. YTD
 - > iv. LY
 - > v. LY %
 - > vi. MoM
 - > vii. QoQ
 - > viii. YoY
- > v. Ratio
- ✓ vi. Report Layout
 - > i. Last Refresh
 - > ii. HEX Colors
 - > iii. Dynamic Titles
 - > iv. SVG's
 - > v. Cascading Slicers (Alberto)

To create subfolder use \



To order folders by ...

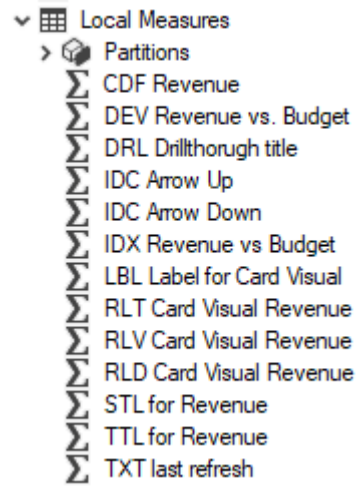
Number	Roman Numeral
1	i
2	ii
3	iii
4	iv
5	v
6	vi
7	vii
8	viii
9	ix
10	x



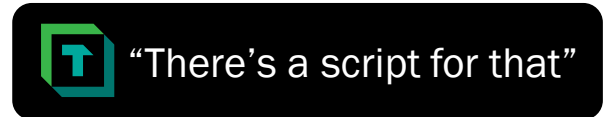
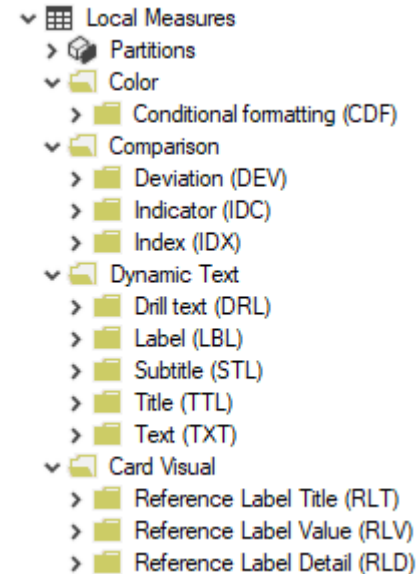


Display folders

If you use consistent naming of measures



It's easier to search for and structure in display folders



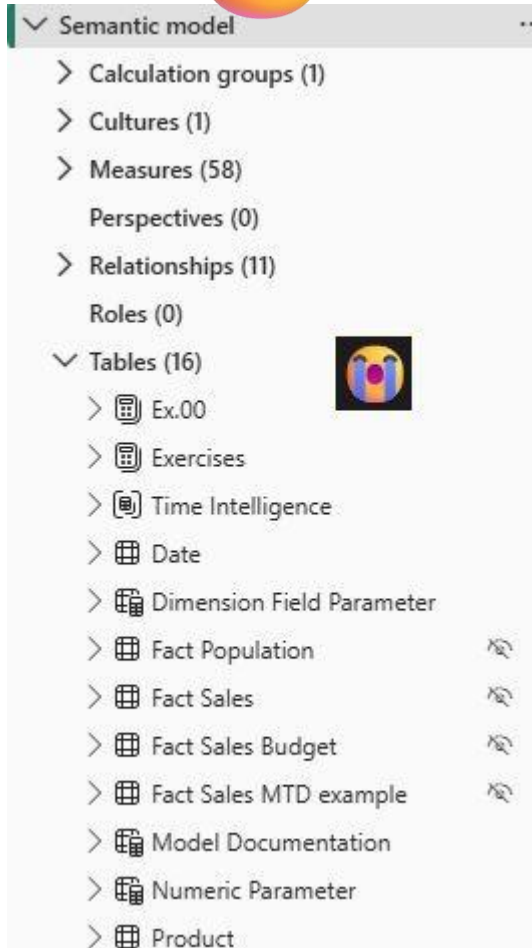
```
// -----
// Prefix-to-Metadata Mapping Table
// -----
// Each entry contains:
//   Prefix = Expected start of the measure name
//   Group1 = Top-level display folder name
//   Group2 = Second-level display folder name
//   Desc = Human-readable description
// -----
var mappings = new[]
{
    new { Prefix = "CDF", Group1 = "Color", Group2 = "Conditional formatting (CDF)", Desc = "Conditional formatting" },
    new { Prefix = "DEV", Group1 = "Comparison", Group2 = "Deviation (DEV)", Desc = "Deviation" },
    new { Prefix = "DRL", Group1 = "Dynamic Text", Group2 = "Drill text (DRL)", Desc = "Drill text (enable/disable)" },
    new { Prefix = "IDC", Group1 = "Comparison", Group2 = "Indicator (IDC)", Desc = "Indicator" },
    new { Prefix = "IDX", Group1 = "Comparison", Group2 = "Index (IDX)", Desc = "Index" },
    new { Prefix = "LBL", Group1 = "Card Visual", Group2 = "Label (LBL)", Desc = "Label" },
    new { Prefix = "RLT", Group1 = "Card Visual", Group2 = "Reference Label Title (RLT)", Desc = "Reference Label Title" },
    new { Prefix = "RLV", Group1 = "Card Visual", Group2 = "Reference Label Value (RLV)", Desc = "Reference Label Value" },
    new { Prefix = "RLD", Group1 = "Card Visual", Group2 = "Reference Label Detail (RLD)", Desc = "Reference Label Detail" },
    new { Prefix = "STL", Group1 = "Dynamic Text", Group2 = "Subtitle (STL)", Desc = "Subtitle" },
    new { Prefix = "TTL", Group1 = "Dynamic Text", Group2 = "Title (TTL)", Desc = "Title" },
    new { Prefix = "TXT", Group1 = "Dynamic Text", Group2 = "Text (TXT)", Desc = "Text" },
    new { Prefix = "HEX", Group1 = "Color", Group2 = "HEX Color (HEX)", Desc = "HEX Color (HEX)" },
    new { Prefix = "SVG", Group1 = "Card Visual", Group2 = "Scalable Vector Graphics (SVG)", Desc = "Scalable Vector Graphics (SVG)" },
};

// -----
// Main Processing Loop
// -----
// For each measure:
// 1. Check whether its name starts with a known prefix.
// 2. When matched:
//    - Set DisplayFolder = "Group1\Group2"
//    - Set Description
// -----
foreach (var measure in Model.AllMeasures)
{
    foreach (var map in mappings)
    {
        // Prefix matching supports "ABC." or "ABC"
        if (measure.Name.StartsWith(map.Prefix + ".", StringComparison.OrdinalIgnoreCase) ||
            measure.Name.StartsWith(map.Prefix, StringComparison.OrdinalIgnoreCase))
        {
            measure.DisplayFolder = map.Group1 + "\\\\" + map.Group2;
            measure.Description = map.Desc;
            break; // Stop searching once mapping is found
        }
    }
}
```





Table Groups in Tabular Editor 3



"There's a script for that"

```
// ***** CREATE TABLE GROUPS *****
// Loop through all tables in the model:
foreach (var table in Model.Tables)
{
    if (table is CalculationGroupTable)
    {
        // Assign table group for calculation groups:
        table.TableGroup = "Calculation Groups";
    }
    else if (!table.UsedInRelationships.Any() && table.Measures.Any(m => m.IsVisible))
    {
        // Tables containing visible measures, but no relationships to other tables:
        table.TableGroup = "Measure Groups";
    }
    else if (table.UsedInRelationships.All(r => r.FromTable == table) && table.UsedInRelationships.Any())
    {
        // Tables exclusively on the "many" side of relationships:
        table.TableGroup = "Facts";
    }
    else if (!table.UsedInRelationships.Any() && table is CalculatedTable && !table.Measures.Any())
    {
        // Tables without any relationships that are calculated tables
        // and do not have measures:
        table.TableGroup = "Parameter Tables";
    }
    else if (table.UsedInRelationships.Any(r => r.ToTable == table))
    {
        // Tables on the "one" side of relationships:
        table.TableGroup = "Dimensions";
    }
    else
    {
        // All other tables:
        table.TableGroup = "Misc";
    }
}
```

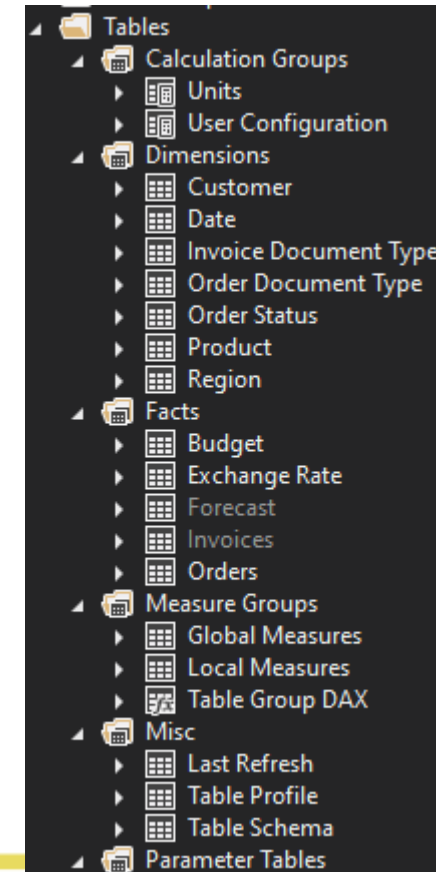
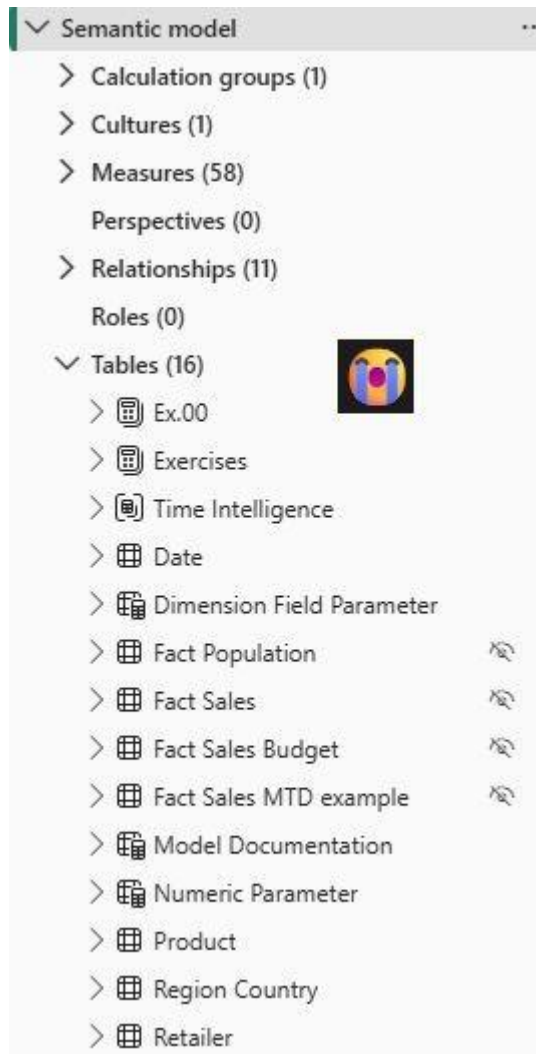




Table Group in Power BI Desktop with INFO.VIEW



“There’s a script for that”

Description	Type	Order	Table Name
Classified as Measure Group	Measure Group	1	Exercises
Classified as Measure Group	Measure Group	1	Ex.00
Fact with many-side relationship	Fact	2	Fact Sales
Fact with many-side relationship	Fact	2	Fact Sales Budget
Fact with many-side relationship	Fact	2	Fact Population
Fact with many-side relationship	Fact	2	Fact Sales MTD example
Dimension with one-side relationship	Dimension	3	Sales Size
Dimension with one-side relationship	Dimension	3	Retailer
Dimension with one-side relationship	Dimension	3	Product
Dimension with one-side relationship	Dimension	3	Date
Dimension with one-side relationship	Dimension	3	Region Country
Dynamic calculation items	Calculation Group	4	Time Intelligence
Dynamic switch between measure or attributes	Field Parameters	5	Dimension Field Parameter
Dynamic slider for end users to select	Numeric Parameter	6	Parameter
Documentation with INFO.VIEW functions	Model Documentation	7	Model Documentation

Semantic Model Information

Table Groups in Power BI Desktop

Using INFO.VIEW DAX Functions & some creativity

Type	Number of tables
Calculation Group	1
User Configuration	1
Dimension	7
Customer	1
Date	1
Invoice Document Type	1
Order Document Type	1
Order Status	1
Product	1
Region	1
Fact	6
Budget	1
Customer	1
Exchange Rate	1
Forecast	1
Invoices	1
Orders	1
Measure Group	6
Global Measures	1
Last Refresh	1
Local Measures	1
Table Profile	1
Table Schema	1
Units	1
Total	21

Calculation Group	Dimension	Fact	Measure Group	Model Document...
-------------------	-----------	------	---------------	-------------------

Rule Specifications

① Measure Group

An “empty” table to store my measures

If not below tables, no relationships, hidden columns, visible measure

② Fact

Fact table with many-side relationship *->1

SELECTCOLUMNS (INFO.VIEW.RELATIONSHIPS(), Table name from [FromTable])

③ Dimension

Dimension table with one-side relationships 1->* *

SELECTCOLUMNS (INFO.VIEW.RELATIONSHIPS(), Table name from [ToTable])

④ Calculation Group

Dynamic calculation items with SELECTEDMEASURE()

IF (INFO.VIEW.TABLES() [CalculationGroupPrecedence] >= 1)

⑤ Field Parameter

Dynamic switch between measure or attributes

IF (INFO.VIEW.TABLES() CONTAINSSTRING(Expression), "NAMEOF")

⑥ Numeric Parameter

Dynamic slider for end users to select

IF (INFO.VIEW.TABLES() CONTAINSSTRING(Expression), "GENERATE")

⑦ Model Documentation

Documentation with INFO.VIEW functions

IF (INFO.VIEW.TABLES() CONTAINSSTRING(Expression), "INFO.VIEW")





Report Visual Elements

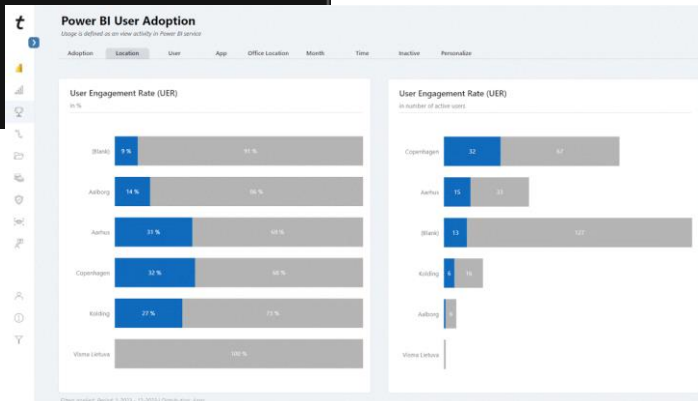
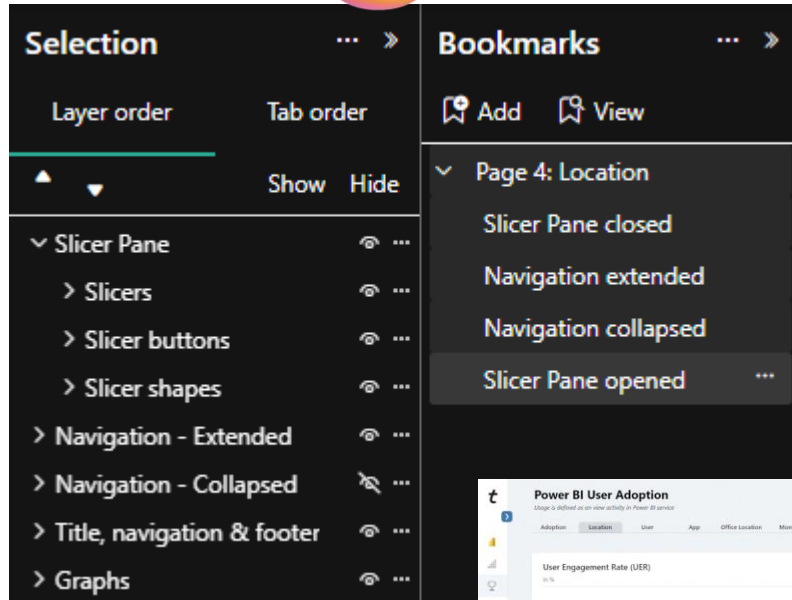


Selection		...	>>
Layer order		Tab order	
▲ ▼		Show	Hide
Slicer		🔍 ...	
Slicer		🔍 ...	
Slicer		🔍 ...	
Button		🔍 ...	
Button		🔍 ...	
Button		🔍 ...	
Button		🔍 ...	
Data last refreshed:		🔍 ...	





Report Visual Elements – especially important if using bookmarks



💡 My Standard:

For non-visual objects:

Use the visual object as Prefix + "-" and then a meaningful name for the object "Slicer – Year" or "Image – Logo"

Relevant for: Slicer, Textbox, Shapes, Images, Button, Page Navigator, Bookmark Navigator, etc.

For **visual elements** the name of the object is the Title in the visual, thus can't be prefixed with Bar chart, etc.

Group visual elements into e.g.

"Slicer Pane",

"Title, Navigation & Footer",

"KPI's"

"Graphs".

Layer order from top left corner to bottom right corner (F or Z structure)

For use cases where only **Bookmarks** can solve the report user need and counting the total cost of ownership, use Selected Visuals and the Group of visuals.

Group Bookmarks together and name it prefix of page no and page name **"Page 4: Sales Region Analysis"**





DAX Query View for testing



DAX queries will be saved to your model. They won't be visible when published in the Power BI service. [Learn more](#)

Run Update model with changes (0)

```

1 // Author - Hariharan Rajendran
2 // Name - Model Summary in DAX Query View
3 // Version - 1.0
4 // Contact - https://www.linkedin.com/in/hariharan/
5 // This query will provide details about your model like property, value and comment to execute the DAX query to see the complete list
6 // Define the properties
7 EVALUATE
8   VAR _tableCount = COUNTROWS(INFO.TABLES())
9   VAR _columnCount = COUNTROWS(INFO.COLUMNS())
10  VAR _calcColumnCount = COUNTROWS(FILTER(
11    INFO.COLUMNS(),
12    [Type] = 2
13  ))
14  VAR _measures = COUNTROWS(INFO.MEASURES())
15  VAR _relationships = COUNTROWS(INFO.RELATIONSHIPS())
16  VAR _Composite = IF(
17    COUNTROWS(SUPPRESS(
18      INFO.VIEW.TABLES(),
19      [StorageNode]
20    )) > 1,
21    "Yes",
22    "No"
23  )
24  VAR _daxTables = COUNTROWS(FILTER(

```

Model Summary

	[Property]	[Value]	[Comment]
1	Tables	21	EVALUATE INFO.TABLES()
2	Columns	197	EVALUATE INFO.COLUMN...
3	--CalculatedColumns	0	EVALUATE FILTER(INFO...
4	--DirectColumns	197	EVALUATE FILTER(INFO...
5	Measures	17	EVALUATE INFO.MEASU...
6	Relationships	13	EVALUATE INFO.RELATI...
7	IsCompositeModel	No	EVALUATE INFO.VIEW.T...
8	DAXTables	2	EVALUATE FILTER(INFO...
9	IsPartitioned	No	EVALUATE FILTER(INFO...
10	Perspectives	0	EVALUATE FILTER(INFO...
11	Calculation Groups	2	EVALUATE FILTER(INFO...
12	Roles	0	EVALUATE FILTER(INFO...
13	PBIDesktopVersion	2.142.1277.0 (25.04)+014fde45...	EVALUATE FILTER(INFO...

Model Issues

	[Property]	[Value]	[Comment]	[Status]
1	Is Partition Required?	Yes	EVALUATE VAR _dt_parti...	Fail
2	Dedicated Date Table	No	EVALUATE FILTER(INFO...	Fail
3	Is there any Redundant Columns	Yes	EVALUATE FILTER(GROU...	Fail
4	Are descriptions added?	No	EVALUATE FILTER(INFO...	Fail
5	Calculated Columns		EVALUATE FILTER(INFO...	Pass
6	Local Date Tables (Auto Time Intelligence)	0	EVALUATE FILTER(INFO...	Pass
7	In Active Relationships	0	EVALUATE FILTER(INFO...	Pass
8	Bi-directional Relationships	0	EVALUATE FILTER(INFO...	Pass
9	Many to Many Relationships	0	EVALUATE FILTER(INFO...	Pass

Measure Dependency

	[MeasureName]	[Expression]	[SourceMeasure]	[SourceExpression]	[Type]	[ReferenceTable]	[ReferenceObject]
1	Net Invoice Value YTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Invo...	TABLE	Invoices	Invoices
2	Net Invoice Value YTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Invo...	COLUMN	Invoices	Net Invoice Value
3	Net Invoice Value MTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Invo...	TABLE	Invoices	Invoices
4	Net Invoice Value MTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Invo...	COLUMN	Invoices	Net Invoice Value
5	Net Invoice Value QTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Invo...	TABLE	Invoices	Invoices
6	Net Invoice Value QTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Invo...	COLUMN	Invoices	Net Invoice Value
7	Number of tables	COUNTROWS('Table Gr...			CALC_TABLE	Table Group DAX	Table Group DAX
8	Sum of Total Budget	SUM('Budget'[Total Bud...			TABLE	Budget	Budget
9	Sum of Total Budget	SUM('Budget'[Total Bud...			COLUMN	Budget	Total Budget
10	Sum of Net Order Quan...	SUM('Orders'[Net Order...			TABLE	Orders	Orders

VertiPaq Memory Size

	[Size in MB]
1	1017.25

Visible in .PBIP and for easy copy/paste

System.SemanticModel > DAXQueries >

.pbi	File folder
DAXQueries	DAX Query File
.platform	DAX Query File
definition.pbism	DAX Query File
diagramLayout	DAX Query File
model	DAX Query File
Measure Dependency	DAX Query File
Model Issues	DAX Query File
Model Summary	DAX Query File
VertiPaq Columns	DAX Query File
VertiPaq Memory Size	DAX Query File
VertiPaq Partitions	DAX Query File
VertiPaq Relationships	DAX Query File
VertiPaq Table	DAX Query File



Run DAX Query and save as delta table in Lakehouse 🤖

DAX Query x Semantic Link Labs x DAX INFO Functions

Purpose of notebook is to collect, analyze and store semantic model INFO functions and DAX Queries in a Lakehouse and share with semantic model or report owners or "crazy solution documenters"

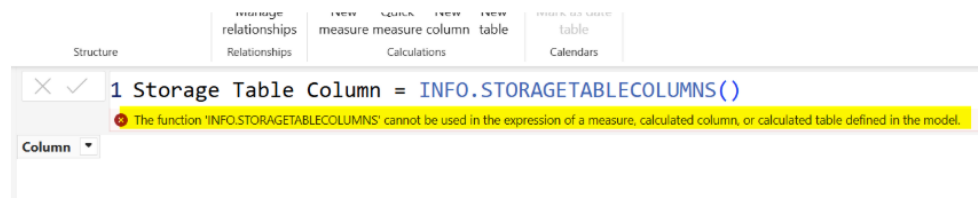
This can be run in a Python notebook

Credit to following

- Michael Kovalsky - Semantic Link Labs
- List of all DAX INFO functions
- Fourmoo on Python Notebook advantages vs spark notebooks
- Hariharan Rajendran's DAX Query on Models Issues and Model Summary
- Power BI Tips on Save to Delta table with Python
- Reza Rad's take on Power BI model analysis using DAX INFO functions
- David Kofod Hanna's GitHub on Power BI Documentation System

If you feel frustrated like me, not all DAX Info functions can be used like INFO.VIEW functions by adding a calculated DAX table.

Then we have been forced to used DAX Query View.



Save your frustration, because I found a way to store DAX Query results back to a lakehouse 🤖

Manual DAX Query

```
In [ ]: # Define DAX Query to execute
dax_query = """
EVALUATE
INFO.TABLES()
"""

In [ ]: # Run DAX Query against Semantic Model
df_result_manual = fabric.evaluate_dax(
    dataset=SemanticModelName,
    dax_string=dax_query,
    workspace=WorkspaceName
)

# Display the result
display(df_result_manual)

In [ ]: labs.save_as_delta_table(
    dataframe=df_result_manual,
    delta_table_name="daxquery",
    write_mode="overwrite", # or "append" based on your requirement
    merge_schema=False, # Set to True if you want to merge schema
    schema=None, # Provide schema if needed
    lakehouse=LakehouseName,
    workspace=None
)
```

Automated DAX Query loop and save to delta table

```
In [ ]: # Defining DF of all INFO DAX Functions and descriptions

dataDAX = [
    ("INFO.ALTERNATEOFDEFINITIONS()", ""),
    ("INFO.ANNOTATIONS()", "Returns a list of all annotations in t"),
    ("INFO.ATTRIBUTEHIERARCHIES()", "Represents the TMSHEMA_ATTRI"),
    ("INFO.ATTRIBUTEHIERARCHYSTORAGES()", ""),
    ...
]
```





JSON Report Theme – For the Nerdy Data Storytellers

Home / JSON to Power BI

JSON Report Theme to Power BI

Power BI custom report themes provide granular control over many aspects of a report theme. It's crucial to build a solid, documented and adopted in an organization for consistency and efficiency.

Theme colors in Power BI

Power BI's data color system lets you define up to 8 colors that map directly to your data points. Along with these colors, Power BI automatically generates 5 different shades of each color that can be dynamically applied throughout the file.

KPI & Divergent Colors

These properties set the status colors used by the waterfall chart and the KPI visual or set the various gradient colors in the conditional formatting dialog box.

Use them to refer in conditional formatting measures where you can simply in the measure, instead of hard-coding a HEX, use "good", "neutral", "bad", "minimum" or "maximum"

Structural Colors

These color classes set the structural colors for elements in the report, such as axis gridlines, highlight colors, and background colors for visual elements.

Active Report Theme:
DKH Self-Service Report Theme

Theme Colors 1-8

For data visualization

Theme	HEX	ID
	#367cff	1
	#fa8100	2
	#6cc6cb	3
	#aa77dd	4
	#d14576	5
	#b26d6d	6
	#8b9fd4	7
	#eae5c9	8

Learn more

Remember to think of brand guidelines together with enough categorical values along side thinking contrast and call-to-action and WCAG and color-blindness.

KPI Colors

For KPI performance status context

KPI	HEX	Property
	#37a78f	Good
	#f2f2f2	Neutral
	#a74d37	Bad

Divergent Min & Max Colors

For divergent heat map

Divergent	HEX	Property
	#fa8100	Minimum
	#f2f2f2	Center
	#367cff	Maximum
		Null

Example measure diff Color =
IF (
[Revenue LY %] > 0,
"good",
"bad")

Learn more

Structural Colors

Non-data ink for colors except the data colors

Structural	HEX	Property
	#485257	Background
	#f1f3f4	BackgroundLight
	#f1f3f4	BackgroundNeutral
	#485257	Foreground
	#485257	ForegroundNeutralSecondary
	#606e74	ForegroundNeutralTertiary
		Hyperlink
	#e6e6e6	TableAccent
		Visited Hyperlink

Learn more

Any formatting elements that aren't included in the JSON file revert to their default values and settings.

Try yourself?
Download at [GitHub](#)

Visual Styles

15 of 52 visual styles defined in theme

Image	VisualStyleAttribute	Defined in JSON Theme	Type
	textbox	textbox	Object
	tableEx	tableEx	Visual
	slicer	slicer	Visual
	shape	shape	Object
	report	report	Object
	pivotTable	pivotTable	Visual
	pageNavigator	pageNavigator	Object
	page	page	Object
	multiRowCard	multiRowCard	Visual
	kpi	kpi	Visual
	image	image	Object
	cardVisual	cardVisual	Visual
	bookmarkNavigator	bookmarkNavigator	Object
	advancedSlicerVisual	advancedSlicerVisual	Visual
	actionButton	actionButton	Object
	aiNarratives		Visual
	areaChart		Visual
	azureMap		Visual
	barChart		Visual
	card		Visual
	clusteredBarChart		Visual
	clusteredColumnChart		Visual
	columnChart		Visual
	decompositionTreeVisual		Visual
	donutChart		Visual
	filledMap		Visual
	filter		Object
	funnel		Visual
	gauge		Visual
	group		Object
	hundredPercentStackedAreaChart		Visual
	hundredPercentStackedBarChart		Visual
	hundredPercentStackedColumnChart		Visual

Visual Styles



Be aware of the new preview visuals as the report theme schema can change more often on these.

Recommendation:

Set Visual Styles for frequently used visuals like

- Card New Visual
- Matrix
- Table
- Line chart
- Column chart
- Bar chart
- Slicer
- Action button
- Page Navigator

Attribute	Property	Value
Filter by keyword	All	Filter by keyword

Detail Properties of Visual Styles in Your JSON Theme

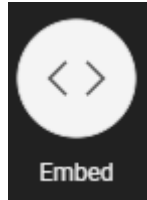
Click on a Visual Style to filter

Image	Name	Attribute	Property	Value
	bookmarkNavigator	Sid	selected	selected
		shape	default	default
		Text	selected	selected
		Color	#AEB8BD	#AEB8BD
		bottom	*	0
		color	Border	#AEB8BD
		fillColor	fill	#F1F3F4
		fontColor	Text	#30373A
		fontFamily	Text	Segoe UI Semibold
		fontSize	Text	10.5
		left	*	0
		radius	Border	8
		right	*	0
		roundEdge	shape	8
		show	Border	FALSE
		fill	TRUE	TRUE
		Outline	FALSE	FALSE
		tileShape	shape	rectangle





Embed videos



Medieval helpdesk with English subtitles

Watch Later Share

Watch on YouTube

it's closed and everything's save inside it.

Embed Video

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
pQHx-SjgVvQ?si=n2JMTj9g4Hi0fptp"
title="YouTube video player"
frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture; web-
share" referrerpolicy="strict-
origin-when-cross-origin"
allowfullscreen></iframe>
```

☐ Start at 0:01

EMBED OPTIONS

Copy



New link

To include a link to an item or website, enter the URL.

* required fields

Link name *

Medieval Help Desk

URL address *

https://www.youtube.com/embed/pQHx-S

Include https:// at the beginning of the URL

Link behavior

- ☐ Open link in new browser tab
- ☒ Embed linked content into app

Create

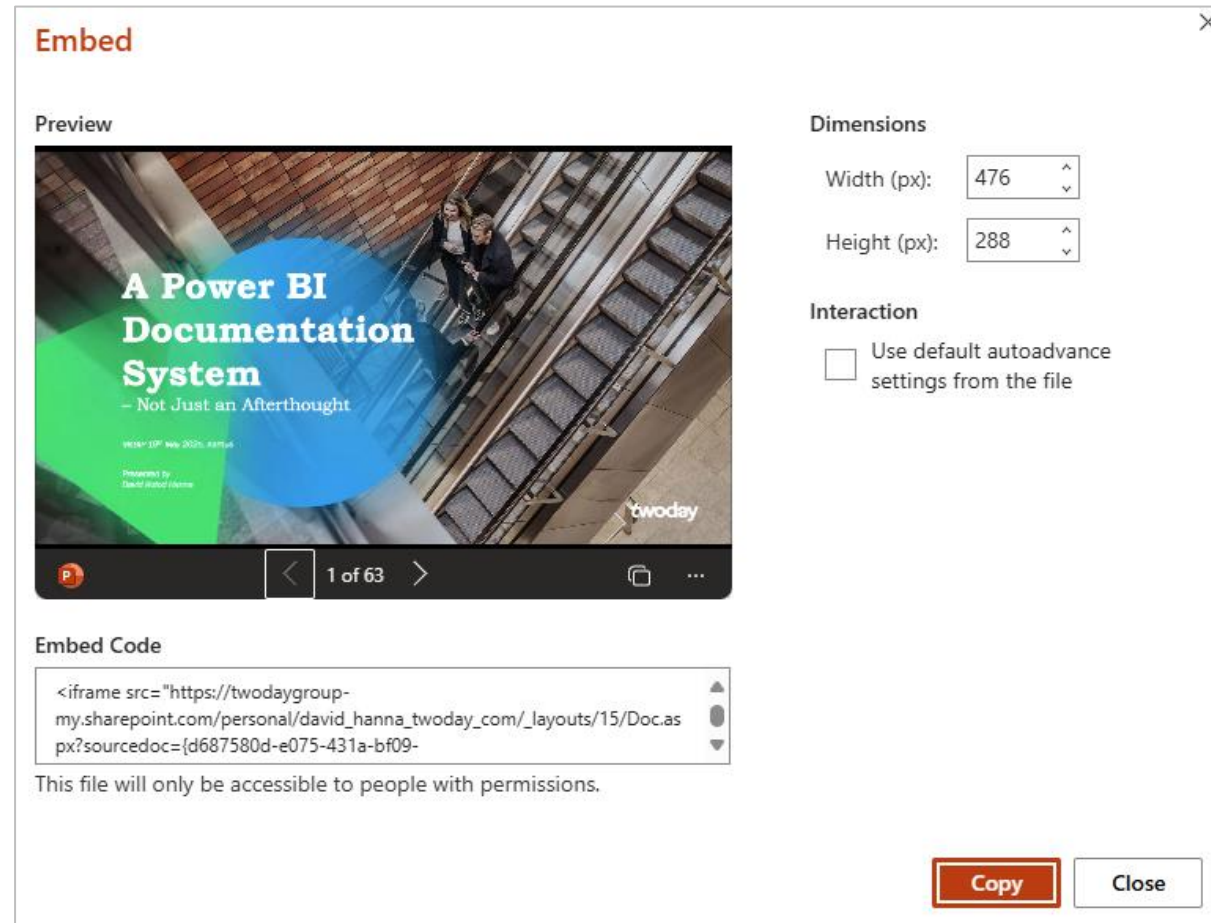
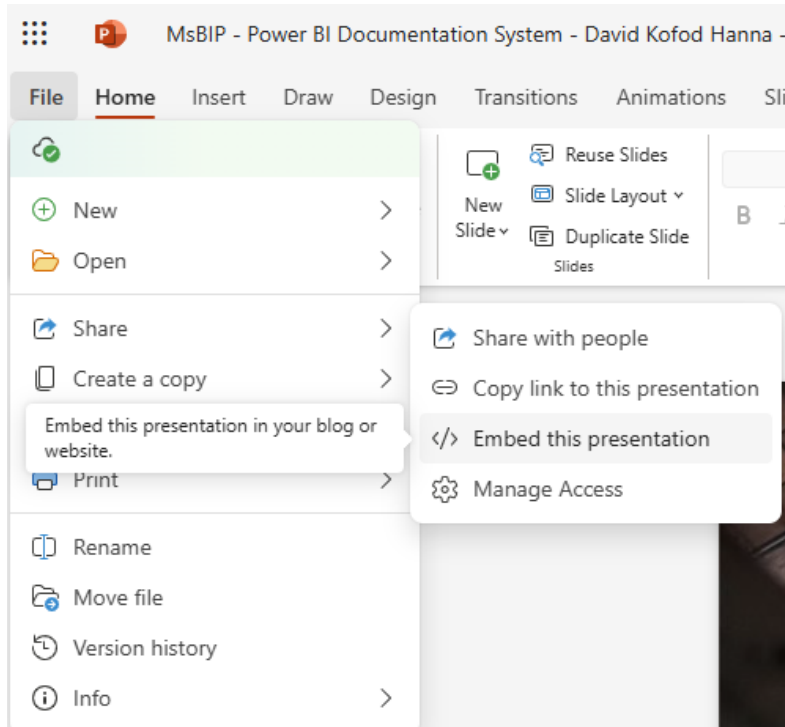
Cancel

[Helpdesk support back in the day of the middle age](#)





Embed PowerPoint, Excel, PDF





Design Documents – Samples

alexbadu-insightsinmotion / PBI-Documentation

- Design Document - Sample Fragment 01 - General and Scope - V0.1.docx
- Design Document - Sample Fragment 02 - Workflow Issues and Business Rules - V0....
- Design Document - Sample Fragment 03 - Data - V0.3.docx
- Design Document - Sample Fragment 04 - Reports - V0.4.docx
- Design Document - Sample Fragment 05 - Validation - V0.5.docx
- Design Document - Sample Fragment 06 - Deployment - V0.6.docx
- Design Document - Sample Validation Spreadsheet - V0.5.xlsx

Power BI Documentation – Design Document
Sample Fragment 04 – Reports

Table of Contents	
1. Introduction	3
2. Scope of Work	3
3. Workflow	3
4. Issues	3
5. Business Rules	3
6. Data	3
7. Reports	3
7.1. Common	3
7.1.1. Theme	3
7.1.2. Filters	4
7.1.3. Navigation	5
7.1.4. Page Header	6
7.1.5. Page Footer	7
7.1.6. Slicers	7
7.2. Specific	8
7.2.1. Semantic Model (specific)	8
7.2.2. AR01 – All Invoices	9
7.2.3. AR02 – Current Invoices	10
7.2.4. AR03 – Upcoming Invoices	10
7.2.5. AR04 – Historical Invoices	11

The items that will be included are described below.

ID	Name (Category / Subcategory)	Design / Selected / Unselected / Hover
N-1	Invoices	DESIGN: <ul style="list-style-type: none"> Type=button Shape=any, with border=off Action=page navigation (subcategory 1) DEFAULT (selected): <ul style="list-style-type: none"> Font=Segoe UI, white, 10 pt Background=dark blue Navigation=page, Invoices-All DEFAULT (unselected): <ul style="list-style-type: none"> Font=Segoe UI, medium grey, 10 pt Background=medium blue Navigation=page, Invoices-All HOVER: <ul style="list-style-type: none"> Font=Segoe UI, dark grey, 11 pt Background=medium grey Navigation=page, Invoices-All
N-2	Invoices / All	(same as N-1 above)
N-3	Invoices / Current	(same as N-1 above, but with adjusted page navigation and selected and unselected defaults reversed)
N-4	Invoices / Upcoming	(same as N-1 above, but with adjusted page navigation and selected and unselected defaults reversed)
N-5	Invoices / Historical	(same as N-1 above, but with adjusted page navigation and selected and unselected defaults reversed)

7.1.6. Slicers

The slicers that will be included on each page of each all report are described below.

ID	Slicer	Design Features / Data / Notes
n/a	General	Title=Segoe UI Semibold, 10 pt Values=Segoe UI, 8 pt Style=dropdown Selection=multi-select; CTRL off; Select All off Header icons=off Search box=enabled
S-1	Fiscal year	Data=Dates[Fiscal Year] Notes=search box unavailable as numeric data
S-2	Fiscal quarter	Data=Dates[Fiscal Quarter] Notes=search box unavailable as numeric data
S-3	Date range	Type=between Slider=on, responsive off Data=Dates[Date]
S-4	Province	Type=text Data=Countries[Province]





Embed Notebooks for technical documentation

```
1 #Author and MVP: Sandeep Pawar
2 #https://fabric.guru/measure-maze-visualizing-measure-dependencies-using-semantic-link-network-analysis
```

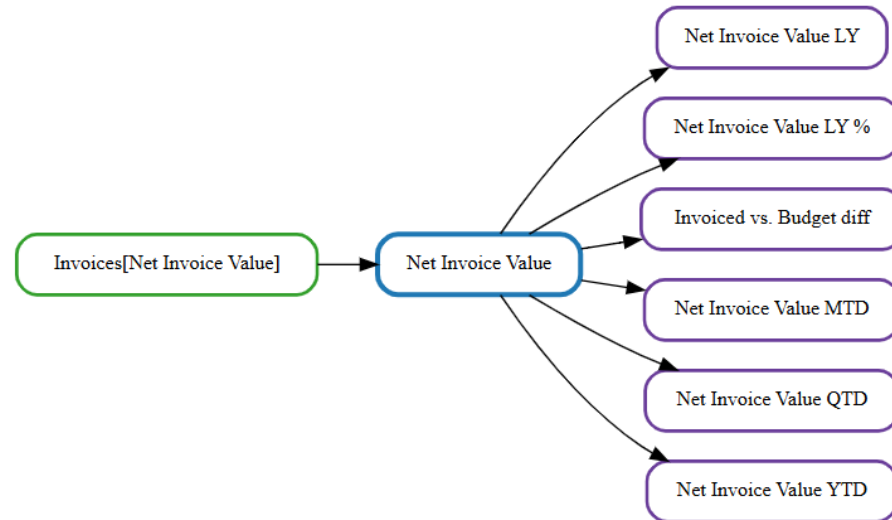
+ Code + Markdown

```
1 #Install Measure Maze in a Fabric Notebook
2 !pip install https://github.com/pawarbi/MeasureMaze/raw/main/measuremaze-0.0.1-py3-none-any.whl --q
3
```

✓

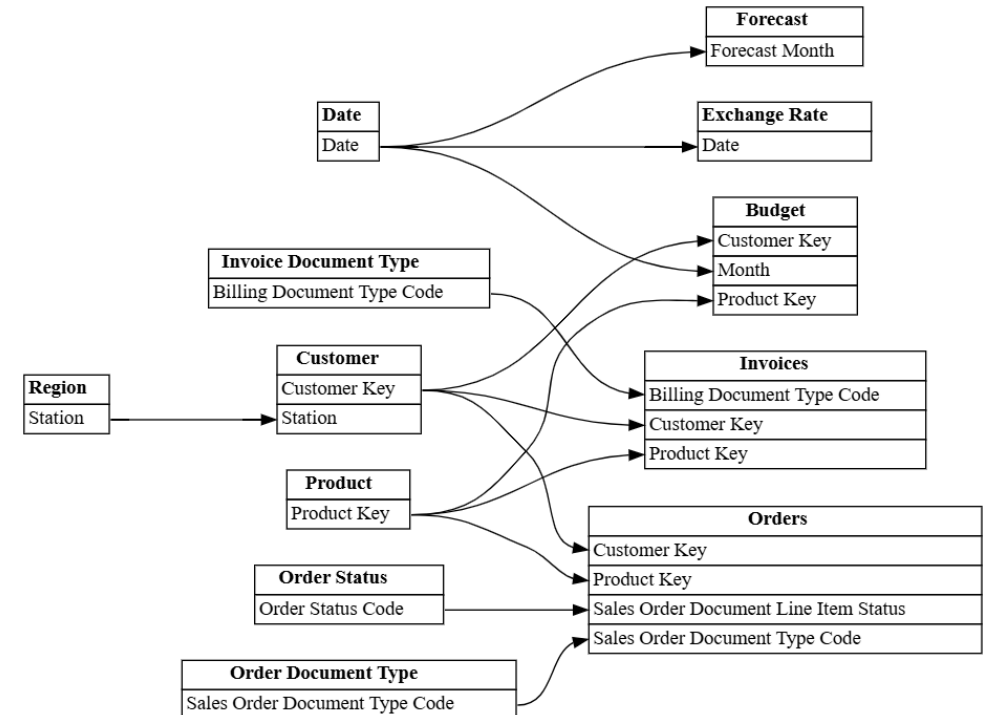
```
1 #Change the layout
2 all_objects.plot('Net Invoice Value', layout='horizontal')
```

[7] ✓ - Command executed in 375 ms by David Kofod Hanna on 1:02:35 PM, 8/12/25



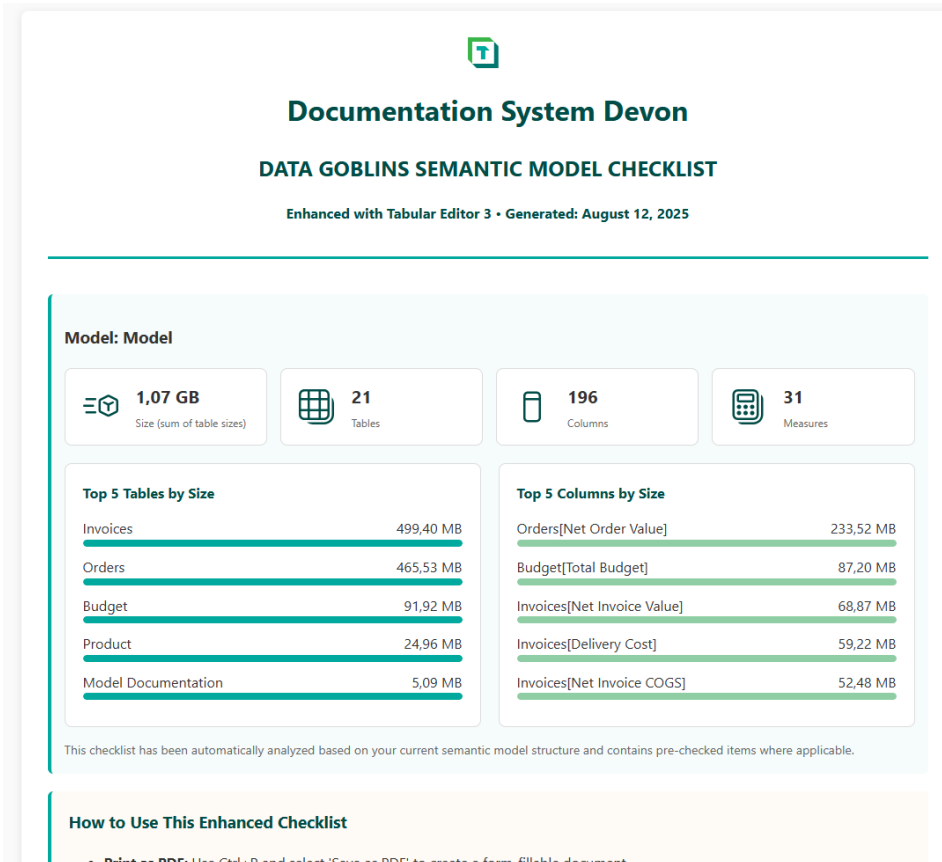
```
1 #Semantic Link can also be used to plot relationships between tables.
2
3 from sempy import fabric
4 from sempy.relationships import plot_relationship_metadata
5
6
7 relationships = fabric.list_relationships(workspace=ws, dataset=ds)
8 plot_relationship_metadata(relationships)
9
```

[10] ✓ - Command executed in 1 sec by David Kofod Hanna on 1:02:59 PM, 8/12/25





HTML Documentation script – Kurt Buhler x Tabular Editor



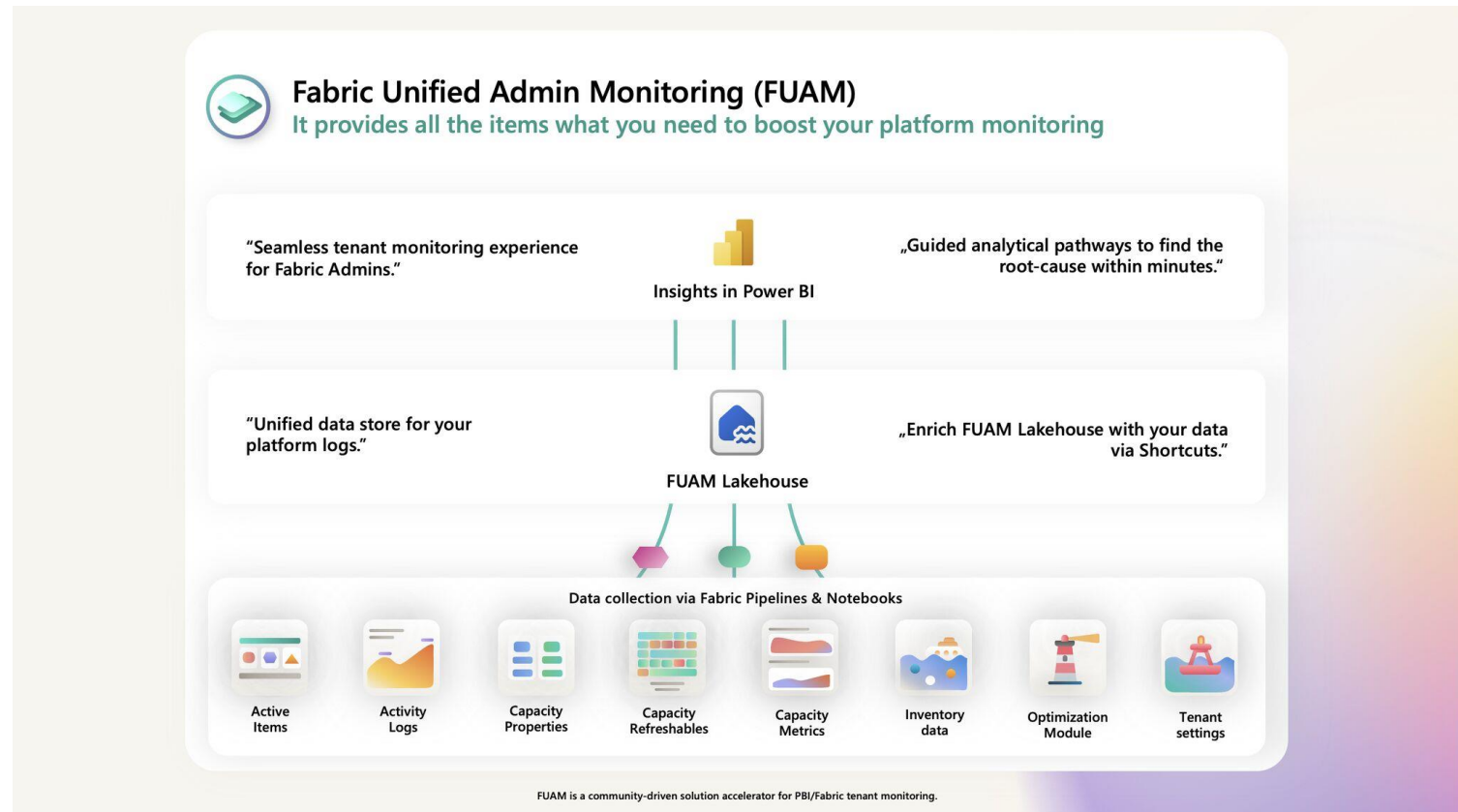
“There’s a script for that”





All into a Lakehouse + GA of PBIP and PBIR in 2026

- Semantic Link Labs
- Dataflow Gen2
- DAX Queries
- FUAM – Fabric Unified Admin Monitoring
- Copilot – Now with F2 SKU
- Copilot to be trained on Model.Bim file and .PBIR
- .PBIR meta data
- Report Best Practice Analyzer
- Best Practice Analyzer Semantic Model
- VertiPaq Analyzer
- Usage Metrics for activity logs
- Data Agents and Copilot on top of your data in Fabric





 Please write my
documentation





Power BI Documentation System – from Model to App

Automated or Script
 Manual
 Requires Fabric SKU



Power Query
{M}



Semantic Model



Power BI Desktop



App



Purpose for docs:

*Clear, Concise,
Complete, Consistent,
Correct, Consumable*

Documentation and transparency from model to app

From report consumers to report developers to semantic model developers and data ingest

Best Practice Analyzer Rules to highlight violations and severity of best practices

Tabular Editor 2/3 for Semantic Model and DAX scripts

Semantic Link Labs in Fabric Notebooks for automation like never before





Power BI Documentation System – from Model to App

 Automated or Script
 Manual
 Requires Fabric SKU



Power Query {M}

Query Folders

Applied Steps
(Rename + Properties)

Table Properties

Parameters for easy switch

Last Refresh & Measure Table & Table.Profile

Powerqueryformatter.com



Semantic Model

Format DAX

DAX Expression as Description

INFO.VIEW Model Docs

Display Folders for Measures & Columns

Table Groups

Semantic Model Docs



Power BI Desktop

Model Layout View per Fact

DAX Query View + VertiPaq Analyzer

Unpin Ineffective Default Visuals

Report Theme JSON Docs

Report Visual elements in Selection Pane

Report Docs



App

Video and PDF in an App

Notebooks in an app

OneLake Catalog

All into a Lakehouse

External Tools & Copilot

End-User Docs

Purpose for docs:

Clear, Concise, Complete, Consistent, Correct, Consumable

Documentation and transparency from model to app

From report consumers to report developers to semantic model developers and data ingest

Best Practice Analyzer Rules to highlight violations and severity of best practices

Tabular Editor 2/3 for Semantic Model and DAX scripts

Semantic Link Labs in Fabric Notebooks for automation like never before





Power BI Documentation System – from Model to App

 Automated or Script
 Manual
 Requires Fabric SKU
 GitHub Copilot with MCP server Agent Mode and .PBIP



Power Query {M}

Query Folders

Applied Steps
(Rename + Properties)

Table Properties

Parameters for easy switch

Last Refresh & Measure Table & Table.Profile

Powerqueryformatter.com



Semantic Model

Format DAX

DAX Expression as Description

INFO.VIEW Model Docs

Display Folders for Measures & Columns

Table Groups

Semantic Model Docs



Power BI Desktop

Model Layout View per Fact

DAX Query View + VertiPaq Analyzer

Unpin Ineffective Default Visuals

Report Theme JSON Docs

Report Visual elements in Selection Pane

Report Docs



App

Video and PDF in an App

Notebooks in an app

OneLake Catalog

All into a Lakehouse

External Tools & Copilot

End-User Docs

Purpose for docs:

Clear, Concise, Complete, Consistent, Correct, Consumable

Documentation and transparency from model to app

From report consumers to report developers to semantic model developers and data ingest

Best Practice Analyzer Rules to highlight violations and severity of best practices

Tabular Editor 2/3 for Semantic Model and DAX scripts

Semantic Link Labs in Fabric Notebooks for automation like never before





*“You do not rise to the level of your **goals**, you fall to the level of your **systems**.”*

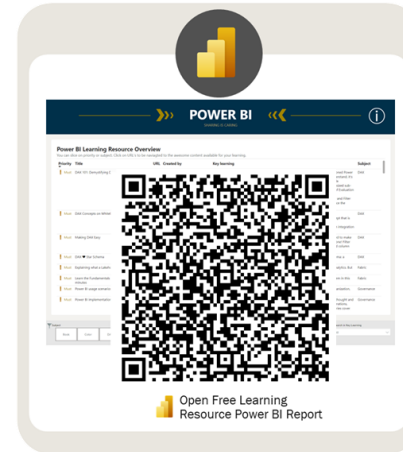
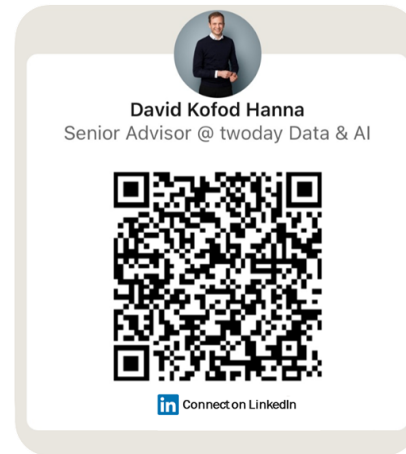
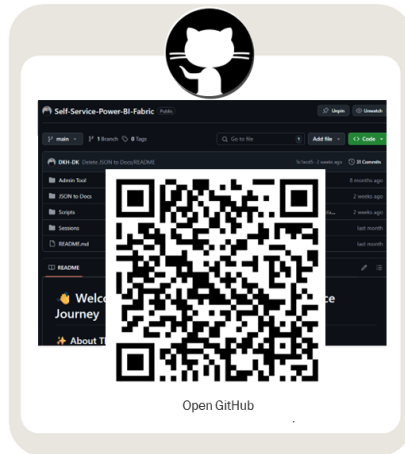
- James Clear, Atomic Habits





Thank you

- Connect to me at:



- Stay online for my live Q&A sessions

