



PUBLIC

2026-09-23

Analyze and Plan with Formulas

Content

1	Using formulas.	3
2	Create a formula in Excel Web.	5
3	Create a formula in Excel Desktop.	7
4	Add a data source for formulas.	10
5	SAP.ASYMMETRICFILTER.	15
6	SAP.BLOCK.	17
7	SAP.GETDATA.	18
8	SAP.GETDATACOMMENT.	20
9	SAP.GETDIMENSIONFILTER.	22
10	SAP.GETDIMENSIONS.	24
11	SAP.GETPLANNINGTRIGGERNAME.	26
12	SAP.GETPLANNINGTRIGGERPARAMETERS.	27
13	SAP.GETPLANNINGTRIGGERPARAMETERVALUE.	28
14	SAP.GETTABLENAME.	29
15	SAP.GETTABLESTATUS.	30
16	SAP.GETVARIABLES.	32
17	SAP.MEMBERPROPERTY.	34
18	SAP.OVERWRITEDIMENSIONFILTER.	36
19	SAP.SETDATA.	38
20	SAP.TIMEOFFSET.	43
21	Keyboard shortcuts for the add-in.	45

1 Using formulas

The SAP Analytics Cloud add-in contains its own set of functions that you can use to create formulas. You can use these functions to include data and metadata of the used data sources in your workbooks. For example, you can insert the table name with the function `SAP.GETTABLENAME` in your workbook, and then use the table name as an argument in other functions, such as `SAP.GETDIMENSIONFILTER`.

A Microsoft Excel formula for the add-in consists of a function and arguments (references to the data source and dimensions). In the formulas, a list separator is used to specify the arguments.

The examples in this guide use a comma as the list separator. If you want to use these examples on your machine, they'll only work properly if you have set the same default list separator.

If you use a different default separator than the examples, you can change the list separator in the examples, or you can change the default list separator for your machine. Please keep in mind that changing the default list separator on your machine may influence other lists you use.

Note

Some formulas (for example, `SAP.GETDATA` and `SAP.SETDATA`) require the member ID in their definition. If a member ID is not unique in a dimension, the formula will randomly use one of the members with the specified ID, but it could be a different member than the expected one.

With the add-in, you can also use the standard functions of Microsoft Excel to build formulas.

→ Recommendation

Use of Volatile Functions

As mentioned in the Microsoft Excel documentation, the use of too many volatile functions can lower the recalculation performance.

As a reminder, the following Microsoft Excel functions are volatile: `NOW`, `TODAY`, `RAND`, `OFFSET`, `INDIRECT`, `INFO` (depending on its arguments), `CELL` (depending on its arguments).

For more information, you can see the Excel help at [Excel performance: Improving calculation performance](#) .

SAP Analytics Cloud functions

The following functions are available in the category SAP Analytics Cloud:

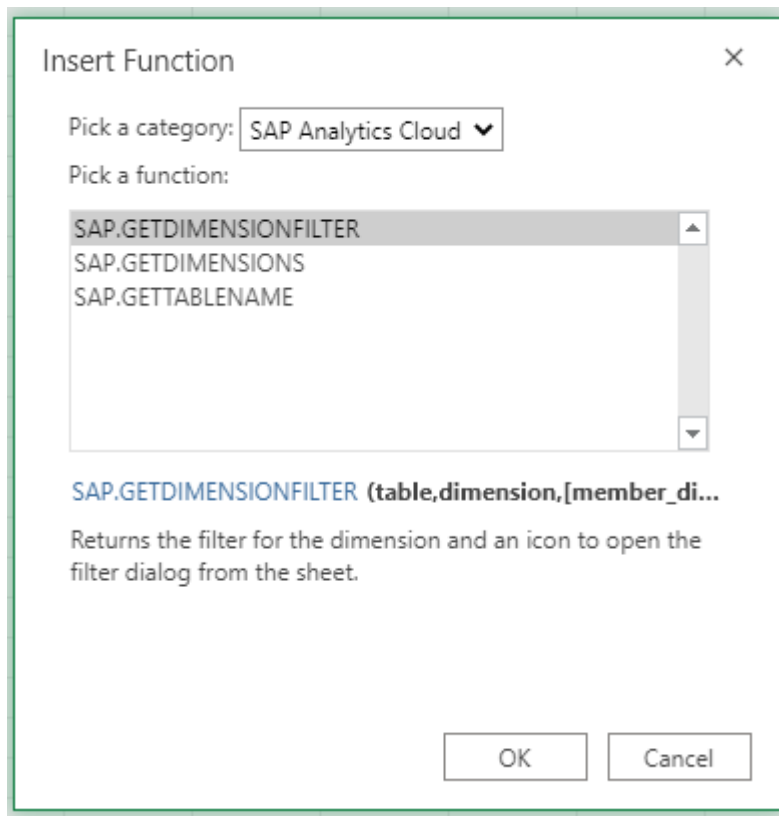
- [SAP.ASYMMETRICFILTER](#) [page 15]
- [SAP.BLOCK](#) [page 17]
- [SAP.GETDATA](#) [page 18]
- [SAP.GETDIMENSIONFILTER](#) [page 22]

- [SAP.GETDIMENSIONS \[page 24\]](#)
- [SAP.GETPLANNINGTRIGGERNAME \[page 26\]](#)
- [SAP.GETPLANNINGTRIGGERPARAMETERS \[page 27\]](#)
- [SAP.GETPLANNINGTRIGGERPARAMETERVALUE \[page 28\]](#)
- [SAP.GETTABLENAME \[page 29\]](#)
- [SAP.GETTABLESTATUS \[page 30\]](#)
- [SAP.GETVARIABLES \[page 32\]](#)
- [SAP.MEMBERPROPERTY \[page 34\]](#)
- [SAP.OVERWRITEDIMENSIONFILTER \[page 36\]](#)
- [SAP.SETDATA \[page 38\]](#)
- [SAP.TIMEOFFSET \[page 43\]](#)

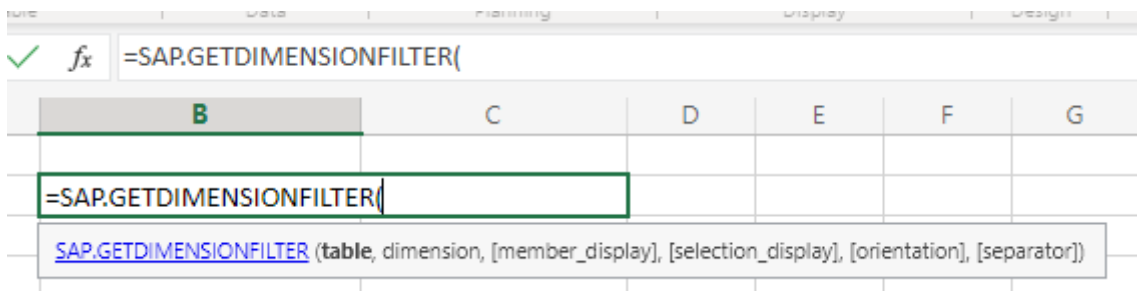
2 Create a formula in Excel Web

To create a formula with SAP Analytics Cloud functions in Excel for the Web:

1. Select the cell in which you want to enter the formula, cell B2 for example.
2. To start the formula with a function, click the [Insert Function](#) button on the formula bar: 
3. Choose [SAP Analytics Cloud](#) in the [Pick a category](#) box.



4. Choose a function in the [Pick a function](#) box.
In this example, SAP.GETDIMENSIONFILTER is selected.
5. Choose [OK](#).
The function is inserted in the cell.



6. Enter the arguments for the function.

fx =SAP.GETDIMENSIONFILTER("Table2","Product",3,1,0)

7. Choose *Enter*.

The formula is added to the cell. You can see the formula with its function and arguments in the formula bar when you select the cell in the sheet.

fx	=SAP.GETDIMENSIONFILTER("Table2","Product",3,1,0)		
	B	C	D
	 Product		
	Carbonated Drinks - PC1	Self & Descendants	
	Juices - PC2	Self & Descendants	
	Coconut Water - PD19	Self Only	

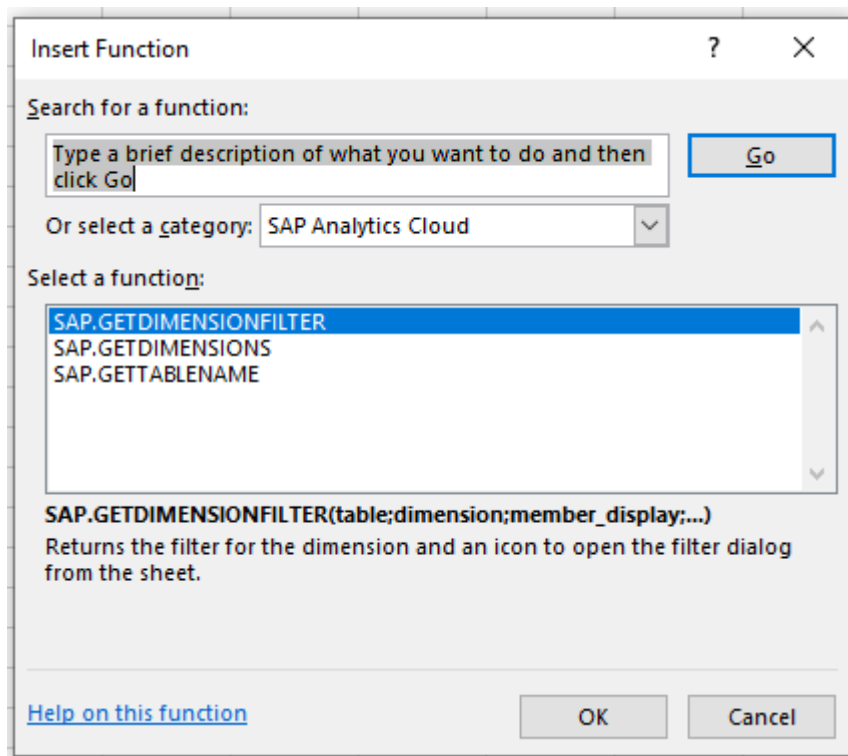
Related Information

[SAP.GETDIMENSIONFILTER \[page 22\]](#)

3 Create a formula in Excel Desktop

To create a formula with SAP Analytics Cloud functions in Excel Desktop:

1. Select the cell in which you want to enter the formula, cell B2 for example.
2. To start the formula with a function, click the *Insert Function* button on the formula bar: 
3. Choose *SAP Analytics Cloud* in the *Or select a category* box.



4. Choose a function in the *Pick a function* box.
In this example, SAP.GETDIMENSIONFILTER is selected.
5. Choose *OK*.
The *Function Argument* dialog box appears.

Function Arguments

SAP.GETDIMENSIONFILTER

Table		↑	=
Dimension		↑	=
Member_display		↑	=
Selection_display		↑	=
Orientation		↑	=

Returns the filter for the dimension and an icon to open the filter dialog from the sheet.

Table is the table name.

Formula result =

[Help on this function](#) OK Cancel

6. Enter the arguments for the function in the dialog box.
At the bottom of the box, you find some information for the current argument, like Member_display in this example.

Function Arguments

SAP.GETDIMENSIONFILTER

Table	Table2	↑	= Table2
Dimension	Product	↑	= Product
Member_display	3	↑	= 3
Selection_display	1	↑	= 1
Orientation	0	↑	= 0

Returns the filter for the dimension and an icon to open the filter dialog from the sheet.

Member_display is 0 for ID, 1 for Description, 2 for ID and Description (default), 3 for Description and ID, or 4 for full ID.

Formula result =

[Help on this function](#) OK Cancel

7. Choose **OK**.

The formula is added to the cell. You can see the formula with its function and arguments in the formula bar when you select the cell in the sheet.

You can also click the Insert Function button, to open the [Function Argument](#) dialog and change your arguments.

<i>f_x</i>	=SAP.GETDIMENSIONFILTER("Table2";"Product";3;1;0)			
	B	C	D	
	Product			
	Carbonated Drinks	Self & Descendants		
	Juices - PC2	Self & Descendants		
	Coconut Water - PC	Self Only		

Related Information

[SAP.GETDIMENSIONFILTER \[page 22\]](#)

4 Add a data source for formulas

You can add data sources for formulas in your workbook. This allows you to create highly flexible and formatted reports. You can use it to enhance workbooks with tables with calculations or to design workbooks and sheets that don't adhere to the table-based layout.

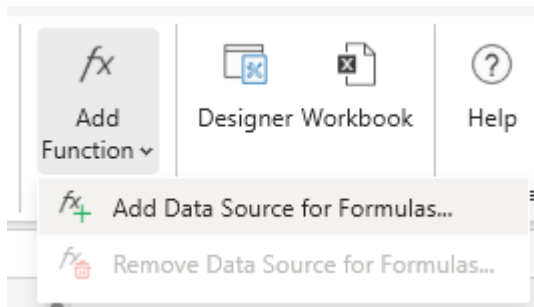
When you add a data source for formulas, no table is inserted in the sheet. Therefore, the panel doesn't show the builder panel with rows and columns, but an optimized formula panel with the data source name, the data source alias, the dimensions of the data source and the filter area.

You can use all data sources connected to the add-in.

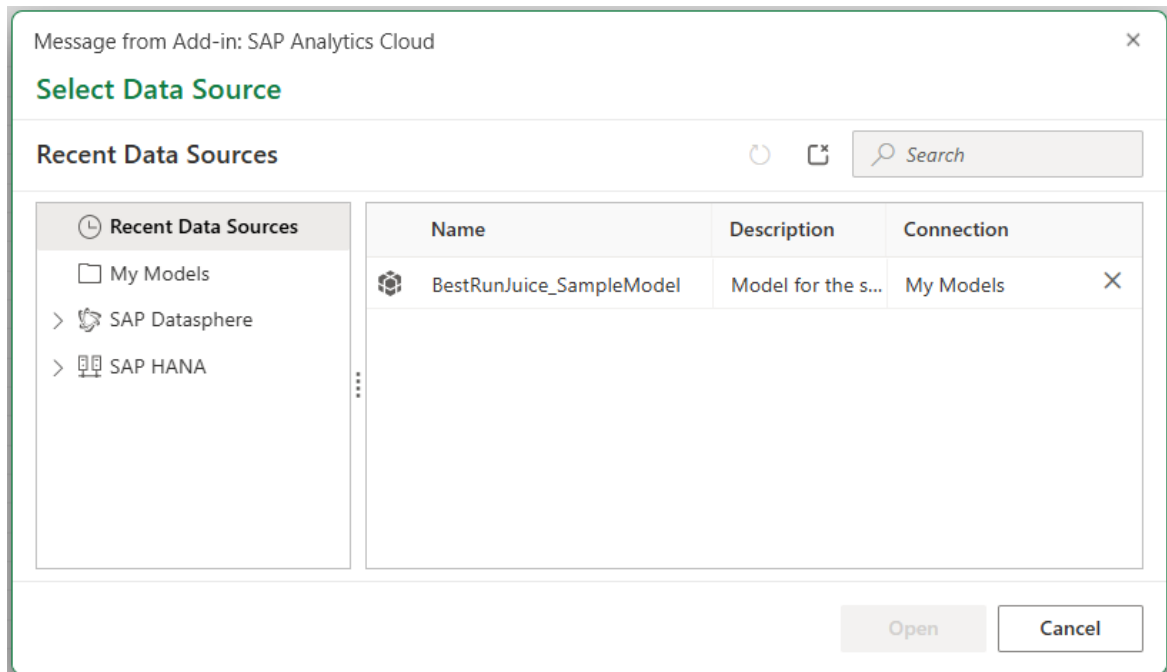
To add a data source for formulas:

1. Choose a cell in the Excel sheet.
2. Choose [Add Data Source for Formulas](#) in the Designer panel.

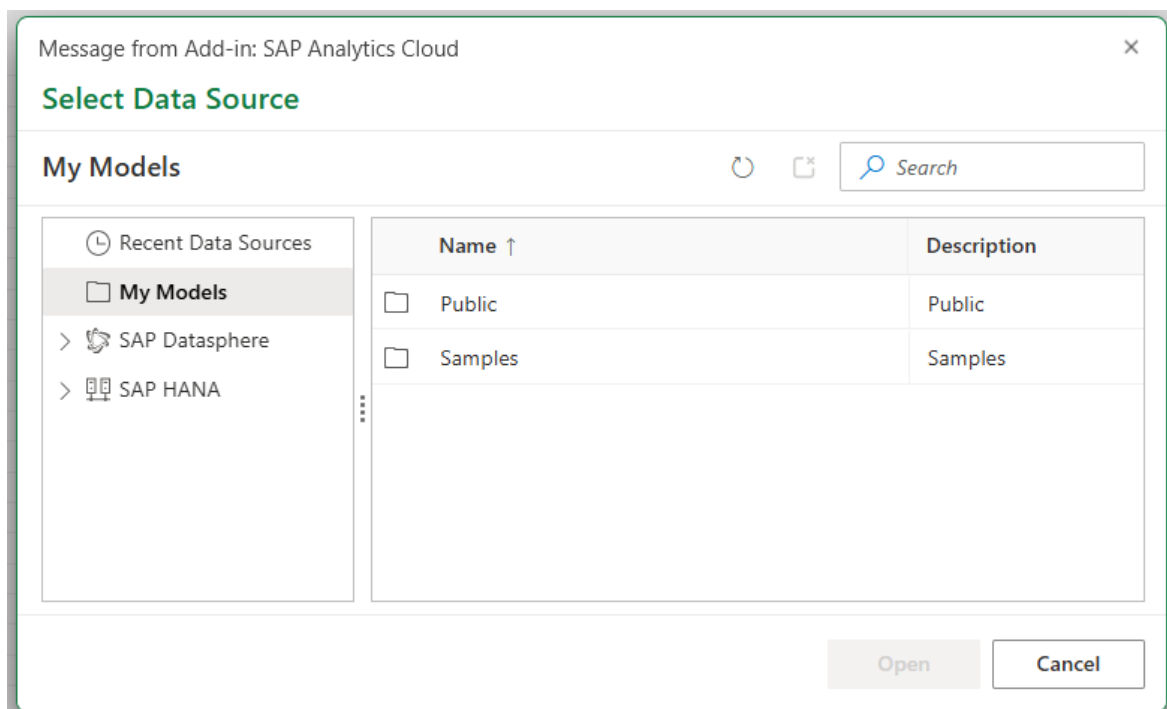
Alternatively, you can also choose ► [Add Function](#) ► [Add Data Source for Formulas](#) ► in the ribbon.



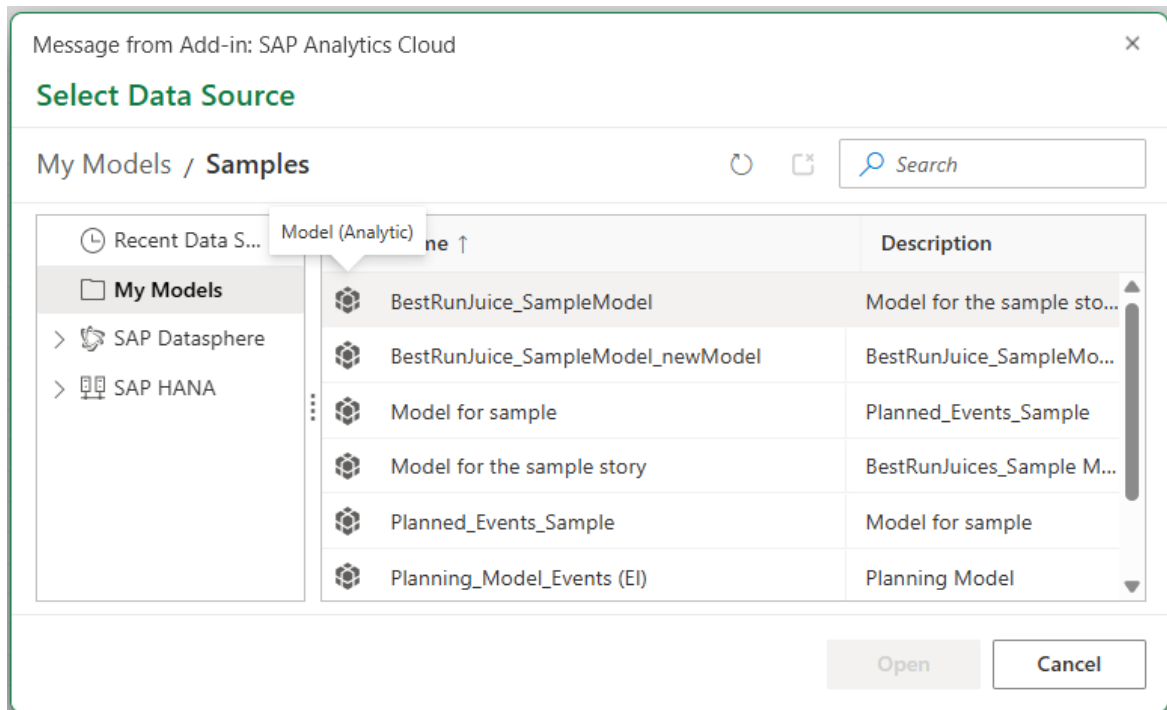
3. In the [Select Data Source](#) dialog, choose the connection where your data source is located.
Note: If you click on a text, the step is directly executed. If you click somewhere in the row, you need to click [OK](#) to complete the step.
The [My Models](#) connection is the connection to the SAP Analytics Cloud you're already connected to. You can choose a model there.
There can be other connections to SAP Datasphere and SAP S/4HANA Cloud tenants where you can select the corresponding data sources.



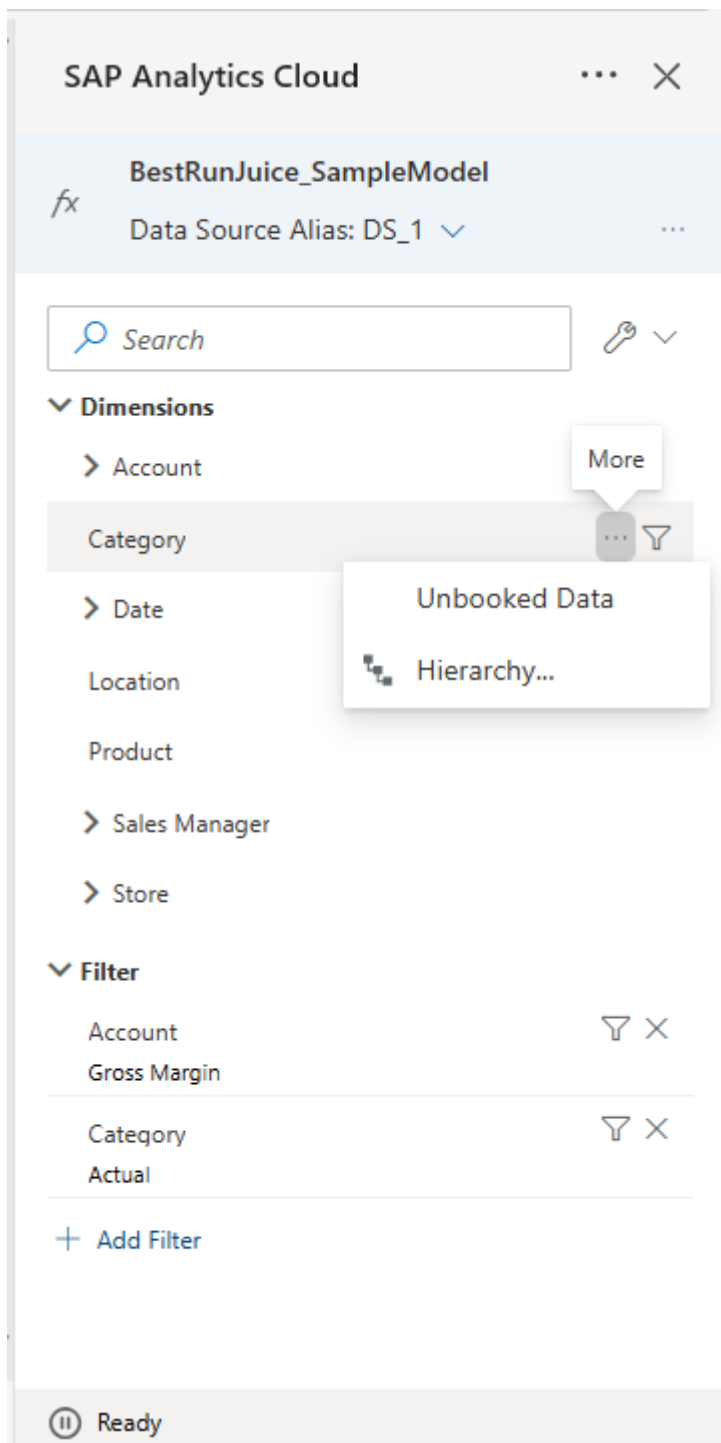
4. Choose a folder from the list.
You can choose one of the folders available in your SAP Analytics Cloud.



5. Choose a model from the list.
You can also enter a string in the Search field to find a model in the list.
If you want to go back to the connections list, you can click [Connections](#) in the breadcrumb.



- The data source is inserted with an SAP.GETDATA formula in the selected cell.
 The formula for the first data source will be `=SAP . GETDATA ( DS_1 )`. The data source alias DS_1 is generated automatically. Each data source you add for formulas will get a different alias, for example, DS_2 for the second data source you add.
 The panel shows the formula panel with the data source name, alias, dimensions, and filter section. You can use the context menu of a dimension to show and hide the unbooked members and define the hierarchy display.
 When adding more than one data source for formulas, you can use the drop-down box to switch between the data sources.

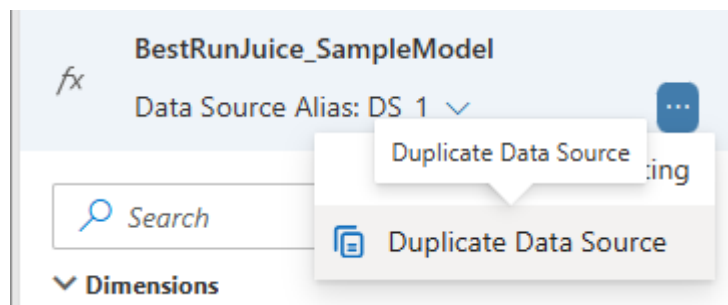


7. Change the formula definition.
 You can change a formula definition by changing the filter in the formula panel or by adding arguments to the formula, for example, `=SAP.GETDATA("Table1", "Account", "Gross_Margin", "Product", "PD19")`.
 After changing a formula, you can recalculate all functions in the sheet with ► [Refresh](#) ► [Refresh Sheet Functions](#) in the ribbon. Or you can select a subset of your functions and choose ► [Refresh](#) ► [Refresh Selected Functions](#).

8. Duplicate the data source.

Instead of changing the formula definition, you can alternatively duplicate the data source and change the filters and arguments there. This allows you to keep the original formula and create an additional one using the duplicated data source.

You duplicate a data source with the context menu of the data source in the builder panel:



After duplicating a data source, it has the same navigation state (filters, prompts, etc.) as the original one.

9. Add other formulas

You can add other formulas based on the data source you added for formulas, for example, an SAP.GETDIMENSIONS formula.

10. Remove a data source for formulas.

To remove a data source for formulas, choose ► [Add Function](#) ► [Remove Data Source for Formulas](#) ► in the ribbon.

A dialog appears where you can select the data sources you want to remove.

Not that the data sources will be removed, but the respective cells in the sheet are not cleaned. You can do that manually if needed.

When adding data formulas (SAP.GETDATA and SAP.SETDATA) in a workbook, a pre-defined specific format for the data is applied automatically. On the Cell format in Excel, you can specify other formats with the standard Excel options. You can use formulas with the pre-defined format and formulas with Excel formatting in the same workbook if they use different data sources.

If you want to use the pre-defined format again, select the cells and choose [General](#) in the Excel Number group.

5 SAP.ASYMMETRICFILTER

The function SAP.ASYMMETRICFILTER returns asymmetric filters on the selected tables. The returned filters overwrite the complex filters of an existing table and enable you to define asymmetric axes.

Note that you can't use this function with SAP BW and SAP S/4HANA Cloud data sources. Additionally, you can't use this function with time-related calculated measures (such as YTD, QTD...). When you use such measures with this formula, they will return empty values.

For a table on which an SAP.ASYMMETRICFILTER formula is applied (to make it asymmetric), you can't use data point commenting anymore.

The SAP.ASYMMETRICFILTER function has the following arguments:

- **Table**
Enter one or more table names in quotation marks.
You can enter the table names as a comma-separated list or as a range.
- **Block**
Enter one or more blocks from your Excel sheet.
The blocks include several rows of dimension and member combinations.
You can enter the blocks as Excel ranges (F6:G9) or as a reference to a block created with the SAP.BLOCK formula (#A10).

When executing the formula, the blocks are applied to the specified tables. To see the changes, you need to refresh the table or the tables. All existing asymmetric filters with an overlap of dimensions used are cleared, and the new asymmetric filter is applied.

Please note: If more than one formula is executed in a roundtrip (for example, with Refresh All) and at least two formulas have an overlap, the formulas with conflicts show an error message and the filters are not applied. In the refresh all scenario, you can then go to one of the formulas and execute it manually to apply this asymmetric filter.

The applied blocks are shown as filters in the filter area of the builder panel. You can't change the blocks in the builder panel, but you can delete them there if you don't longer need them.

You can apply several SAP.ASYMMETRICFILTER formulas to the same table.

Note

If you use dimension members of interconnected hierarchy levels positioned in two blocks (for example, 2024 in one block and 2024Q4 in a second block), to get the expected result, you need to position this dimension in first position in the row or column axis of the table, and then expand the displayed upper hierarchy node of this dimension in the table after having applied the asymmetric filter to the table.

Example

Cell B20: =SAP.ASYMMETRICFILTER("Table1", F6:G9, #A10)

After executing the formula and refreshing the table *Table1*, the asymmetric filters defined in the Excel sheet block F6:G9 and with the SAP.Block formula in cell A9 (referenced as #A10) are applied to Table1.

Related Information

[SAP.BLOCK \[page 17\]](#)

6 SAP.BLOCK

The function SAP.BLOCK returns one or several blocks that you can use in the function SAP.ASYMMETRICFILTER to build asymmetric reports.

The SAP.Block function has the following arguments:

- *Dimension*
Enter the dimension name or ID in quotation marks.
- *Member*
You can enter a single member, a list of members, and cells containing the members as a single cell or a range of cells.
Please note: In the formula, the comma is used as a separator. Therefore, you can't use member descriptions including a comma, like the member *France,Bordeaux*.

To create the formula, one dimension /member argument is mandatory. You can use up to five dimension and member arguments to define the block in the formula.

The created block can be referenced in the SAP.ASYMMETRICFILTER function.

Example

Cell A9: =SAP.BLOCK ("Version", "Actual", "Time", G4:G8)

If you enter the formula in cell A9, you create a block starting in cell A9 with the dimension *Version* and the member *Actual* in cell B9. In cell A10, the block continues with the dimension *Time* and the first member taken from cell G4. The other four Time members (G5-G8) are added in the same way.

You can reference the new block as an argument in a SAP.ASYMMETRICFILTER formula with #A10.

Related Information

[SAP.ASYMMETRICFILTER \[page 15\]](#)

7 SAP.GETDATA

The function SAP.GETDATA returns the data value for a specified set of dimension and member combinations.

→ Recommendation

Please consider the following recommendations for optimizing performance when using SAP.GETDATA formulas:

- We recommend not using more than 5000 SAP.GETDATA formulas in a workbook.
- Keep in mind that the sparsity of the data used in the SAP.GETDATA formulas can impact the performance.
- Be aware that the performance of SAP.GETDATA formulas depends on the amount of data in the model.
- Retrieving values on calculated members with SAP.GETDATA formulas impacts performance more significantly than retrieving values on standard dimension members.
- To improve the performance, you can enable the preference [Optimize Data Retrieval](#).

You can also use this formula in the offline mode. This means that the formula continues to show the last calculated value in offline workbooks, even if the workbook is modified. After reconnecting to a tenant, you can switch to online mode, and the formula refreshes automatically.

The SAP.GETDATA function has the following arguments:

- [Table](#)
Enter the table name in quotation marks.
- [Dimension and member](#)
You can enter a dimension ID followed by a member's ID in quotation marks.
Note that to retrieve the data, the formula needs a dimension and member information for each dimension of the data source.
You can specify the information for a measure dimension and several pairs of one dimension followed by one member to specify the data you want to retrieve.
If some dimensions of the data source are not specified in the formula, the add-in uses the specified dimension member for this dimension in the filter area of the panel. For dimensions that are not available in the panel, the default value for this dimension provided by the backend is used.
Please note that a virtual dimension member, such as "All members", cannot be used in the formula because it will not be recognized.
When a filter is added or changed while a SAP.GETDATA formula already exists, you need to refresh the SAP.GETDATA formula to retrieve the data corresponding to the current filter state. You can use the option [► Refresh ► Refresh Sheet Functions ►](#) in the ribbon to refresh all functions in the current sheet. Or you can select a subset of your functions and choose [► Refresh ► Refresh Selected Functions ►](#).
If a dimension is available in the formula setting and the panel, the dimension member specified in the formula is used for the formula result.

When using a table as data source for the formula, the default format for the data is applied automatically. You can disable the default format by choosing the context menu for data sources in the builder panel header and [Apply Default Format](#).

When using a data source for formula for the formula, a pre-defined specific format for the data is applied. On the Cell format in Excel, you can specify other formats with the standard Excel options. You can use formulas with the pre-defined format and formulas with Excel formatting in the same workbook if they use different data sources. If you want to use the pre-defined format again, select the cells and choose *General* in the Excel Number group.

For more information about performance optimization when using SAP.GETDATA formulas, you can see the SAP Note [3323835](#).

Example

Cell E9: =SAP.GETDATA("Table1", "Account", "Gross_Margin", "Product", "PD19")

If you enter the formula in cell E9, you get the data value for the measure dimension Account and its member (ID) Gross_Margin and the dimension ID Product and its member PD19 (Coconut Water) in cell E9.

In this example, you can see the corresponding data value in the table in cell C9:

E9					=SAP.GETDATA("Table1", "Account", "Gross_Margin", "Product", "PD19")
	A	B	C	D	E
1	BestRunJuice_SampleModel				
2					
3		Category	Actual		
4	Account	Product			
5	Gross_Margin Gross Margin	► PC4 Alcohol	\$34,032,932.49		
6		► PC1 Carbonated Drinks	\$62,082,721.78		
7		► PC2 Juices	\$137,505,126.23		
8		▼ PC3 Others	\$1,416,168.88		
9		PD19 Coconut Water	\$147,994.04		147994.04
10		PD20 Soda	\$190,897.37		
11		PD29 Sparkling Water	\$1,077,277.46		
12					

8 SAP.GETDATACOMMENT

The function SAP.GETDATACOMMENT returns the data point comment for a specified set of dimension and member combinations. If there is more than one comment for the specified dimension and member combination, the formula returns the last comment of the thread.

In the formula definition, you can also specify to return the data value in addition to the data point comment.

To use this function, data point commenting needs to be enabled, and the table must be based on an SAP Analytics Cloud model.

You can also use this formula in the offline mode. This means that the formula continues to show the last calculated value in offline workbooks, even if the workbook is modified. After reconnecting to a tenant, you can switch to online mode, and the formula refreshes automatically.

The SAP.GETDATACOMMENT function has the following arguments:

- **Table**
Enter the table or data source name in quotation marks.
- **Display**
Specify how the dimensions are displayed in the sheet. You have the following options:
 - **0** for the last comment of the thread (default)
 - **1** for data value on the left and last comment of the thread on the right
 - **2** for last comment of the thread on the left and data value on the right
 - **3** for data value above the last comment of the thread
 - **4** for last comment of the thread above the data value
- **Dimension and member**
You can enter a dimension ID followed by a member's ID in quotation marks.
Note that to retrieve the data comment, the formula needs a dimension and member information for each dimension of the data source.
You can specify the information for a measure dimension and several pairs of one dimension followed by one member to specify the data comment you want to retrieve.
If some dimensions of the data source are not specified in the formula, the add-in uses the specified dimension member for this dimension in the filter area of the panel. For dimensions that are not available in the panel, the default value for this dimension provided by the backend is used.
If a dimension is available in the formula setting and the panel, the dimension member specified in the formula is used for the formula result.
Please note that a virtual dimension member, such as "All members", cannot be used in the formula because it will not be recognized.

Example

Cell E11: =SAP.GETDATACOMMENT("Table1", 3, "Account", "Gross_Margin", "Product", "PD29")

If you enter the formula in cell E11, you get the data value for the measure dimension Account and its member (ID) Gross_Margin and the dimension ID *Product* and its member *PD29* (Sparkling Water) in cell E11, and the last comment of the thread is displayed in cell E12.

9 SAP.GETDIMENSIONFILTER

The function SAP.GETDIMENSIONFILTER returns the filter for a dimension and an icon to open the filter dialog from the sheet.

The SAP.GETDIMENSIONFILTER function has the following arguments:

- *Table*
Enter the table name in quotation marks.
- *Dimension*
Enter the dimension name in quotation marks.
- *Member_display*
Specify how the members of the dimension are displayed in the sheet. You have the following options:
 - 0 for ID
 - 1 for description
 - 2 for ID and description (default)
 - 3 for description and ID
 - 4 for full ID
 - 5 for ID and description in two separate cells
 - 6 for description and ID in two separate cells
- *Selection_display*
Specify the selection display:
 - 0 for ID
 - 1 for description
 - 2 for ID and description
 - 3 for description and ID
 - 4 for display nothing (default)
- *Orientation*
Specify the orientation of the list:
 - 0 for vertical (default)
 - 1 for horizontal
- *Separator*
Enter the separator you want to use between ID and description.

Example

Cell B2: =SAP.GETDIMENSIONFILTER("Table2","Product",3,1,0)

If you enter the formula in cell B2, you get the current filter applied for the dimension *Product* of table *Table2* starting in cell B2. You can choose the icon next to the dimension title to open the filter dialog to change the filter for the dimension.

In this example, you see the members displayed with description and ID (3), the selection with description (1), and the list has a vertical orientation (0).

fx =SAP.GETDIMENSIONFILTER("Table2","Product",3,1,0)			
B		C	D
Product			
Carbonated Drinks - PC1		Self & Descendants	
Juices - PC2		Self & Descendants	
Coconut Water - PD19		Self Only	

10 SAP.GETDIMENSIONS

The function SAP.GETDIMENSIONS lists all dimensions of a data source.

The SAP.GETDIMENSIONS function has the following arguments:

- *Table*
Enter the table name in quotation marks.
- *Display*
Specify how the dimensions are displayed in the sheet. You have the following options:
 - 0 for ID
 - 1 for description
 - 2 for ID and description (default)
 - 3 for description and ID
 - 4 for full ID
- *Orientation*
Specify the orientation of the list:
 - 0 for vertical (default)
 - 1 for horizontalIf both, ID and description, are displayed, the results are displayed accordingly in 2 rows or 2 columns.
- *Sort_order*
Specify the sorting order of the dimensions:
 - 0 for no sorting (default)
 - 1 for ascending sorting
 - 2 for descending sorting

Example

Cell B2: =SAP.GETDIMENSIONS("Table2";1; 0; 1)

If you enter the formula in cell B2, you get a list of all dimensions of the data source used for table *Table2*. The list starts in cell B2.

In this example, the list shows the dimensions with description (1), has a vertical orientation (0) and is sorted ascending (1).

11 SAP.GETPLANNINGTRIGGERNAME

The function SAP.GETPLANNINGTRIGGERNAME returns the planning trigger name from a cell.

The SAP.GETPLANNINGTRIGGERNAME function has the following arguments:

- *Cell*
Enter a cell that is part of the planning trigger.

12

SAP.GETPLANNINGTRIGGERPARAMETERS

The function SAP.GETPLANNINGTRIGGERPARAMETERS lists all parameters of a planning trigger.

The SAP.GETPLANNINGTRIGGERPARAMETERS function has the following arguments:

- *Planning Trigger*
Enter the name of the planning trigger in quotation marks.
- *Display*
Specify how the parameters are displayed in the sheet. You have the following options:
 - 0 for ID
 - 1 for description
 - 2 for ID and description (default)
 - 3 for description and ID
- *Orientation*
Specify the orientation of the list:
 - 0 for vertical (default)
 - 1 for horizontal
- *Sort Order*
Specify the sorting order of the dimensions:
 - 0 for no sorting (default)
 - 1 for ascending sorting
 - 2 for descending sorting

13

SAP.GETPLANNINGTRIGGERPARAMETERVALUE

The function SAP.GETPLANNINGTRIGGERPARAMETERVALUE lists the values of a parameter of a planning trigger and an icon to open a dialog where you can change the values from the sheet.

The SAP.GETPLANNINGTRIGGERPARAMETERVALUE function has the following arguments:

- *Planning Trigger*
Enter the name of the planning trigger in quotation marks.
- *Parameter*
Enter the name of the planning parameter in quotation marks.
- *Display*
Specify how the parameter values are displayed in the sheet. You have the following options:
 - 0 for ID
 - 1 for description
 - 2 for ID and description (default)
 - 3 for description and ID
- *Orientation*
Specify the orientation of the list:
 - 0 for vertical (default)
 - 1 for horizontal

14 SAP.GETTABLENAME

The function SAP.GETTABLENAME returns the table name of a selected cell.

The SAP.GETTABLENAME function has the following arguments:

- *Cell*
Enter a cell that is part of the table.

→ Recommendation

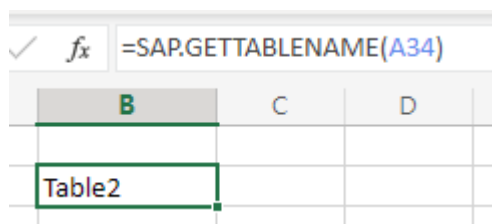
You can use the returned table name in other formulas that require the table name as a parameter, like SAP.GETDATA or SAP.SETDATA.

As the performance can be impacted if you reference directly the cell containing the SAP.GETTABLENAME formula in other formulas, we recommend to copy the result of the SAP.GETTABLENAME formula to another cell and reference this cell in the other formulas.

Example

Cell B2: =SAP.GETTABLENAME(A34)

If you enter the formula in cell B2, the name of the table to which cell A34 belongs is displayed in cell B2. In this example the table name is *Table2*.



The screenshot shows the SAP Excel interface. The formula bar at the top displays the formula `=SAP.GETTABLENAME(A34)` with a checkmark and the *fx* icon. Below the formula bar is a table grid with columns labeled B, C, and D. In row 2, cell B2 contains the text *Table2*, which is the result of the formula entered in that cell.

B	C	D
Table2		

15 SAP.GETTABLESTATUS

The function SAP.GETTABLESTATUS returns the refresh status of the defined tables.

The SAP.GETTABLESTATUS function has the following arguments:

- **Tables**
Enter one or several table names in quotation marks or a range.
You can set the tables separated by a comma, for example. "Table1, Table2, Table3", or you can use a cell range from the sheet, for example, K13:K15.
- **Display**
Specify the information you want to see in the sheet:
 - 0 for table name and status (default)
 - 1 for status
 - 2 for status and status message
 - 3 for table name and status message
 - 4 for table name, status and status message

Depending on the formula definition, it can return the following information per table:

Table Name	Status	Message
Table Name	0	Table refreshed.
(for example, Table1)	1	Table not refreshed.
	2	Remote connection logon error.
	3	Table data source not available.
	4	Table with pending filters.

Example

Cell E5: =SAP.GETTABLESTATUS("Table1, Table2"; 4)

If you enter the formula in cell E5, you get the refresh status for the defined tables *Table1* and *Table2* with the table name, status and status message (Display=4).

In this example, the table status shows that *Table1* has been refreshed and *Table2* hasn't been refreshed yet.

E5		✕ ✓ f_x		=SAP.GETTABLESTATUS("Table1, Table2"; 4)				
	A	B	C	D	E	F	G	H
1	BestRunJuice_SampleModel							
2	in m USD							
3	Account	Product						
4	Gross Margin	▸ Alcohol	34,03					
5		▸ Carbonated Drinks	62,08		Table1	0	Table refreshed.	
6		▸ Juices	137,51		Table2	1	Table not refreshed.	
7		▸ Others	1,42					
8								

16 SAP.GETVARIABLES

The function SAP.GETVARIABLES returns the variable values of a data source.

The SAP.GETVARIABLES function has the following arguments:

- **Table**
Enter the table name or data source alias in quotation marks.
- **Variables**
Enter a comma-separated list of variable IDs.
You can also refer to a single cell or a range of cells containing the variables you want to use in the formula.
This parameter is optional. If you don't specify a variable, the formula will return all values from all variables of the data source.
- **Variable Display**
Specify the variable display:
 - 0 to display the ID of the variable
 - 1 to display the description of the variable (default value)This argument is optional. If you don't specify it, the default value is applied.
- **Value Display**
Specify the value display:
 - 0 to display the ID of the variable value (default value)
 - 1 to display the description of the variable valueThis argument is optional. If you don't specify it, the default value is applied.
- **Layout**
Specify the layout:
 - 0 for the variable in a cell and all values in one cell
With this option, all values are displayed in the same cell, separated by carriage returns. You can manually wrap the text or merge cells to adjust the display.
 - 1 for the variable in a cell and the values in separated cells (default)
 - 2 for all variable values in one cell without the variable
With this option, all values are displayed in the same cell, separated by carriage returns. You can manually wrap the text or merge cells to adjust the display.
 - 3 for the variable values in separated cells without the variableThis argument is optional. If you don't specify it, the default value is applied.
- **Orientation**
Specify the orientation of the list:
 - 0 for vertical
 - 1 for horizontal (default)This argument is optional. If you don't specify it, the default value is applied.

The variable values are returned according to your definition.

Please note the following behavior:

- If no variable can be found for the specified data source, an error message is displayed.

- If the variable values don't have a description, the formula will automatically display the ID.
- For SAP BW and SAP S/4HANA live connections, the description of the variable value can't be retrieved in view mode. Therefore, an error message is displayed.

Example

Cell D2: =SAP.GETVARIABLES("Table1", "City", 1, 1, 3, 0)

If you enter the formula in cell D2, you get the values for the variable City displayed in separated cells starting in cell D2.

The variable values are displayed in separated cells (3) with description (1) and in a vertical orientation (0) starting in cell D2. The variable *City* is not displayed due to the layout specification 3.

17 SAP.MEMBERPROPERTY

The function SAP.MEMBERPROPERTY returns the value of a specified property for specified dimension members.

→ Recommendation

To maintain good performance when working with the add-in, we recommend relying on range selection to select a large number of members at once and ideally using one single SAP.MEMBERPROPERTY formula per dimension.

You can also use this formula in the offline mode. This means that the formula continues to show the last calculated value in offline workbooks, even if the workbook is modified. After reconnecting to a tenant, you can switch to online mode, and the formula refreshes automatically.

The SAP.MEMBERPROPERTY function has the following arguments:

- *Table*
Enter the table name or data source alias in quotation marks.
- *Dimension*
Enter the dimension name or ID in quotation marks.
- *Dimension Member*
Enter one or more dimension member IDs. You have the following options to enter values directly or as a cell reference:
 - One member ID directly in the formula
 - One member ID as a cell reference
 - Several member IDs as a range referenceIf you enter more than one member ID, the formula returns the property value for each specified dimension member.
- *Property*
Enter one or more dimension member properties with the description or ID. You can enter the values directly or as a cell reference.
To use the workbook with the formula in different languages, we recommend using the property ID, as it is language-independent. Alternatively, you can use the property description as long as the data access language for the table is the same as the language used in the formula
If you enter more than one property, the formula returns the values of the different specified properties for each dimension member specified in the previous argument.
Note that a blank value is displayed for a dimension member if the specified property cannot be found for this dimension member.
- *Orientation*
Specify the orientation of the list:
 - *0* for vertical (default)
 - *1* for horizontalThis argument is optional. If you don't specify it, the default value is applied.

Example

Cell B2: =SAP.MEMBERPROPERTY("Table1","Customer", E10, "Description, Country (Short_Text)", 0)

If you enter the formula in cell B2, you get the properties description and short text for the country for the member of the dimension *Customer* that is specified as ID in cell E10.

The properties description and the short text for the country are displayed in a vertical orientation (0) starting in cell B2.

18 SAP.OVERWRITEDIMENSIONFILTER

The function SAP.OVERWRITEDIMENSIONFILTER overwrites the selected members of a dimension on the selected tables and thus replaces the current filter. The function always returns the new filter displayed with ID.

The SAP.OVERWRITEDIMENSIONFILTER function has the following arguments:

- **Tables**
Enter one or several table names or data source aliases in quotation marks or a range.
You can set the tables separated by a comma, for example. "Table1, Table2, Table3", or you can use a cell range from the sheet, for example, K13:K15.
For data source aliases, you can also use a single alias (DS_1), a list of aliases ("DS_1, DS_2, DS_3"), and a combination of tables and aliases ("Table1, DS_3").
- **Dimension**
Enter the dimension name in quotation marks.
You can also add a hierarchy or a version to specify the dimension further. This can be added directly after the dimension name or separated with a slash.
- **Members**
Enter the name of the members you want to use for the filter. You can enter a single member directly.
If you want to enter a range of members, the members should be referenced from the sheet. For example, the members are in the range F10-F15 in the sheet, then you can reference that range in the formula.
- **Selection_types**
Specify the selection types:
 - 0 for Self and Descendants (default)
 - 1 for Descendants only
 - 2 for Self and Children
 - 3 for Children Only
 - 4 for Self Only
 - 5 for Leaf Members OnlyYou can enter the selection type directly or reference the values from the sheet.
Please note the following behavior: If you enter 2 for Self and Children, only the defined member and its children are returned by the formula, even if you have also entered additional members in the *Members* argument.
- **Member_display**
Specify how the members of the dimension are used in the formula. You have the following options:
 - 0 for ID
 - 1 for description (default)
 - 2 for ID and description
 - 3 for description and ID
 - 4 for full IDThe member display that you specify here, must be also used for the *Members* argument in this formula.
For example, if you enter 0 for the member display, the member that you enter or reference in the *Members* argument must be displayed with their ID.
- **Separator**

Enter the separator you want to use between ID and description.
This argument is optional.

Example

Cell B6:=SAP.OVERWRITEDIMENSIONFILTER("Table2";"Product";"Juices";2;1)

If you enter the formula in cell B6, you overwrite the current filter applied for the dimension *Product* of table *Table2*. The new filter is displayed in cell B6.

In this example, we overwrite the existing filter that was returned by the example for the function SAP.GETDIMENSIONFILTER.

You see the new filter with ID *PC2* for the dimension *Product* displayed in cell B6. The member is displayed with children (2) and you have entered the member with description (1).

To apply the new filter, refresh the table *Table2* and then recalculate the SAP.GETDIMENSIONFILTER formula to get the new filter in cell B2:

fx =SAP.OVERWRITEDIMENSIONFILTER("Table2";"Product";"Juices";2;1)		
B	C	
Product		
Juices - PC2	Self & Children	
Product: PC2		

Related Information

[SAP.GETDIMENSIONFILTER \[page 22\]](#)

19 SAP.SETDATA

The function SAP.SETDATA sets a new data value for the specified dimension and member combinations. So, you can use it to enter new data values for planning.

When using dimensions with hierarchies in a formula, you can only enter new values on the leaf level in the formula. However, if a dimension is not specified in the formula, the add-in uses the filter set for that dimension in the panel, and that filter can also be set at the node level. Therefore, you can define values at the node level for a formula by creating a corresponding filter in the panel.

→ Recommendation

Please consider the following recommendations for optimizing performance when using SAP.SETDATA formulas:

- We recommend not using more than 5000 SAP.SETDATA formulas in a workbook.
- Keep in mind that the sparsity of the data used in the SAP.SETDATA formulas can impact the performance.
- Be aware that the performance of SAP.SETDATA formulas depends on the amount of data in the model.
- To improve the performance, you can enable the preference [Optimize Data Retrieval](#).

When using SAP.SETDATA formulas with the optimized planning area, the filter of the optimized planning area is defined based on the combination of the table filters and the dimension members used in the refreshed formulas. To do so, the add-in follows these rules:

- If a dimension is filtered but also used in formulas, the add-in takes the members used by the formulas.
- If a dimension is filtered but not used in formulas, the add-in takes the dimension filter.

Additionally, you can use SAP.SETDATA formulas to extend the current planning area. So, you can continue planning without the need to publish the planning data. After an input to a cell outside of the planning area, the area is extended, and the input is accepted.

You can also use this formula in the offline mode. This means that the formula continues to show the last calculated value in offline workbooks, even if the workbook is modified. After reconnecting to a tenant, you can switch to online mode, and the formula refreshes automatically.

The SAP.SETDATA function has the following arguments:

- **Table**
Enter the table name or data source alias in quotation marks.
- **Value**
Enter the data value you want to send to the server.
The format of the data value must match with the number format currently applied to the cell.
When using a table as data source for the formula, the default format for the data is applied automatically. You can disable the default format by choosing the context menu for data sources in the builder panel header and [Apply Default Format](#).
When using a data source for formula for the formula, a pre-defined specific format for the data is applied. On the Cell format in Excel, you can specify other formats with the standard Excel options. You can use formulas with the pre-defined format and formulas with Excel formatting in the same workbook if they use

different data sources. If you want to use the pre-defined format again, select the cells and choose [General](#) in the Excel Number group.

Without adding a value, the function behaves like the SAP.GETDATA function and returns the current value of the specified dimension and member combinations.

- [Dimension and member](#)

You can enter a dimension ID followed by a member's ID in quotation marks.

Note that to send the data, the formula needs a dimension and member information for each dimension of the data source.

You can specify the information for a measure dimension and several pairs of one dimension followed by one member to specify the data you want to retrieve.

If some dimensions of the data source are not specified in the formula, the add-in uses the specified dimension member for this dimension in the filter area of the panel. For dimensions that are not available in the panel, the default value for this dimension provided by the backend is used.

Please note that a virtual dimension member, such as "All members", cannot be used in the formula because it will not be recognized.

When a filter is added or changed while a SAP.SETDATA formula already exists, you need to refresh the SAP.SETDATA formula to set the data corresponding to the current filter state. You can use the option

 [Refresh](#)  in the ribbon to refresh all functions in the current sheet. Or you can select a subset of your functions and choose  [Refresh](#) .

If a dimension is available in the formula setting and the panel, the dimension member specified in the formula is used for the formula result.

When you execute the formula, the add-in recognizes that there is a new value for the specified dimension and member combinations, and the option [Process Data](#) in the ribbon is enabled. You can choose it to submit your changes to the database. You can also enter several new data values before you process them.

After submitting your changes, you can continue entering planning values or publish or revert your changes.

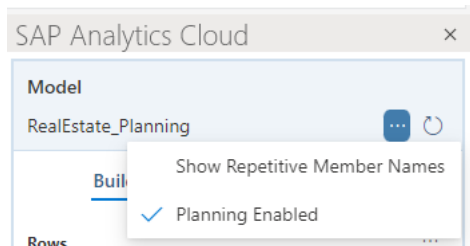
Please note the following behavior when entering planning values with an SAP.SETDATA formula:

- You have two options to enter planning values in Excel: You can add values with the SAP.SETDATA formula, and you can enter values manually using the mass data entry mode.
The manual planning of the add-in is based on the Mass Data Entry mode. In this mode, you can enter single and multiple values in a table. For more information, you can see the planning documentation at [Plan data](#).
When you enter new values with the formula and manually in your workbook and choose [Process Data](#), only the values entered manually are submitted to the database at first. Once this step is completed, the [Process Data](#) option is still enabled, and you can choose it to submit the values entered with the formula.
- With the option [Process Data](#), you submit all changes available in the workbook and not only the changes of the sheet you have currently selected.
- You can enter values when the workbook is not connected. After connecting to the tenant and refreshing the data functions, the add-in recognizes the new values, and you can process your changes.
- After processing the planning values entered with a formula, you get a message about how many of your changes have been submitted to the database. The message could say, for example, *4 out of 5 changes submitted to the database*. If not all of your changes have been submitted, one reason could be that you have executed a formula on a hierarchy node, and you can only submit values on the leaf level when working with hierarchies.

Example

The following procedure shows how you can enter planning data with an SAP.SETDATA formula. The planning model in this example is *RealEstate_Planning*.

1. Check if your model is planning enabled.
This step is a recommendation to check if your model is planning enabled before you start planning. You can see the status of the planning model in the model's context menu in the designer panel:



2. Enter the formula in cell E8: `=SAP.SETDATA("Table1";F5;"Version";"public.Strategic Growth Plan";"Measure";"sqm per Workplace")`
This formula sets a new value in Table1 in your workbook for the dimension member combinations for *Version* with member *Strategic Growth Plan* and *Measure* with member *sqm per Workplace*. The table displays the dimensions and members with their ID that must be used in the formula. The current value is 579, and the new value is 650. The new value is taken from cell F5 as defined in the formula.
You could also define the new value directly in the formula. Then the formula would be:
`=SAP.SETDATA("Table1";650;"Version";"public.Strategic Growth Plan";"Measure";"sqm per Workplace")`
After entering the formula in cell E8, it returns the current value for this dimension member combination: 579. And the option *Process Data* in the ribbon is enabled as the add-in recognizes that there is a new value.

Version	public.Actual	public.Strategic Growth Plan
Capacity	sqm520.000	sqm1.320.000
Workspace	sqm398.000	sqm1.104.960
Headcount	HC31.239	HC96.591
sqm per Workplace	500	579
sqm per Workplace %	—	33,0 %
OverheadCostPerHC	2.500	10.250
OverheadCost	5,28 m	38,23 m

3. Choose *Process Data* in the ribbon to submit the new value to the database.
The new value is now shown in the formula cell E8. The value for this dimension member combination in the table (cell C8) has not yet changed.

RealEstate_Planning

Version	public.Actual	public.Strategic Growth Plan
Capacity	sqm520.000	sqm1.320.000
Workspace	sqm398.000	sqm1.104.960
Headcount	HC31.239	HC96.591
sqm per Workplace	500	579
sqm per Workplace %	—	33,0 %
OverheadCostPerHC	2.500	10.250
OverheadCost	5,28 m	38,23 m

SAP Analytics Cloud

Model: RealEstate_Planning

Builder: Number Format

Rows: Measure

Columns: Version

Filter: Measure, Version

1 out of 1 have been successfully sent to the server.

4. Refresh the table.

To refresh the table, you can choose **Refresh** **Refresh Table** in the ribbon or click the icon in the designer panel.

After the refresh, the new value is also shown in the table (cell C8).

RealEstate_Planning

Version	public.Actual	public.Strategic Growth Plan*
Capacity	sqm520.000	sqm1.320.000
Workspace	sqm398.000	sqm1.104.960
Headcount	HC31.239	HC96.591
sqm per Workplace	500	650
sqm per Workplace %	—	33,0 %
OverheadCostPerHC	2.500	10.250
OverheadCost	5,28 m	34,05 m

SAP Analytics Cloud

Model: RealEstate_Planning

Builder: Number Format

Rows: Measure

Columns: Version

Filter: Measure, Version

You can now continue planning and set new data values in your workbook. To change the value for the dimension member combination of this example, you only need to change the value in cell F5.

When you finished your planning, you can publish or revert your changes.

20 SAP.TIMEOFFSET

The function SAP.TIMEOFFSET returns a time dimension member based on another time dimension member defined as the reference date.

Please be aware of the current restrictions when using SAP.TIMEOFFSET:

- The function only works on hierarchies, not on flat lists.
- You can't use the function for SAP BW and SAP S/4HANA data sources.

The SAP.TIMEOFFSET function has the following arguments:

- *Table*
Enter the table name or data source alias in quotation marks.
- *Reference Date*
Enter the reference date directly or as a cell reference containing the value.
- *Offset Value*
Enter a comma-separated list of offset values, for example, +1.
- *Display*
Specify the selection display:
 - 0 for ID (default)
 - 1 for description
 - 2 for ID and description (displayed in two separate cells)
 - 3 for description and ID (displayed in two separate cells)
- *Orientation*
Specify the orientation of the list:
 - 0 for vertical
 - 1 for horizontal (default)
- *Dimension setting*
This argument is optional.
If you want to apply the formula to another dimension that is not the default time dimension of the data source, you can specify this other dimension here, for example, the dimension *Product*.

Example

Cell B5: =SAP.TIMEOFFSET("Table2","2022", "-1, 0, 1, 2")

If you enter the formula in cell B5, you get the following dates calculated based on the reference date:

- 2021 in cell B5 (reference date 2022-1)
- 2022 in cell C5 (reference date 2022+/-0)
- 2023 in cell D5 (reference date 2022+1)
- 2024 in cell E5 (reference date 2022+2)

21 Keyboard shortcuts for the add-in

You can use key combinations (shortcuts) for some options when working with the add-in.

The shortcuts are available for Windows and Mac.

Command	Windows Key Combination	Mac Key Combination
Refresh All	Shift + Ctrl + Alt + F5	control + option + command + F5
Refresh Sheet	Shift + Ctrl + Alt + F6	control + option + command + F6
Refresh Sheet Functions	Shift + Ctrl + Alt + F10	control + option + command + F10
Refresh Selected Functions	Shift + Ctrl + Alt + F8	control + option + command + F8
Clear Data	Shift + Ctrl + Alt + D	control + option + command + D
Deactivate / Activate automatic refresh	Shift + Ctrl + Alt + F7	control + option + command + F7
Open designer panel	Shift + Ctrl + Alt + ↑	control + option + command + ↑
Close designer panel	Shift + Ctrl + Alt + ↓	control + option + command + ↓
Expand hierarchy node	Shift + Ctrl + Alt + →	control + option + command + →
Collapse hierarchy node	Shift + Ctrl + Alt + ←	control + option + command + ←
Process data	Shift + Ctrl + Alt + Enter	control + option + command + Enter
Enable / Disable a model for planning	Ctrl + Alt + I	control + option + command + I
Execute a mouse click on a cell, for example, choosing the pencil icon of the functions SAP.GETDIMENSIONFILTER and SAP.GETPLANNINGTRIGGERPARAMETERVALUE in the sheet	Shift + Ctrl + Alt + Space	control + option + command + Space

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.

