



A Power BI Documentation System

– Not Just an Afterthought

*Devon & Cornwall User group
12th August 2025*

*Presented by
David Kofod Hanna*

twoday

David Kofod Hanna



Senior Advisor, Data Storytelling @ twoday

+200 courses as Academy Trainer and 10 years as consultant
MCT, Certified Tabular Editor 3 Trainer, DP-600, DP-700, PL-300, CPUX-F



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 Silkeborg with wife and 3 kids
Love football and running half-marathons



twoday

twoday academy

Who **loves** documentation?



Who **writes** documentation?



Who **updates** documentation?



Who **reads** documentation?



Who can **find** documentation?

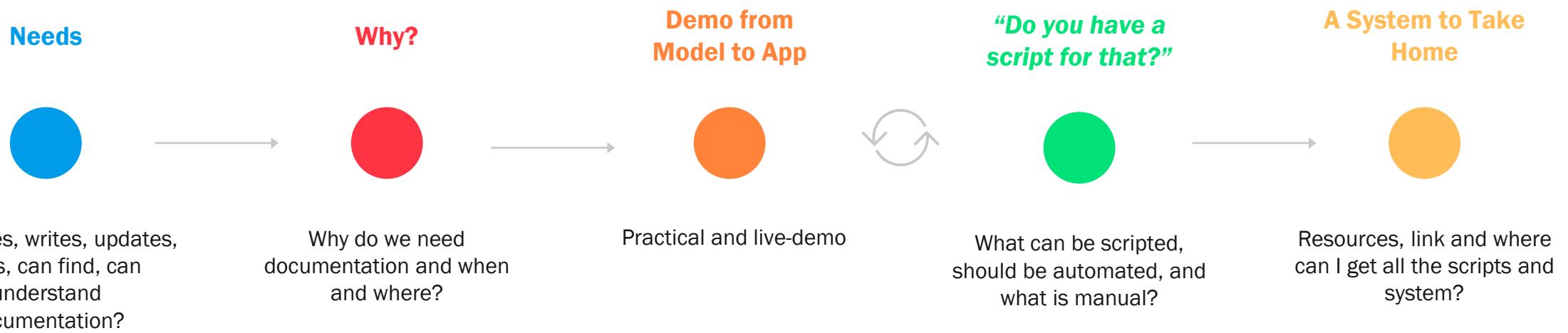


Who can **understand** documentation?



A Power BI Documentation System

– Not Just an Afterthought



Example of project

Requirements

Approve

Develop

Validate

Test

Training

Documentation

Requirements

Approve

Develop

Validate

Test

Training

Documentation

Why documentation?

Transparency

Reproducibility

Collaboration

Compliance

Scalability

Easier debugging

I CREATED 175 BOOKMARKS





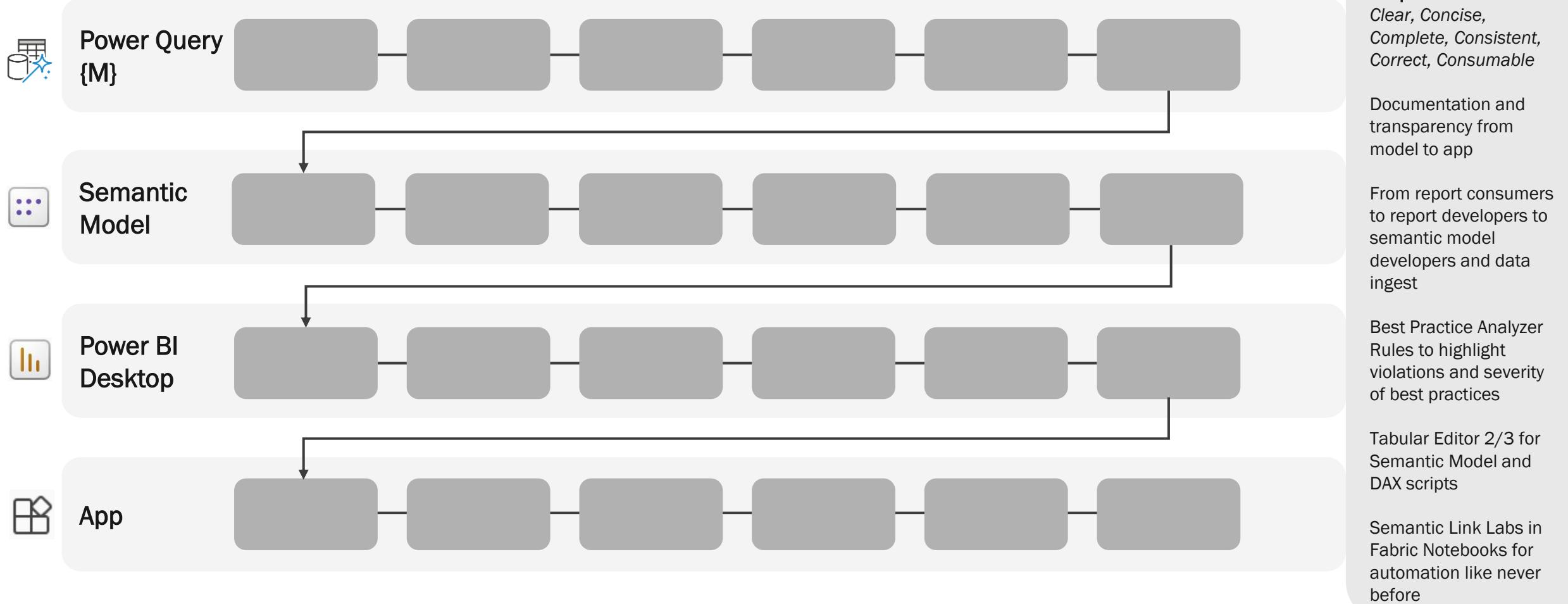
Please write my
documentation

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

- James Clear, Atomic Habits



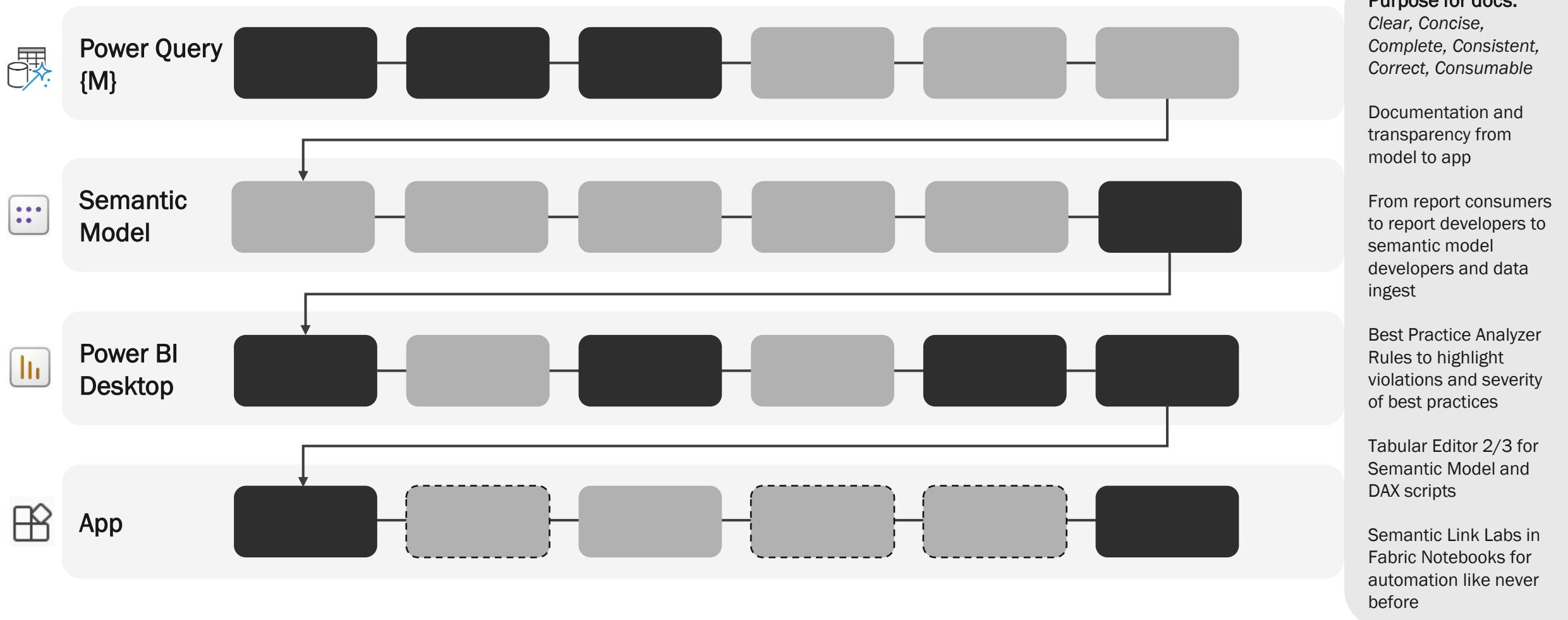
Power BI Documentation System – from Model to App





Power BI Documentation System – from Model to App

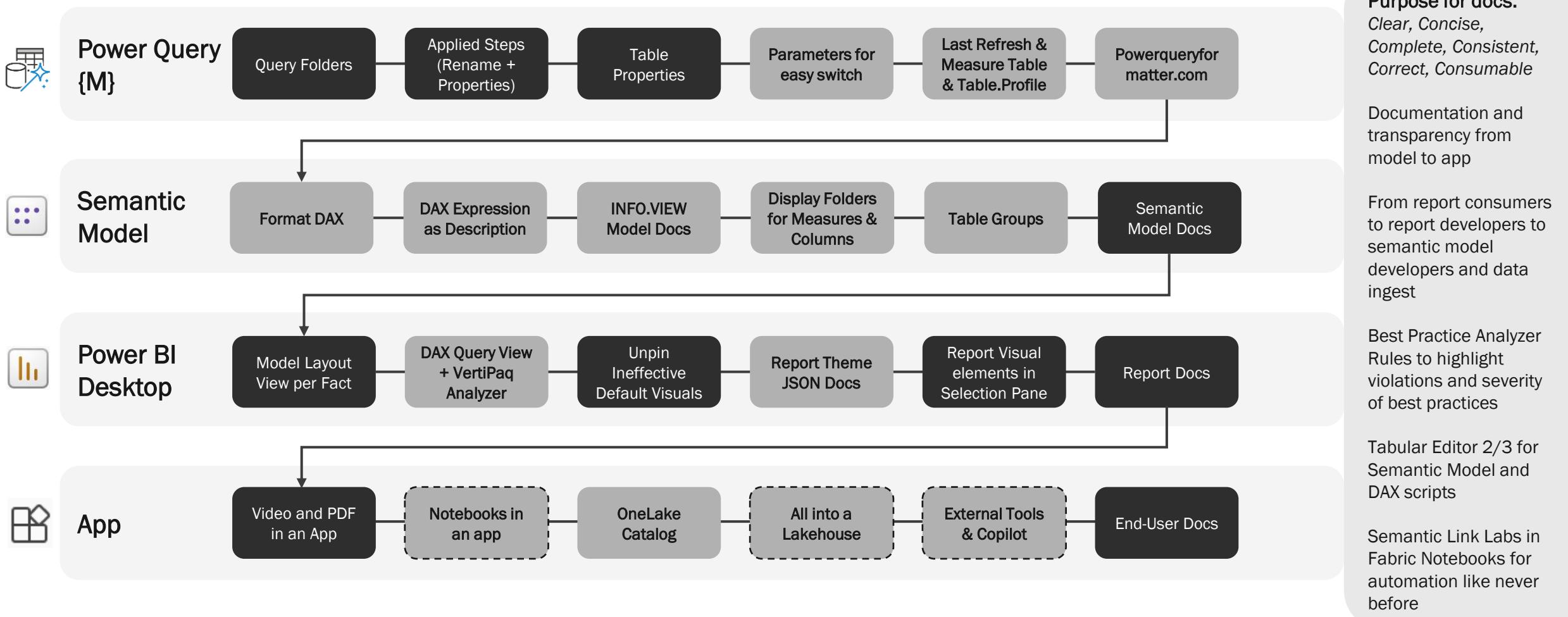
 Automated or Script Manual Requires Fabric SKU





Power BI Documentation System – from Model to App

Automated or Script Manual Requires Fabric SKU



A Power BI Documentation System

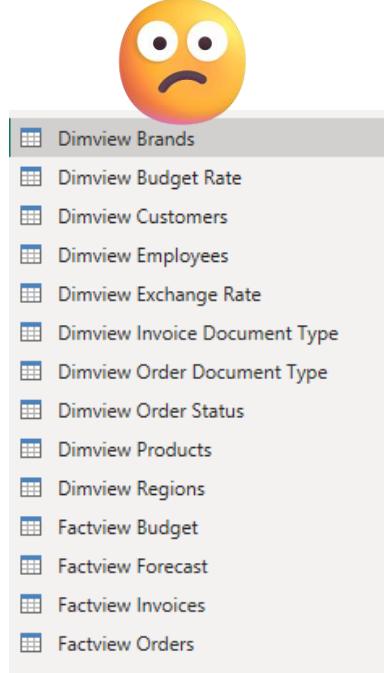
– Not Just an Afterthought

**Released on
GitHub**
and soon a
LinkedIn article

1. Create M Parameter	C# Script Source File
2. Create Global Measure Table	C# Script Source File
3. Create Last Refresh	C# Script Source File
4. Format Power Query (M)	C# Script Source File
5. Format DAX Measures	C# Script Source File
6. Measure DAX Expression as Description	C# Script Source File
7. Model Documentation DAX Script	TE3DAXS File
7. Model Documentation DAX	Text Document
8. Display Folders for Measures & Columns	C# Script Source File
9. Create Table Groups TE3	C# Script Source File
9. Table Groups in Power BI Desktop with INFO.VIEW DAX	File
9. Table Groups in Power BI Desktop with INFO.VIEW DAX Script	TE3DAXS File
10. Best Practice Analyzer Rules incl. John Kerski PQ Doc Rules	JSON Source File
11. Measure Dependency	DAX Query File
11. Model Issues	DAX Query File
11. Model Summary	DAX Query File
11. VertiPaq Column	DAX Query File
11. VertiPaq Memory Size	DAX Query File
11. VertiPaq Partition	DAX Query File
11. VertiPaq Relationship	DAX Query File
11. VertiPaq Table	DAX Query File
12. DKH Self-Service Report Theme - Raw Template	JSON Source File
12. JSON to Power BI Docs - Raw Template	Microsoft.MicrosoftPowerBI/Desktop
13. Report Analysis Notebook - Michael Kovalsky	Jupyter Source File
14. Measure Maze Dependency Sandeep Pawar	Jupyter Source File
15. Design Document - Sample Fragment 01 - General and Scope - V0.1	Microsoft Word Document
15. Design Document - Sample Fragment 02 - Workflow Issues and Business Rules - V0.2	Microsoft Word Document
15. Design Document - Sample Fragment 03 - Data - V0.3	Microsoft Word Document
15. Design Document - Sample Fragment 04 - Reports - V0.4	Microsoft Word Document
MsBIP - Power BI Documentation System - David Kofod Hanna - May 2025	Microsoft PowerPoint Presentation



Query folders by type – and Diagram View in Dataflow Gen2



twoday academy

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... <small>(i)</small>	#
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 ...

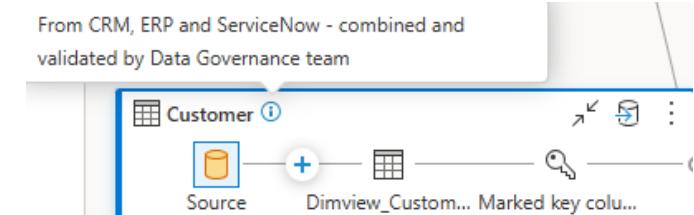
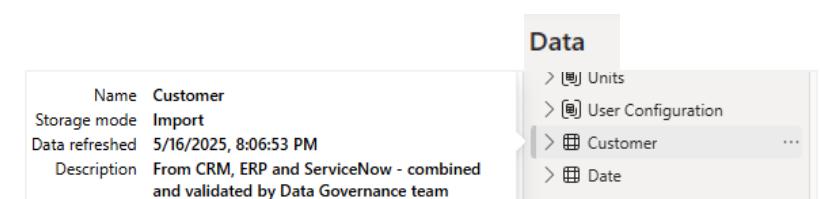
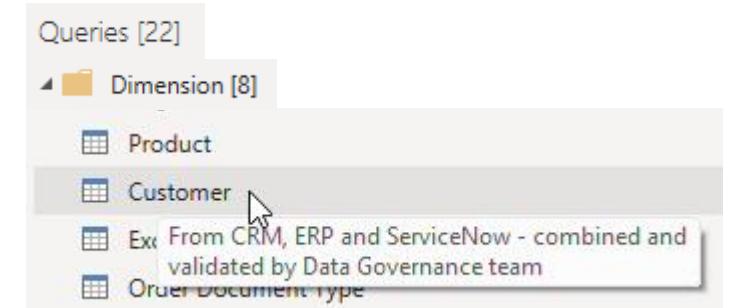
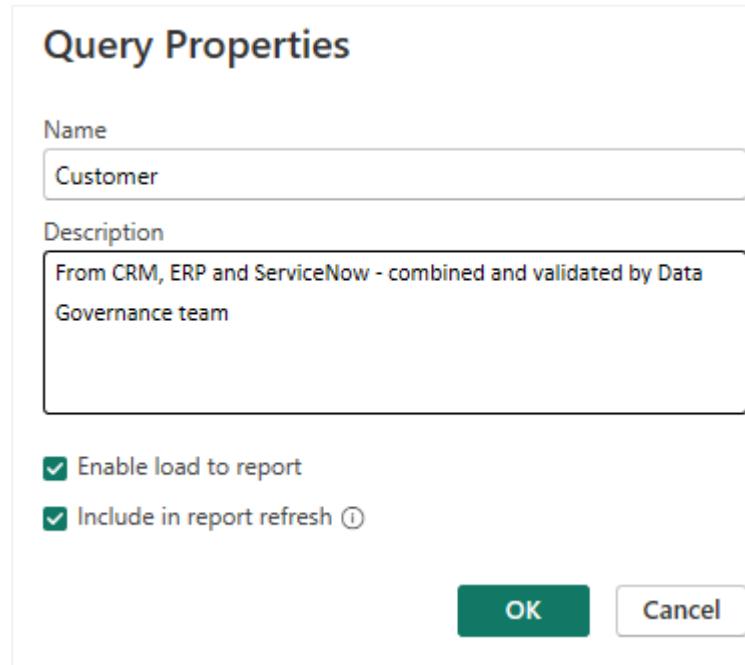
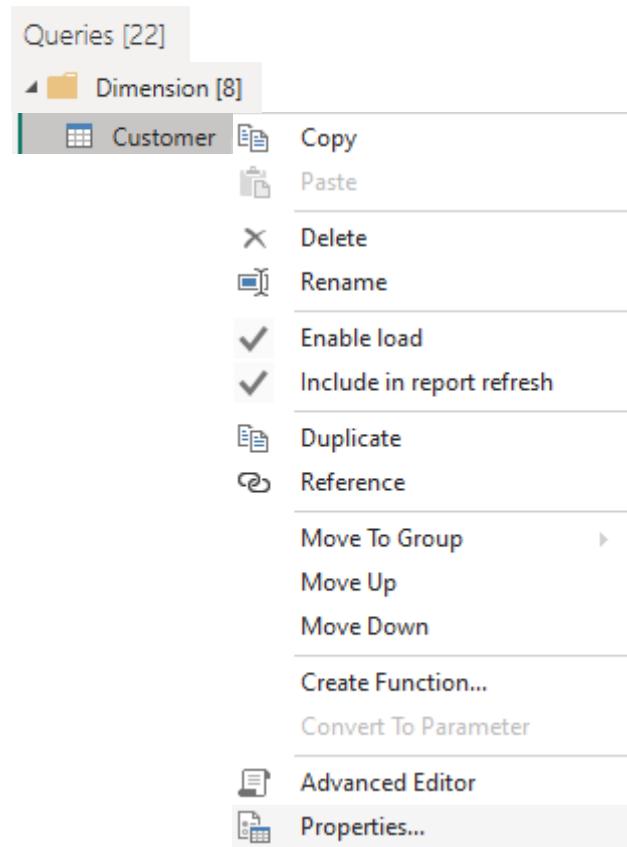
Not an excuse, but curious if the Power Query engine will optimize this in the engine call?

Like | Reply · 2 replies | 1,863 impressions

 Chandee 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





Semantic Model – OneLake catalog

OneLake catalog

Explore Govern (preview)

Power BI & Fabric Show... Data types: (All) Filter by keyword

All items My items Endorsed items Favorites Workspaces All workspaces My workspace 20 Power Query (M)... Fabric Analyst in A ... Mastering Compos... Power BI & Fabric S... More workspaces...

Name Power BI Docs - System Inspiration Power BI StagingWarehouseForDataflows_202... Supply Chain Analytics Template

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 Invoice Document Type Table MS F&O Last Refresh Table Last refresh of semantic model - not the underlying job and pi... Local Measures Table Store Local Report Level Measures

Filter by keyword

OneLake Catalog: <https://app.powerbi.com/onelake/explore?experience=power-bi>

twoday academy



DAX expression as Measure Description

Name [SVG]
Description VAR_SelectedColor = "#B2B2B2"

RETURN
"data:image/svg+xml;utf8,<svg width='36' height='36' viewBox='0 0 36 36' fill='none' xmlns='http://www.w3.org/2000/svg'><rect width='36' height='36' rx='4' fill=''" & _SelectedColor & "'/></svg>"

The screenshot shows a DAX measure named "[SVG]" with a description "VAR_SelectedColor = "#B2B2B2)". The RETURN part contains an SVG string that changes the fill color of a rectangle based on the value of the variable _SelectedColor.

- > ii. HEX Colors
- > iii. Dynamic Titles
- ✓ iv. SVG's
 - SVG
 - ✓ v. Cascading Slicers (Alberto)
 - Alberto Effect
 - ✓ vi. Active Filters Footer
 - Active Filters Footer

"There's a script for that"

```
1 foreach (var m in Model.AllMeasures)
2 { m.Description = m.Expression; }
```

Properties

General

Name: SVG

Home table: Global Measures

Description:
VAR_SelectedColor = "#B2B2B2"
RETURN

Create with Copilot (preview)

The screenshot shows a Power BI Copilot interface with a title "There's a script for that". It displays a script snippet to iterate through all measures and set their description to their expression. Below the script are properties for a measure named "SVG", including its name, home table (Global Measures), and a description field containing the DAX code shown earlier.

<https://docs.tabulareditor.com/te2/Useful-script-snippets.html>



Model Docs DAX INFO.VIEW

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()

Model Documentation

Attributes

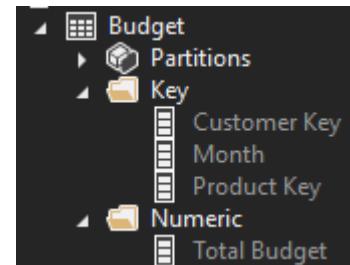
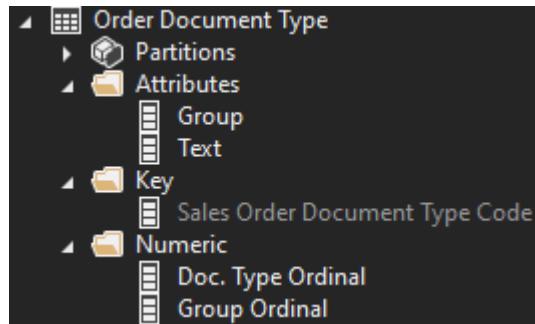
```
1 Model Documentation =
2 VAR _columns =
3   SELECTCOLUMNS(
4     FILTER(
5       INFO.VIEW.COLUMNS( ),
6       [Table] <> "Model Documentation" && NOT ( [IsHidden] )
7     ),
8       "Type", "Column",
9       "Name", [Name],
10      "Description", [Description],
11      "Location", [Table],
12      "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 )
```



"There's a script for that"



Display folders for measures and columns



"There's a script for that"

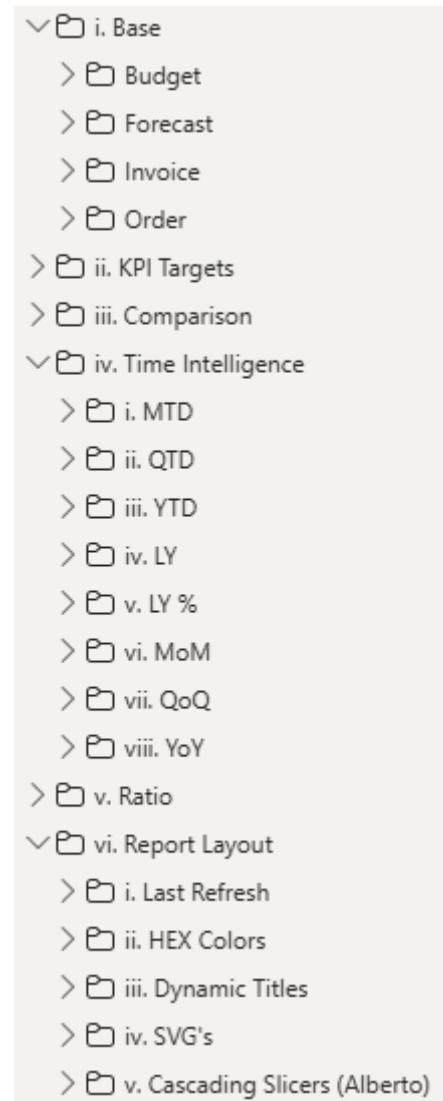
```
//Go through each table in the model
foreach(var table in Model.Tables)
{
    if(table.Name != "Date")
    {
        //First look at columns
        foreach(var column in table.Columns)
        {
            var keySuffix = "Key";
            var columnDataType = column.DataType.ToString();
            //DwCreatedDate column should be hidden in a separate 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.SummaryBy = 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.SummaryBy = 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())
                {
                    if(column.DisplayFolder == "Key")
                    {
                        column.isHidden = true;
                    }
                    else{
                        column.DisplayFolder = "Dates";
                    }
                }
            }
        }
    }
}
```



Display folders in Global and Local Measure Groups

> Global Measures
Info
Semantic Model Developer's DAX Measures

> Local Measures
Info
Local Report Level Measures



To create subfolder use \

Properties
Display folder
iv. Time Intelligence\iii. YTD

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



Table Groups in Tabular Editor 3



Semantic model

- > Calculation groups (1)
- > Cultures (1)
- > Measures (58)
- Perspectives (0)
- > Relationships (11)
- Roles (0)
- > Tables (16)
 - > Ex.00
 - > Exercises
 - > Time Intelligence
 - > Date
 - > Dimension Field Parameter
 - > Fact Population
 - > Fact Sales
 - > Fact Sales Budget
 - > Fact Sales MTD example
 - > Model Documentation
 - > Numeric Parameter
 - > Product



"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";
    }
}
```



- > Tables
 - > Calculation Groups
 - > Units
 - > User Configuration
 - > Dimensions
 - > Customer
 - > Date
 - > Invoice Document Type
 - > Order Document Type
 - > Order Status
 - > Product
 - > Region
 - > Facts
 - > Budget
 - > Exchange Rate
 - > Forecast
 - > Invoices
 - > Orders
 - > Measure Groups
 - > Global Measures
 - > Local Measures
 - > Table Group DAX
 - > Misc
 - > Last Refresh
 - > Table Profile
 - > Table Schema
 - > Parameter Tables

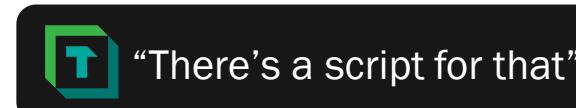


Table Group in Power BI Desktop with INFO.VIEW

Semantic model

- > Calculation groups (1)
- > Cultures (1)
- > Measures (58)
- Perspectives (0)
- > Relationships (11)
- Roles (0)
- > Tables (16) 

 - > Ex.00
 - > Exercises
 - > Time Intelligence
 - > Date
 - > Dimension Field Parameter
 - > Fact Population
 - > Fact Sales
 - > Fact Sales Budget
 - > Fact Sales MTD example
 - > Model Documentation
 - > Numeric Parameter
 - > Product
 - > Region Country
 - > Retailer



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

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 *-->
- 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")



Table Group with TMDL Annotations & DAX Query View

Annotations in TMDL View (created by Tabular Editor or do it yourself)

```
391 partition Budget = m
392     mode: import
393     queryGroup: Fact
394     source =
395         let
396             Source = Sql.Database(Server, Database),
397             Factview_Budget = Source{[Schema="Factview",Item="Budget"]}[Data],
398             #"Removed Other Columns" = Table.SelectColumns(Factview_Budget,{"Month", "Total Budget", "Customer Key", "Product Key"}),
399             #"Calculated End of Month" = Table.TransformColumns(#"Removed Other Columns",{{"Month", Date.EndOfMonth, type date}})
400         in
401             #"Calculated End of Month"
402
403 annotation PBI_NavigationStepName = Navigation
404
405 annotation PBI_ResultType = Table
406
407 annotation TabularEditor_TableGroup = Facts
408
```

DAX Query to filter on Table Group Annotations and merge with INFO.TABLES()

```
1 // Author - David Kofod Hanna
2 // Create DAX Query Table on Table Groups Feature in Tabular Editor from TMDL Annotations
3 // Contact - https://www.linkedin.com/in/davidkofod/
4
5 EVALUATE
6
7 // Extract all table-level metadata: ID, Name, and Description
8 VAR _tables =
9     SELECTCOLUMNS(
10        INFO.TABLES(),
11        "TableID", [ID],
12        "TableName", [Name],
13        "Description", [Description]
14    )
15
16 // Extract annotations where the name matches "TabularEditor_TableGroup"
17 // These annotations are assigned to tables via Tabular Editor scripts https://docs.tabulareditor.com/te3/features/table-groups.html#metadata-and-scripting
18 // Alternatively, you can manually assign an annotation in TMDL view for each table/partition
19 VAR _annotations =
20     SELECTCOLUMNS(
21        FILTER(
22            INFO.ANNOTATIONS(),
23            [Name] = "TabularEditor_TableGroup"
24        ),
25        "TableID", [ObjectID],
26        "Annotation Name", [Name],
27        "Annotation Value", [Value]
28    )
29
30 // Join tables and annotations based on matching Table IDs
31 RETURN
32     NATURALINNERJOIN(_tables, _annotations)
33     ORDER BY [Annotation Value]
34
```



“There’s a script for that”



DAX Query View for testing



DAX Query View

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 - Harisharan Rajendran
2 // Name - Model_Summary_in_DAX_Query_View
3 // Version - 1.0
4 // Contact - https://www.linkedin.com/in/lehar-harsh/
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 // Details of the properties
7 EVALUATE
8 VAR _tableCount = COUNTROWS(INFO.TABLES())
9 VAR _columnCount = COUNTROWS(INFO.COLUMNS())
10 VAR _calculatedColumnCount = 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(FILTER(
18         VISIBLE(TABLES()),
19         [StorageMode]
20     )) > 3,
21     "Yes",
22     "No"
23 )
24 VAR _daxTables = COUNTROWS(FILTER(

```

Model Summary

[Property]	[Value]	[Comment]
Tables	21	EVALUATE INFO.TABLES()
Columns	197	EVALUATE INFO.COLUMNS()
--CalculatedColumns	0	EVALUATE FILTER(INFO.CALCULATEDCOLUMNS())
--DirectColumns	197	EVALUATE FILTER(INFO.DIRECTCOLUMNS())
Measures	17	EVALUATE INFO.MEASURES()
Relationships	13	EVALUATE INFO.RELATIONSHIPS()
IsCompositeModel	No	EVALUATE INFO.VIEWTYPE()
DAXTables	2	EVALUATE FILTER(INFO.DAXTABLES())
IsPartitioned	No	EVALUATE FILTER(INFO.ISPARTITIONED())
Perspectives	0	EVALUATE FILTER(INFO.PERSPECTIVES())
Calculation Groups	2	EVALUATE FILTER(INFO.CALCULATIONGROUPS())
Roles	0	EVALUATE FILTER(INFO.ROLES())
PBIDesktopVersion	2.142.1277.0 (25.04)+014fde45...	EVALUATE FILTER(INFO.PBIDESKTOPVERSION())

Model Issues

[Property]	[Value]	[Comment]	[Status]
1 Is Partition Required?	Yes	EVALUATE VAR_dt_part...	Fail
2 Dedicated Date Table	No	EVALUATE FILTER(INFO.D...	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 Inv...	TABLE	Invoices	Invoices
2 Net Invoice Value YTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Inv...	COLUMN	Invoices	Net Invoice Value
3 Net Invoice Value MTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Inv...	TABLE	Invoices	Invoices
4 Net Invoice Value MTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Inv...	COLUMN	Invoices	Net Invoice Value
5 Net Invoice Value QTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Inv...	TABLE	Invoices	Invoices
6 Net Invoice Value QTD	CALCULATE([Sum of Ne...	Sum of Net Invoice Value	SUM('Invoices'[Net Inv...	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 .PBIR and for easy copy/paste

System.SemanticModel > DAXQueries >

.pbi	File folder
Measure Dependency	DAX Query File
Model Issues	DAX Query File
Model Summary	DAX Query File
definition.pbism	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.

1 Storage Table Column = `INFO.STORAGECOLUMN()`

The function 'INFO.STORAGECOLUMN()' cannot be used in the expression of a measure, calculated column, or calculated table defined in the model.

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 TMSCHHEMA_ATTR"),
    ("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.

Try yourself! [Download at GitHub](#)

Active Report Theme
DKH Self-Service Report Theme

Theme Colors 1-8
For data visualization

Theme	HEX	ID
Blue	#367cff	1
Orange	#fa8100	2
Cyan	#6cc6cb	3
Purple	#aa77dd	4
Red	#d14576	5
Brown	#b2bd6d	6
Dark Blue	#8b9fd4	7
Yellow	#eae5c9	8

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

Learn more

KPI Colors
For KPI performance status context

KPI	HEX	Property
Green	#37a78f	Good
Yellow	#2f2f2f	Neutral
Red	#a74d37	Bad

Divergent Min & Max Colors
For divergent heat map

Divergent	HEX	Property
Orange	#fa8100	Minimum
Yellow	#2f2f2f	Center
Blue	#367cff	Maximum
Grey	#367cff	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
Black	#485257	Background
Light Grey	#F1F3F4	BackgroundLight
Neutral	#485257	BackgroundNeutral
Foreground	#485257	Foreground
Neutral Secondary	#485257	ForegroundNeutralSecondary
Neutral Tertiary	#606E74	ForegroundNeutralTertiary
Hyperlink	#ebeef6	TableAccent
Visited Hyperlink	#ebeef6	TableAccent

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

Learn more

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

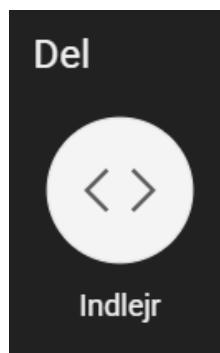
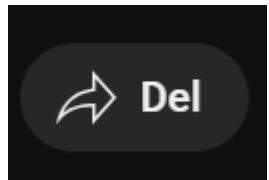
- Card New Visual
- Matrix
- Line chart
- Column chart
- Bar chart
- Slicer
- Action button
- Page Navigator

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
bookmarkNa	vigator	sid	fill	selected
		shape	default	
		text	selected	
		border	color	#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
			outline	FALSE
		tileShape	shape	rectangle

Embed videos



```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/pQHX-SjgQvQ?si=RGiIqX5jFU2LoFs0"
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">
```

Helpdesk support back in the day of the middle age

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-SjgQvQ?si=RGiIqX5jFU2LoFs0>

Include https:// at the beginning of the URL

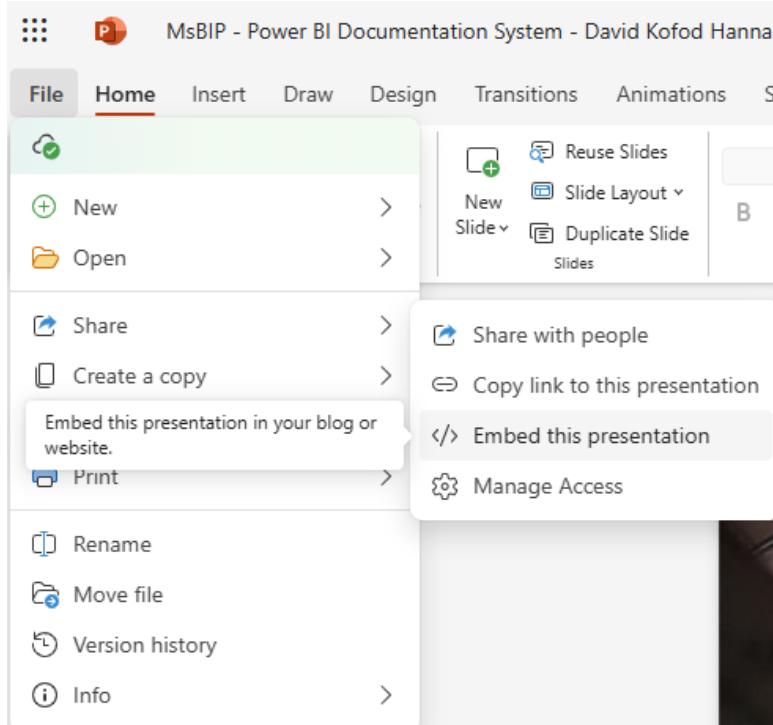
Link behavior

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

Create

Cancel

Embed PowerPoint, Excel, PDF



Embed

Preview

The preview window shows a slide from a presentation titled 'A Power BI Documentation System - Not Just an Afterthought'. The slide features a blue circular graphic in the center with the title text. In the background, there is a photograph of two people walking on an escalator. The preview includes navigation controls at the bottom.

Dimensions

Width (px): 476
Height (px): 288

Interaction

Use default autoadvance settings from the file

Embed Code

```
<iframe src="https://twodaygroup-my.sharepoint.com/personal/david_hanna_twoday_com/_layouts/15/Doc.aspx?sourcedoc={d687580d-e075-431a-bf09-...>
```

This file will only be accessible to people with permissions.

Buttons

Copy **Close**

Guy in a Cube: <https://www.youtube.com/watch?v=fr1yjm-uFRE>

twoday academy

Design Documents – Samples

A screenshot of a GitHub repository named "alexbadiu-insightsinmotion / PBI-Documentation". The repository contains several files listed on the left:

- 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

A screenshot of a "Power BI Documentation – Design Document Sample Fragment 04 – Reports" page. The page features a "Table of Contents" section with the following 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	<p>DESIGN:</p> <ul style="list-style-type: none">Type=buttonShape=any, with border=offAction=page navigation (subcategory 1) <p>DEFAULT (selected):</p> <ul style="list-style-type: none">Font=Segoe UI, white, 10 ptBackground=dark blueNavigation=page, Invoices-All <p>DEFAULT (unselected):</p> <ul style="list-style-type: none">Font=Segoe UI, medium grey, 10 ptBackground=medium blueNavigation=page, Invoices-All <p>HOVER:</p> <ul style="list-style-type: none">Font=Segoe UI, dark grey, 11 ptBackground=medium greyNavigation=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]



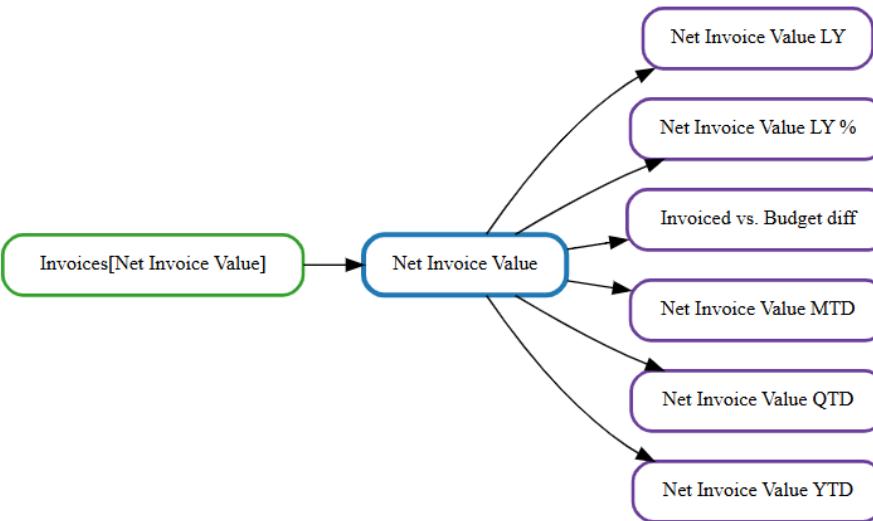
Embed Notebooks for technical documentation

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

+ Code + Markdown

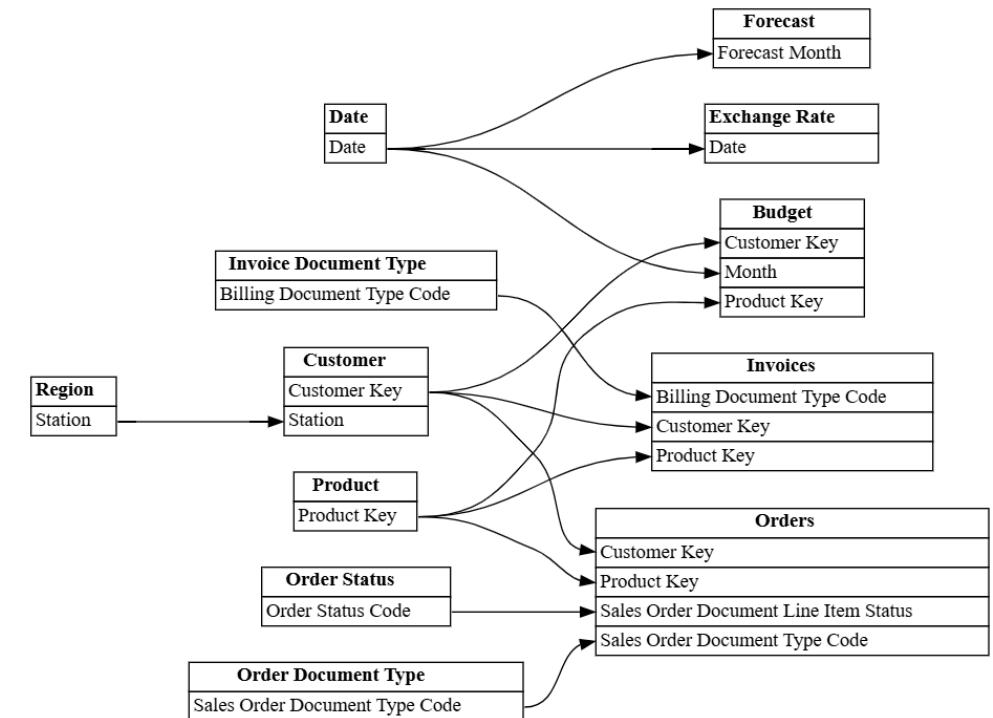
```
1 #Install Measure Maze in a Fabric Notebook
2 !pip install https://github.com/pawarbi/MeasureMaze/raw/main/measuremaze-0.0.1-py3-none-any.whl --q
3
```

[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

Documentation System Devon

DATA GOBLINS SEMANTIC MODEL CHECKLIST

Enhanced with Tabular Editor 3 • Generated: August 12, 2025

Model: Model

1,07 GB
Size (sum of table sizes)

21
Tables

196
Columns

31
Measures

Top 5 Tables by Size

Invoices	499,40 MB
Orders	465,53 MB
Budget	91,92 MB
Product	24,96 MB
Model Documentation	5,09 MB

Top 5 Columns by Size

Orders[Net Order Value]	233,52 MB
Budget[Total Budget]	87,20 MB
Invoices[Net Invoice Value]	68,87 MB
Invoices[Delivery Cost]	59,22 MB
Invoices[Net Invoice COGS]	52,48 MB

This checklist has been automatically analyzed based on your current semantic model structure and contains pre-checked items where applicable.

How to Use This Enhanced Checklist

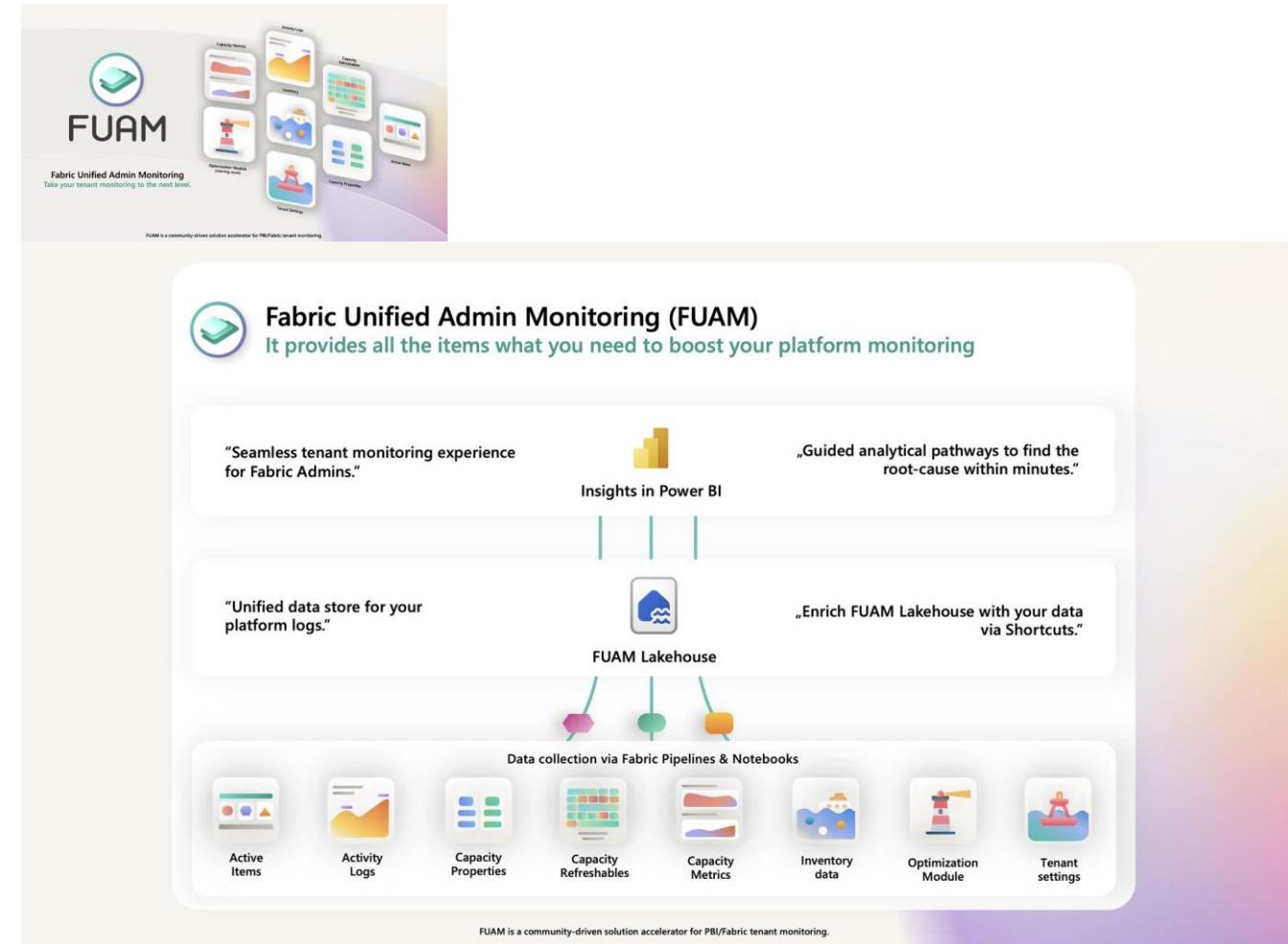
- **Print as PDF:** Use Ctrl+P and select 'Save as PDF' to create a form-fillable document

“There’s a script for that”



All into a Lakehouse

- 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
- .PBIR meta data into
- 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



FUAM & Fabric Toolbox: <https://github.com/microsoft/fabric-toolbox>



Report Visual Elements

The screenshot shows the 'Report Visual Elements' pane from Power BI. At the top is a large orange sad face icon. Below it is a 'Selection' dropdown and a '... >>' button. Underneath are two tabs: 'Layer order' (which is selected) and 'Tab order'. A 'Show' and 'Hide' button follows. The main list contains the following items:

- Slicer
- Slicer
- Slicer
- Button
- Button
- Button
- Button
- Data last refreshed:





Report Visual Elements – especially important if using bookmarks

The screenshot shows the Power BI interface with the 'Selection' pane open on the left. The 'Bookmarks' pane is open, displaying a list of bookmarks for 'Page 4: Location'. The 'Slicer Pane opened' bookmark is selected. Below the panes, a report page titled 'Power BI User Adoption' is visible, featuring two bar charts: 'User Engagement Rate (UER)' and 'User Engagement Rate (UER) in %'.

City	User Engagement Rate (UER)	User Engagement Rate (UER) in %
Paris	38%	81%
Aberystwyth	34%	66%
Aarhus	31%	69%
Copenhagen	32%	68%
Kolding	27%	72%
Vilnius	30%	70%

💡 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**”

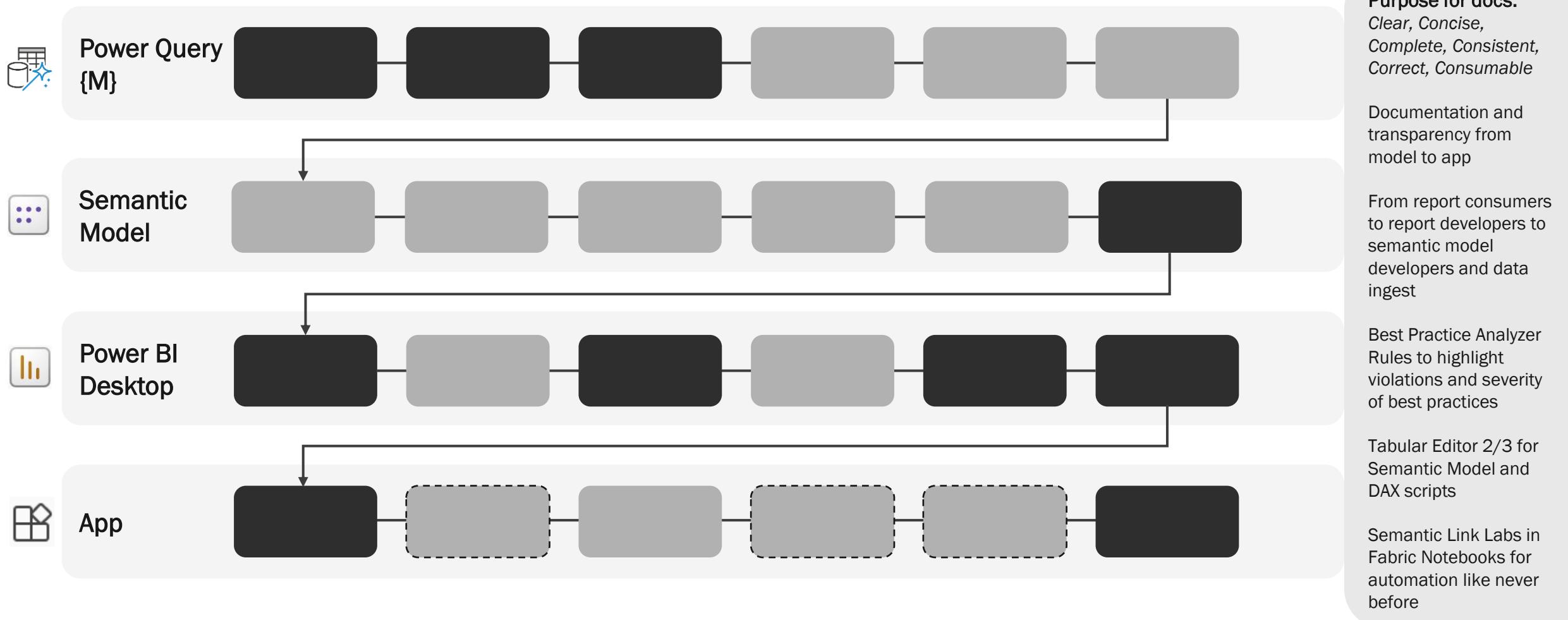


Please write my
documentation



Power BI Documentation System – from Model to App

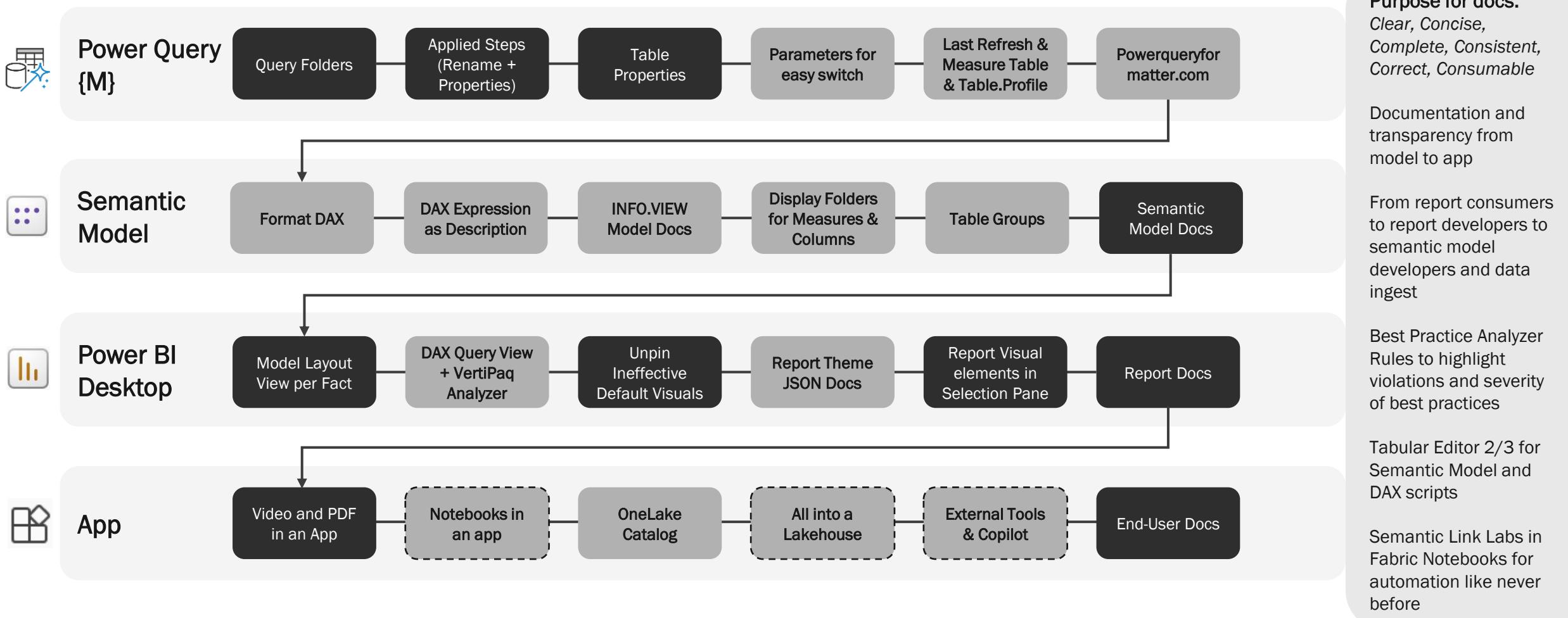
 Automated or Script Manual Requires Fabric SKU





Power BI Documentation System – from Model to App

Automated or Script Manual Requires Fabric SKU



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

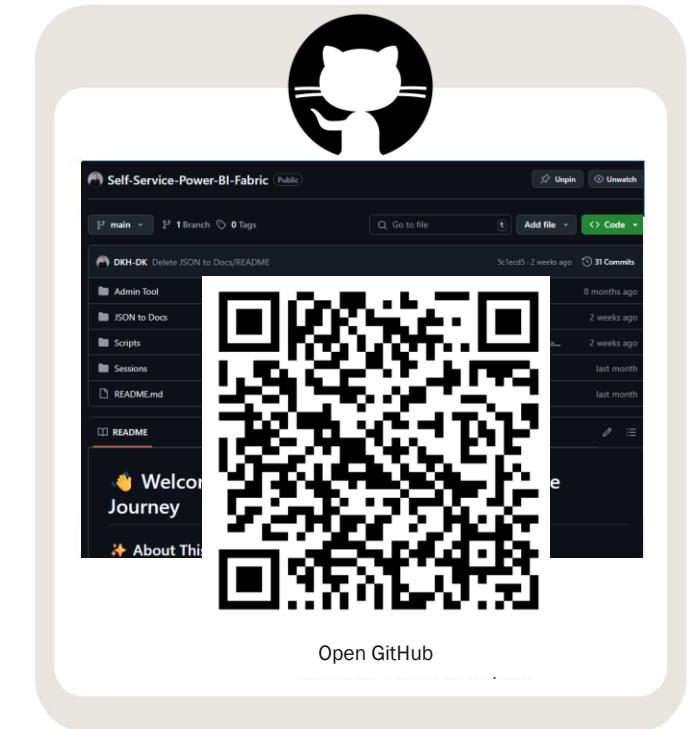
- James Clear, Atomic Habits

A Power BI Documentation System

– Not Just an Afterthought

**Released on
GitHub**
and soon a
LinkedIn article

1. Create M Parameter	C# Script Source File
2. Create Global Measure Table	C# Script Source File
3. Create Last Refresh	C# Script Source File
4. Format Power Query (M)	C# Script Source File
5. Format DAX Measures	C# Script Source File
6. Measure DAX Expression as Description	C# Script Source File
7. Model Documentation DAX Script	TE3DAXS File
7. Model Documentation DAX	Text Document
8. Display Folders for Measures & Columns	C# Script Source File
9. Create Table Groups TE3	C# Script Source File
9. Table Groups in Power BI Desktop with INFO.VIEW DAX	File
9. Table Groups in Power BI Desktop with INFO.VIEW DAX Script	TE3DAXS File
10. Best Practice Analyzer Rules incl. John Kerski PQ Doc Rules	JSON Source File
11. Measure Dependency	DAX Query File
11. Model Issues	DAX Query File
11. Model Summary	DAX Query File
11. VertiPaq Column	DAX Query File
11. VertiPaq Memory Size	DAX Query File
11. VertiPaq Partition	DAX Query File
11. VertiPaq Relationship	DAX Query File
11. VertiPaq Table	DAX Query File
12. DHK Self-Service Report Theme - Raw Template	JSON Source File
12. JSON to Power BI Docs - Raw Template	Microsoft.MicrosoftPowerBIDesktop
13. Report Analysis Notebook - Michael Kovalsky	Jupyter Source File
14. Measure Maze Dependency Sandeep Pawar	Jupyter Source File
15. Design Document - Sample Fragment 01 - General and Scope - V0.1	Microsoft Word Document
15. Design Document - Sample Fragment 02 - Workflow Issues and Business Rules - V0.2	Microsoft Word Document
15. Design Document - Sample Fragment 03 - Data - V0.3	Microsoft Word Document
15. Design Document - Sample Fragment 04 - Reports - V0.4	Microsoft Word Document
MsBIP - Power BI Documentation System - David Kofod Hanna - May 2025	Microsoft PowerPoint Presentation



twoday academy

Credit and resources

- Alex Badiu and Greg Philips GitHub on Documentation: <https://github.com/alexbadiu-insightsinmotion/PBI-Documentation/tree/main>
- Kurt Buhler Data Goblin: <https://data-goblins.com/power-bi/dataset-checklist>
- JSON to Docs Template: <https://www.linkedin.com/feed/update/urn:li:activity:7323594556698968064/>
- Table Groups in Power BI Desktop: <https://www.linkedin.com/pulse/table-groups-power-bi-desktop-infoview-david-kofod-hanna-aotof>
- Model Documentation in DAX: <https://www.linkedin.com/pulse/model-documentation-automation-dax-scripts-tabular-editor-hanna-7bhyf/>
- David's GitHub: <https://github.com/DKH-DK/Self-Service-Power-BI-Fabric/tree/main>
- One App to Rule Them All: <https://www.linkedin.com/pulse/one-app-rule-them-all-david-kofod-hanna-pvopf/>
- Best Practice Analyzer: <https://powerbi.microsoft.com/da-dk/blog/best-practice-rules-to-improve-your-models-performance/>
- Michael Kovalsky BPA Rules: <https://github.com/m-kovalsky/Tabular?tab=readme-ov-file>
- Sandeep Pawar: <https://fabric.guru/measure-maze-visualizing-measure-dependencies-using-semantic-link-network-analysis>
- Hariharan Rajendran: <https://www.linkedin.com/feed/update/urn:li:activity:7307061810375925760/>
- Jon Vöge: <https://downhill-data.com/2024/12/10/fabric-quick-tips-hiding-power-bi-report-page-navigation-in-workspace-apps-and-organizational-apps/>
- Tabular Editor Scripts: <https://docs.tabulareditor.com/common/CSharpScripts>
- FUAM & Fabric Toolbox: <https://github.com/microsoft/fabric-toolbox>
- Powerops: <https://powerops.app/>
- Power Query Formatter: www.powerqueryformatter.com
- John Kerski Power Query BPA Rules: <https://gist.github.com/kerski/30bfe29526db6aa9c109e780e7902579>
- Guy in a Cube: <https://www.youtube.com/watch?v=fr1yjm-uFRE>
- Alice Aguiar Costa: <https://www.linkedin.com/pulse/designing-self-learning-power-bi-demo-end-users-alice-costa-le4uf/>
- OneLake Catalog: <https://app.powerbi.com/onelake/explore?experience=power-bi>