PUBLIC

SAP IBP, add-in for Microsoft Excel

How to use the sample VBA templates provided in SAP Note 1790530

SAP Digital Supply Chain

THE BEST RUN

TABLE OF CONTENTS

INTRODUCTION

Planning view template development is an important task during an SAP IBP implementation.

Useful, easy-to use, and good-looking templates will contribute significantly to end user adoption and implementation project success. The SAP IBP sample templates enhanced with VBA coding contain some standard formatting, functions and settings that can be generally useful.

The VBA templates provided by SAP are samples that can be used to kickstart your implementation of SAP IBP. Please feel free to customize and enhance the VBA code to your preference and towards the needs of your end users.

*The versions of the VBA samples included in the folder "**VBA_Template_2302_andOnwards**" can only be used in combination with SAP IBP, add-in for Microsoft Excel (Excel add-in) versions 2302.2.0 and above.*

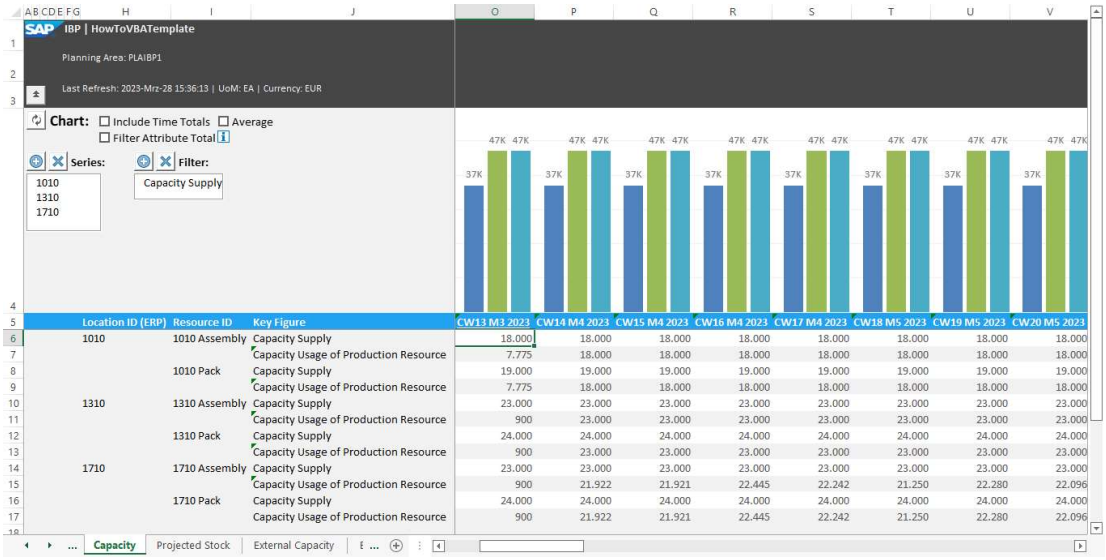
CHANGES COMPARED TO PREVIOUS VERSIONS

- Updated SAP IBP Formatting Sheet based on the enhanced formatting for specific member or properties, see What's New 2302: Enhanced Formatting for a Specific Member or Property.
- Revised template header section to support the "Scroll to current time period" feature, see What's New 2302: Scrolling to Current Time Period When Planning Views Are Opened

Advantages of using VBA templates

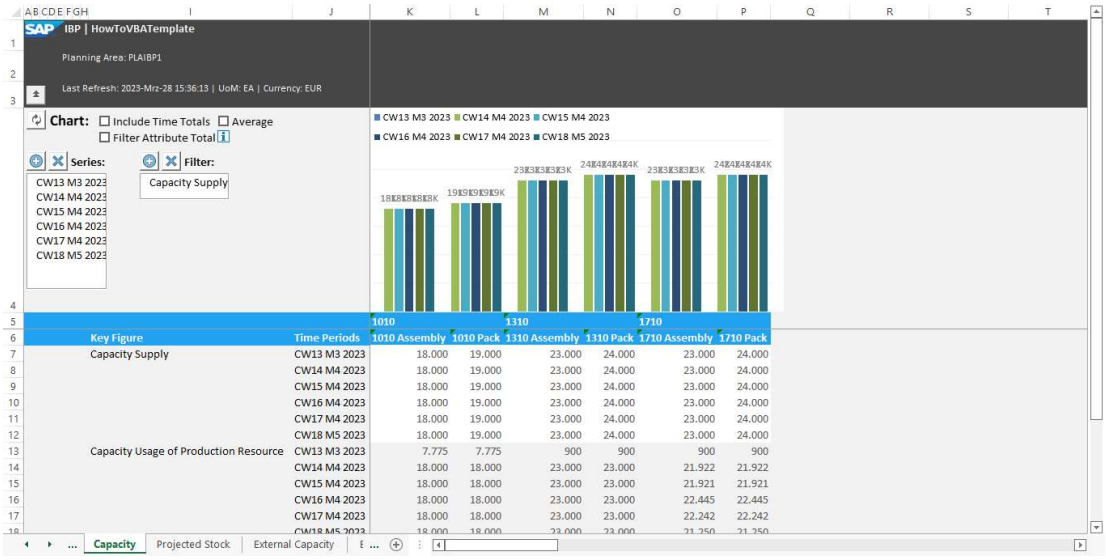
VBA-driven charts provide quite some flexibility to graphically display your planning data. The template can handle non-standard layouts and can plot different members such as attribute values, key figures, and time periods.

Consider this chart, where we compare the total available capacity in three plants:



Location ID (ERP)	Resource ID	Key Figure	CW13 M3 2023	CW14 M4 2023	CW15 M4 2023	CW16 M4 2023	CW17 M4 2023	CW18 M5 2023	CW19 M5 2023	CW20 M5 2023
1010	1010 Assembly	Capacity Supply	18.000	18.000	18.000	18.000	18.000	18.000	18.000	18.000
		Capacity Usage of Production Resource	7.775	18.000	18.000	18.000	18.000	18.000	18.000	18.000
	1010 Pack	Capacity Supply	19.000	19.000	19.000	19.000	19.000	19.000	19.000	19.000
		Capacity Usage of Production Resource	7.775	18.000	18.000	18.000	18.000	18.000	18.000	18.000
1310	1310 Assembly	Capacity Supply	23.000	23.000	23.000	23.000	23.000	23.000	23.000	23.000
		Capacity Usage of Production Resource	900	23.000	23.000	23.000	23.000	23.000	23.000	23.000
	1310 Pack	Capacity Supply	24.000	24.000	24.000	24.000	24.000	24.000	24.000	24.000
		Capacity Usage of Production Resource	900	23.000	23.000	23.000	23.000	23.000	23.000	23.000
1710	1710 Assembly	Capacity Supply	23.000	23.000	23.000	23.000	23.000	23.000	23.000	23.000
		Capacity Usage of Production Resource	900	21.922	21.921	22.445	22.242	21.250	22.280	22.096
	1710 Pack	Capacity Supply	24.000	24.000	24.000	24.000	24.000	24.000	24.000	24.000
		Capacity Usage of Production Resource	900	21.922	21.921	22.445	22.242	21.250	22.280	22.096

Or this, where we compare the available capacity of six resources over six weeks:

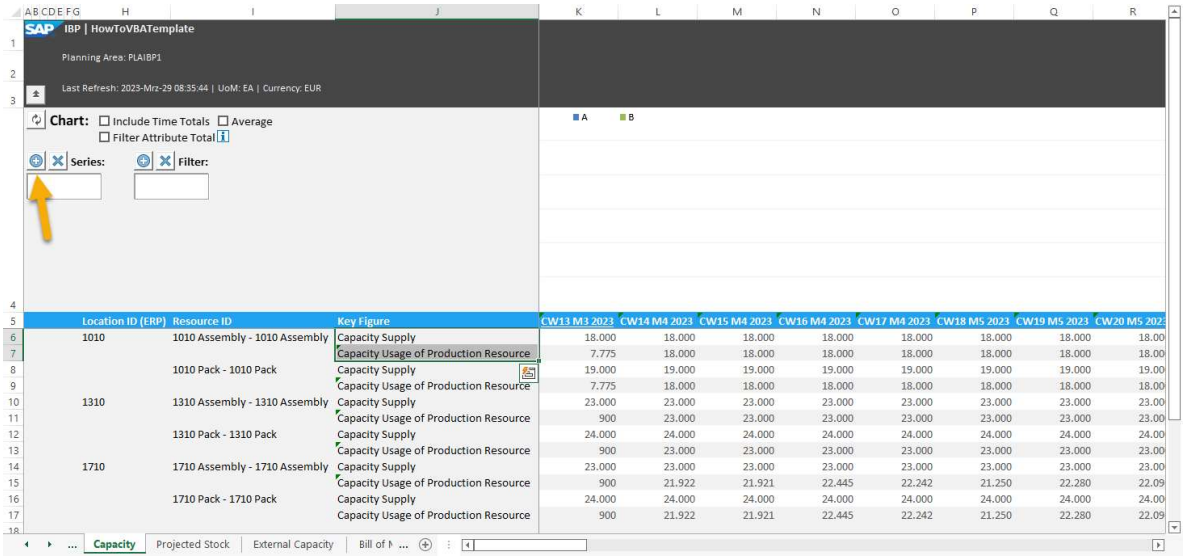


			1010		1310		1710	
Key Figure	Time Periods		1010 Assembly	1010 Pack	1310 Assembly	1310 Pack	1710 Assembly	1710 Pack
Capacity Supply	CW13 M3 2023		18.000	19.000	23.000	24.000	23.000	24.000
	CW14 M4 2023		18.000	19.000	23.000	24.000	23.000	24.000
	CW15 M4 2023		18.000	19.000	23.000	24.000	23.000	24.000
	CW16 M4 2023		18.000	19.000	23.000	24.000	23.000	24.000
	CW17 M4 2023		18.000	19.000	23.000	24.000	23.000	24.000
	CW18 M5 2023		18.000	19.000	23.000	24.000	23.000	24.000
Capacity Usage of Production Resource	CW13 M3 2023		7.775	7.775	900	900	900	900
	CW14 M4 2023		18.000	18.000	23.000	23.000	21.922	21.922
	CW15 M4 2023		18.000	18.000	23.000	23.000	21.921	21.921
	CW16 M4 2023		18.000	18.000	23.000	23.000	22.445	22.445
	CW17 M4 2023		18.000	18.000	23.000	23.000	22.242	22.242

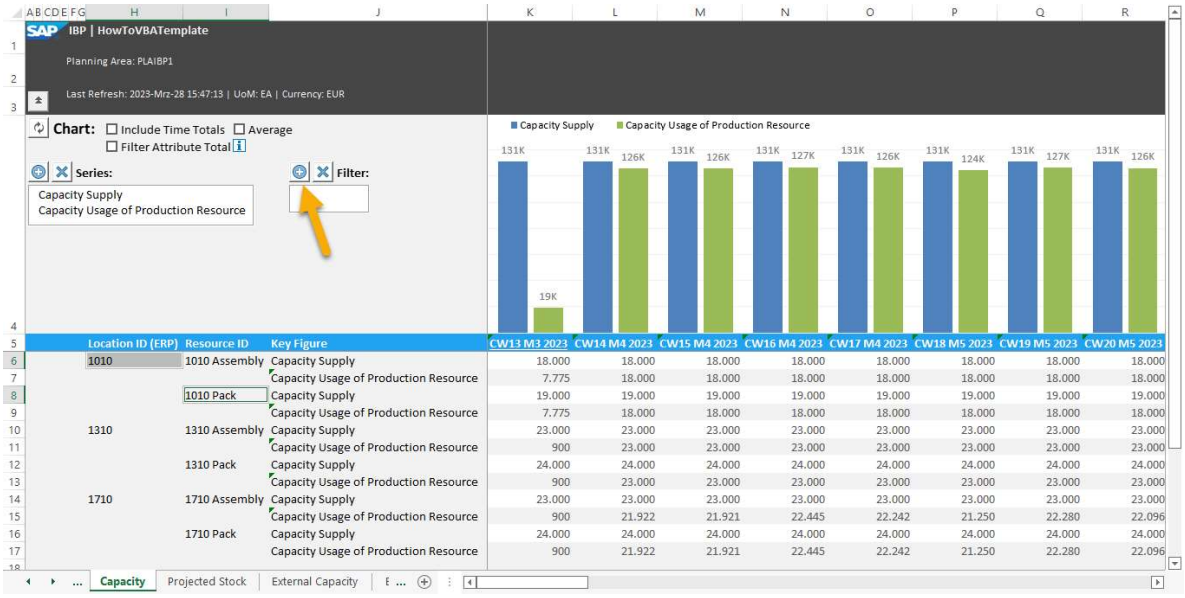
Caution: *This flexibility makes it easy to add apples to oranges. If, for example, we hadn't filtered by Available Capacity in above chart, the chart would have summed up capacity and usage.*

How to work with the VBA templates

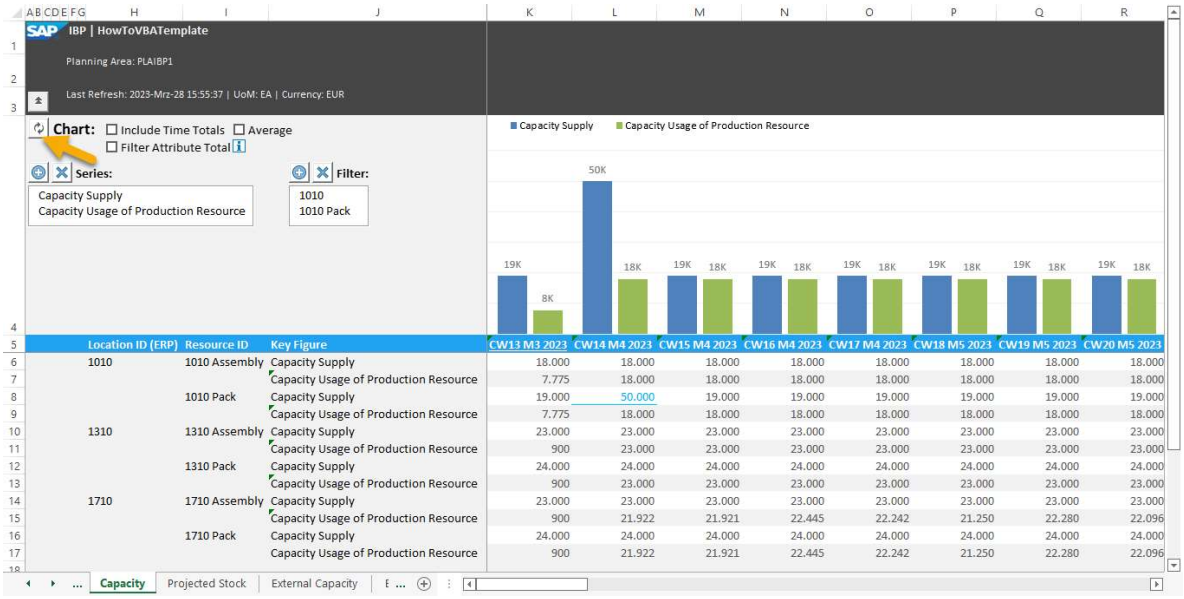
To define your chart's content, select the key figures you want to plot in your row axis and press the button next to ***Series:***



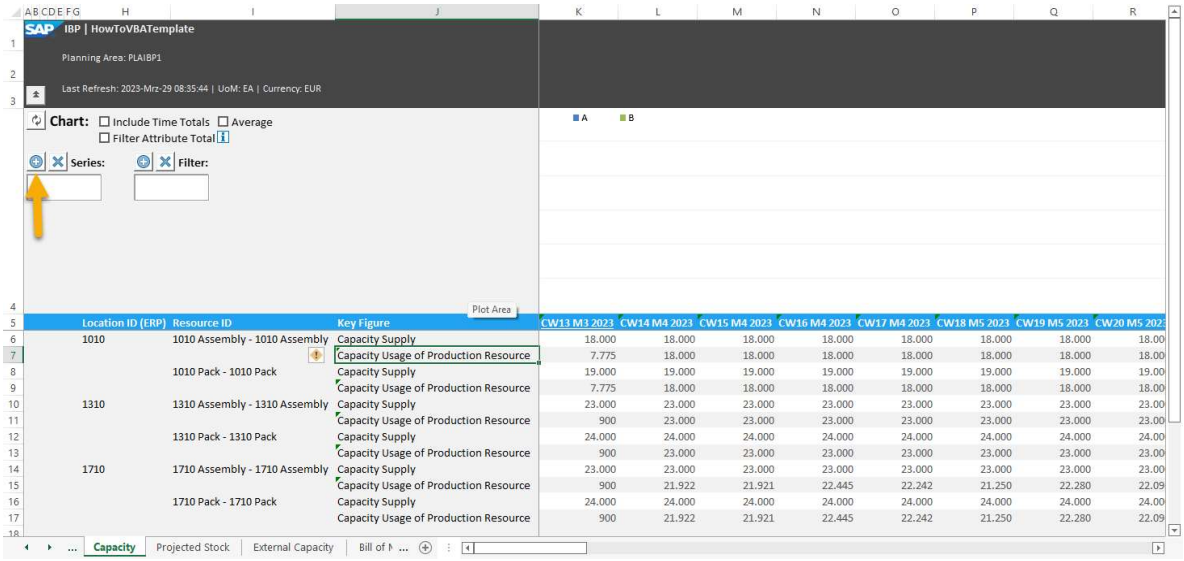
The system will note your selection in the box below and adjust the chart. To restrict the chart to certain attribute values, select those values and press the button next to ***Filter:***



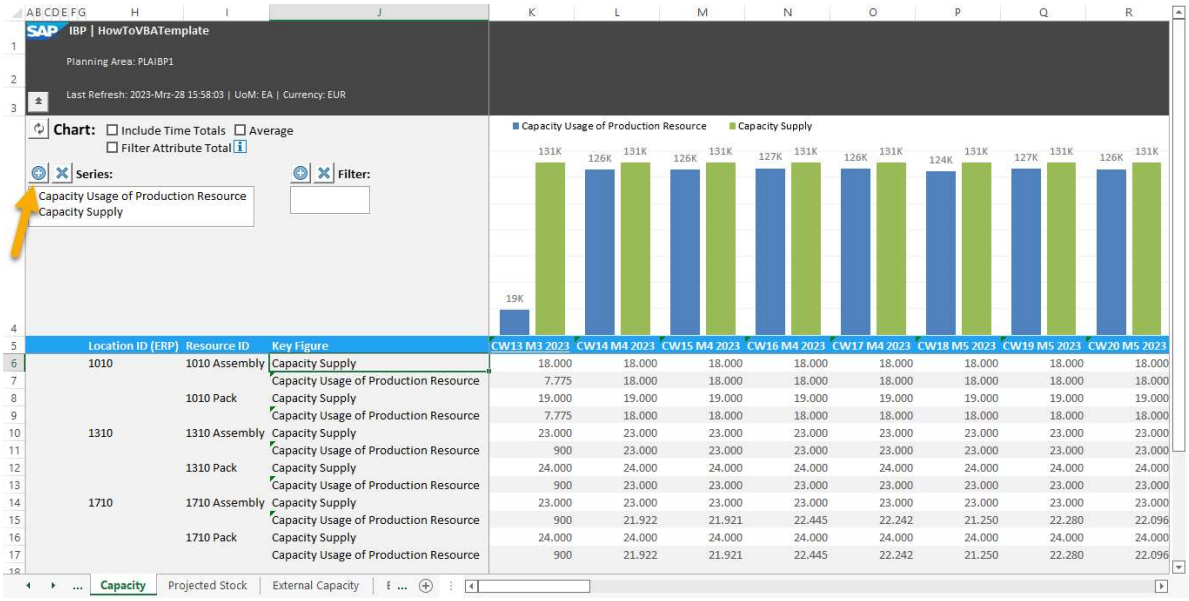
The system will note your selection in the box below and adjust the chart. To refresh the chart after manual edits, press the button next to ***Chart:***



It is not necessary to select all values at once. If values aren't next to each other or you prefer a different sequence in your series, you select the values one by one. In our example we start over by clearing the Series with the ✕ button. Then we select one key figure and add it with the ⊕ button:

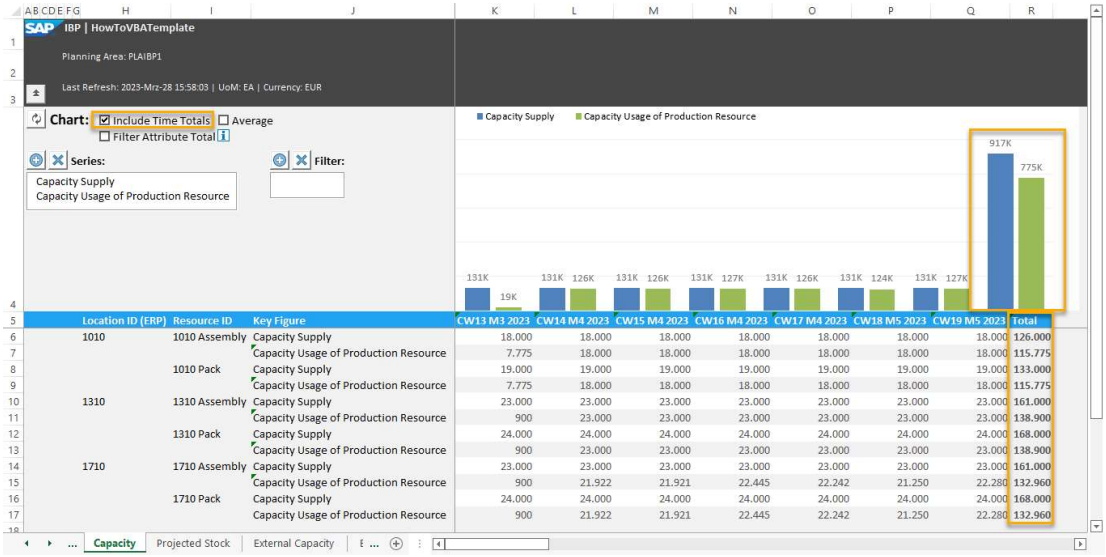


Then you add the second key figure in the same way. Afterwards the sequence in the series is reversed compared to the sequence in the axis

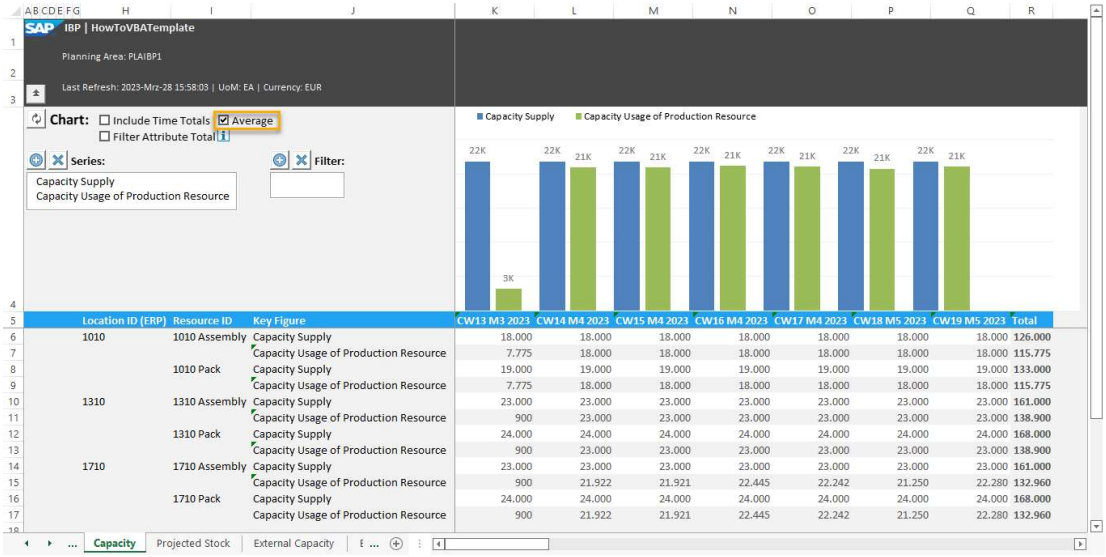


Important to know:

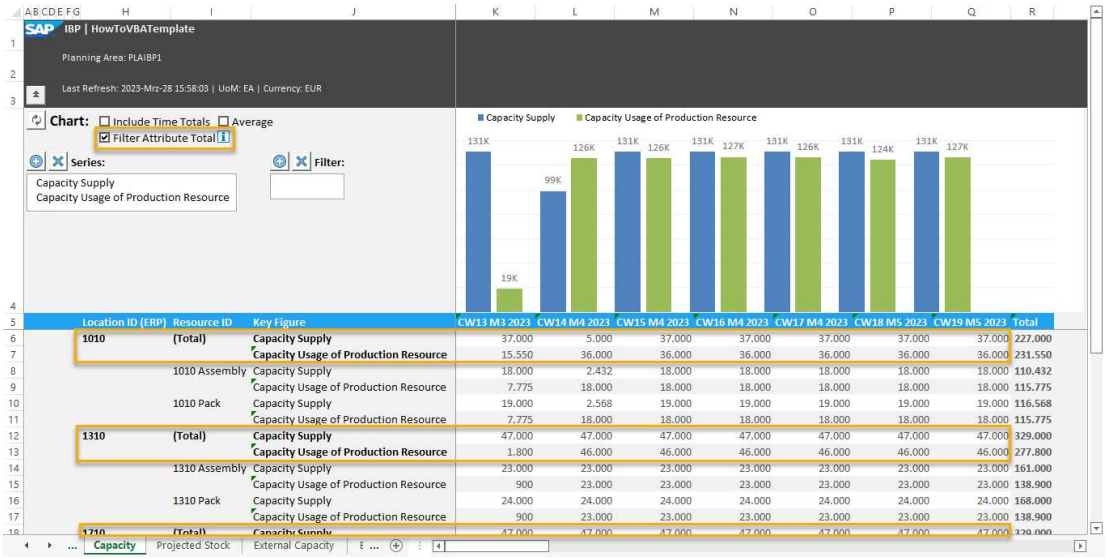
- The values you plot by using ***Series*** must be from the same column in the row axis.
- If you add a value to either box that conflicts with previous content, the new value will overwrite the existing value or values.
- The sequence for adding filters is irrelevant.
- By default, the chart plots the sum of values that fit to the selection criteria.
- If you select the ***Include Time Totals*** checkbox, your chart will plot time-based totals as well as the other selected values. Attribute-based totals will not be considered in this case.



- If you select the ***Average*** checkbox, the chart plots the averages of the values that fit to the selection criteria.



- If you select ***Filter Attribute Total***, the chart plots the highest aggregated level defined by the chart filters by attribute totals. This function can be helpful for example in cases where values are defined as percentages and should be shown in the graph correctly.



- The ***Filter Attribute Total*** behavior is the following:
  - In case there are no filters defined in the ***Filter*** list, then the chart shows the grand total for the selected series (the row is being considered in which you can see the most TOTAL attributes)
  - In case there is one or more filters in the ***Filter*** list, then those rows are being considered that are filtered by the filters AND where the TOTAL attributes count is the maximum among the filtered rows (a grand total defined by the filters).
- You can remove all items from the ***Series*** by clicking the ✕ button.
- You can remove individual items from the ***Filter*** by selecting the items you want to remove before clicking the ✕ button. Selecting none or all items clears the control completely.

The chart updates automatically after you choose ***Save Data***, ***Simulate***, ***Refresh***, ***Edit View***, and other IBP functions that cause the planning view to render. It also updates after changes to ***Series*** and ***Filter*** wherever possible. It does not react immediately to manual edits, like a chart based on formulas – in this case you need to refresh the chart manually by pressing the button.

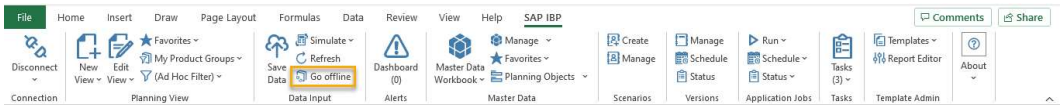
Caution: *In case you are using the option* ***Repeat Row Headers*** *in combination with* ***Use Microsoft Excel cell references,*** *adding Series and Filters won't be working for the reference cells as they are not recognized as IBP Formulas.*
Deselect the ***Use Microsoft Excel cell references*** *in this case in the* ***About*** *group of the* ***SAP IBP*** *ribbon under* ***Settings*** *>* ***Options…*** *in the* ***Display*** *tab.*

Sharing SAP IBP Planning Views with non-SAP IBP Users (Working in Offline Mode)

The ***Go offline*** function in the ***Data Input*** group of the ***SAP IBP*** ribbon can be used to share SAP IBP planning views with non-SAP IBP users. When choosing ***Go offline***, all existing SAP IBP formulas are converted into readable Excel formulas and only the data area is changeable, the rest is protected and cannot be changed.

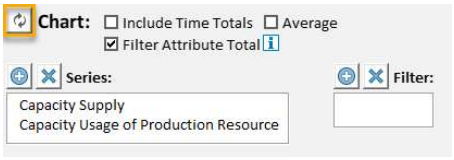
The VBA templates support working in offline mode. The end user experience according to the VBA chart, like setting ***Series*** and ***Filters***, should remain the same independent whether you are working online or offline.

Let's assume you would like to share a planning view with a user who does not have access to your ***SAP IBP*** system. You open your template or favorite, refresh the planning view(s) you would like to share, and press ***Go offline***.



When switching to offline mode…

- the system is asking you to refresh all the planning views that haven't been refreshed, which is recommended to have the latest data in the workbook.
- it is recommended to refresh the controls of the VBA Template by clicking the Refresh button



- it is recommended to unhide the plot area (chart area including the VBA controls). The plot area in offline mode is protected like all other areas outside of the data area and cannot be hidden or unhidden by the user when the workbook is used in offline mode.

Once your workbook is in offline mode you can save it as Excel file (.xlsm) locally and share it with any other user. An offline non-IBP user can change the chart ***Series*** and ***Filters*** and adjust the data (change key figure values). Once the changes are done, the non-IBP user saves the Excel Workbook and sends it back. Afterwards it's possible to review the changed data and simulate and/or save them when being logged in to the ***SAP IBP*** system.

Decide which version you want to use

There are two versions of the SAP IBP VBA sample template available:

Embedded (ChartVBA_Embedded.xlsm)

With this version, the needed VBA code to control the chart is embedded in the template itself.

The advantage is that the end user does not need to add an additional add-in to be able to use the VBA templates.

The downside is that the underlying VBA code is replicated and saved with every template and favorite, which makes it difficult to correct or improve any functions. Therefore, you would have to correct all templates and ask your end users to recreate their favorites from the corrected templates.

To use the embedded template, extract ***ChartVBA_Embedded.xlsm*** from the archive delivered with SAP note 1790530 and proceed with the next chapter.

VBA add-in (ChartVBA_AddIn.xlsm + SAP_IBP_Chart.xlam)

With this version, the central VBA code to control the chart resides in the ***SAP_IBP_Chart.xlam*** file, only control events, IBP-specific events, and control manipulation code are located directly in the template.

The downside is that users must add an additional add-in. The advantage is that replacing the ***SAP_IBP_Chart.xlam*** file with a newer version will correct any central code issue for all templates and favorites without the need to fix individual workbooks.

In case you are planning to enhance the VBA code and to include your own custom VBA coding, we recommend using this version of the SAP IBP VBA template.

To enable the ***SAP_IBP_Chart*** add-in, do the following:

1. Extract ***SAP_IBP_Chart.xlam*** from the archive delivered with note 1790530 and save it in the folder **%appdata%\Microsoft\AddIns**.
2. Open Excel and go to File > Options > Add-ins. You should see Sap_Ibp_Chart under Inactive Application Add-ins with type Excel Add-in.
3. Choose **Excel Add-ins** in the ***Manage*** dropdown and then click on ***Go…***
4. Check the checkbox for ***Sap_Ibp_Chart*** to activate it and confirm with OK.

Note: *Dependent on the* ***Macro Settings*** *defined in* ***Microsoft Excel*** *the* ***SAP_IBP_Chart*** *add-in will not load unless it is in a trusted location. Go to* ***File*** *>* ***Options*** *>* ***Trust Center*** *>* ***Trust Center Settings…*** *>* ***Trusted Locations*** *and click on* ***Add new location…****, then on* ***Browse*** *and copy* ***%appdata%\Microsoft\AddIns*** *into the path field. Press OK until you are done.*

Your VBA add-in has been added and is ready to use. To update it, it's sufficient to replace the ***SAP_IBP_Chart.xlam*** file in the folder **%appdata%\Microsoft\AddIns** with a newer version (without changing the name). You don't need to do the steps to activate it again.

Note that all users of the VBA templates will have to add and activate the VBA add-in once.

Now extract ***ChartVBA_AddIn.xlsm*** from the archive delivered with SAP note 1790530 and proceed with the next chapter.

How to create and add a template

To create a template, do the following:

1. Open ChartVBA***_AddIn.xlsm*** or *ChartVBA****_Embedded.xlsm*** (refer to previous chapter to make the choice).
2. Choose the worksheet you prefer – you can delete or hide other worksheets in case they are not required for now.

3. If desired, adjust the controls, chart, header section, and VBA code to your needs (refer to chapter “How to adjust the formats” for more information).
4. If you need more than 10 columns for the row headers, insert more columns at the left of column A and insert additional SOP_Heading formulas (up to =SOP_Heading15) in row 5. Always start with =SOP_Heading1 in the column closest to the data area. The SOP_Heading formulas will be automatically adjusted dependent on the planning view definition. Reduce the column width of the additional columns you have added to a minimum that less space is used.
5. For a multi worksheet template, copy the worksheet as many times as you want.
   Remember:
   - You can only copy the worksheet with the ***Microsoft Excel*** functionality if it has no planning view in it. Deleting extra sheets later is no problem. Once a planning view is added you can copy a worksheet by selecting the option ***Copy Current Sheet*** in the ***New View*** dropdown of the ***Planning View*** group. This way a new worksheet is created that contains the same planning view definition, which then can be adjusted.
   - If you copy an extra worksheet from your “empty template” workbook later, you need to delete any links to the original workbook.
6. Save your version of the empty template.
7. Log on to your ***SAP IBP*** system.
8. Select the cell next to “Start here” and below “Start below” and create a planning view using the function ***Without Template On Current Sheet***… in the ***New View*** dropdown in the ***Planning View*** group and define the planning view to your needs.
9. When the planning view is rendered, follow the instructions in chapter “How to work with the VBA templates” to set meaningful defaults in Series and Filter.
10. Repeat the last three steps for every worksheet.
11. Adjust the formatting by using the ***SAP IBP Formatting Sheet***. Note that you can copy an existing formatting sheet from a previous template – this template comes pre-formatted for flexible key figure editability (see the “Formatting a Template” section and Appendix 1 in this document for more information).
12. Hide the formatting sheet, select the worksheet users should start with, ensure that the worksheets start at a meaningful position, upload your template, and optionally save a versioned copy locally.

Note: *Regardless of the version of the VBA template you opt for, dependent on the* ***Macro Settings*** *defined in* ***Microsoft Excel*** *the VBA templates might not load properly unless the file path for the related workbooks is added as a Trusted Location or the VBA code is signed.*

In the SAP IBP, add-in for Microsoft Excel the file path where planning view workbooks are downloaded to is defined in the ***About*** *group under* ***Settings*** *>* ***User Settings*** *>* ***File Path for Workbooks****, see SAP Help at Setting Up File Paths for Workbooks for further details.*
Copy the file path and in Excel go to ***File*** *>* ***Options*** *>* ***Trust Center*** *>* ***Trust Center Settings…*** *>* ***Trusted Locations*** *and click on* ***Add new location…****, then on* ***Browse*** *and paste it into the path field. Press OK until you are done.*

Do not place the file ***SAP_IBP_Chart.xlam*** *in the same folder mentioned above.*
If you have administrative rights for your PC, an option is to put the file SAP_IBP_Chart.xlam to the default trusted location for Excel. The path is C:\Program Files (x86)\Microsoft Office\root\<office version>\Library.

To be able to upload templates with VBA code that you have signed to your ***SAP IBP*** *system, you need to first create the template with all its planning view definitions and then sign the VBA code just before uploading it. The signature gets removed when* ***Create New*** *>* ***Without Template On Current Sheet…*** *is used.*

How to adjust the formats

Feel free to adjust the templates to your needs, below you can find some more information about how the template can be adjusted and what needs to be considered.

Adjusting the Controls

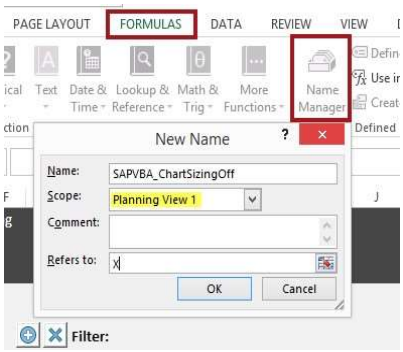
To render properly the active-x controls, an autofit needs to be done for the columns that show the chart controls. To also have the possibility to avoid the autofit, the template takes the option ***Autofit column width*** as an indicator. You can find this option in the ***About*** group of the ***SAP IBP*** ribbon under ***Settings*** > ***Options***… in the ***Display*** tab.

If you do not want the autofit to be used for every column but only for the columns that show the chart controls, you can deselect the option, and set the variable ***GetAutoFitColumnWidth*** in the VBA code of the worksheet to false.

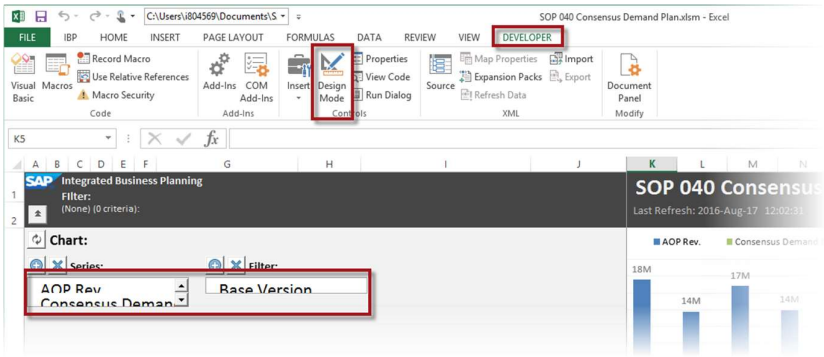
If changes made to a planning view result in changes in its size, the chart adapts to the new width in two ways:

- It adjusts the x-axis of the chart automatically to the new number of columns if you edit the view.
- It resizes the width of the chart's plot area to match the width of the planning view columns after rendering

If this feature is not wanted, it can be switched off by creating Excel Name ***SAPVBA_ChartSizingOff*** as shown in the following screenshot - make sure to set the ***Scope*** to the worksheet that contains the chart. Entering any value in the ***Refers to*** field will switch the sizing off.



A possible issue with ActiveX controls is that some parts of them are not visible if the same workbook is used with different screen resolutions:



This situation can usually be corrected by toggling the ***Design Mode*** on and off (the ***Developer*** tab may need to be switched on to access the button).

Caution: *Please note that changing to* ***Design Mode*** *is not possible when you are working in offline mode due to the Worksheet Protection.*

Hiding the Controls

Finally, the / button will collapse / expands row 4 and hides the chart controls – like row grouping would. Please do not use row grouping to hide this row as the controls can lose their anchors.

Preventing Automatic Color Changes

If you change the time settings for any planning view, then the color of time header will change for the modified time setting. We kept this behavior because of performance reasons. If you want to change this behavior in your planning view, do the following:

1. Open Microsoft Excel and log in to ***SAP IBP***
2. Open or create the respective template
3. Choose ***Settings*** and select ***Options…***
4. Go to the General tab and deselect Keep Formulas Static that Reference Planning View Cells
5. Click OK

Per default this option is set. Deselecting this option might affect the performance of your planning view. You need to change this setting for every template and favorite.

Caution: *Please note that hiding the controls is not possible when you are working in offline mode due to the Worksheet Protection.*

Extending the Header

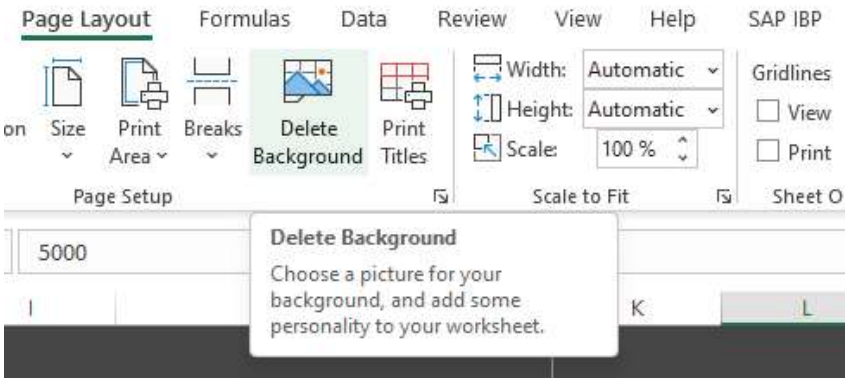
If more space is needed for the header section of the planning view as you might want to display for example key figures and time periods on the column axis, you can insert an extra row below the chart (row 4) and above the blue header line (row 5). You may need to change the formatting to your needs, but the chart and the VBA code doesn't need to be adjusted.

If you want to add additional rows above the chart area additional changes would be needed. The chart controls would need to be rearranged and the VBA code for the / button to collapse and expand the chart area needs to be adjusted as row 4 is hard coded in the ***RemoveControls*** sub.

Caution: *Please note that inserting rows is not possible when you are working in offline mode due to the Worksheet Protection.*

Changing the Background

If you want a different background for the template, you can first delete the grey background in the ***Page Layout*** tab by clicking ***Delete Background*** and then insert a new background.



Caution: *Please note that changing the Background is not possible when you are working in offline mode due to the Worksheet Protection.*

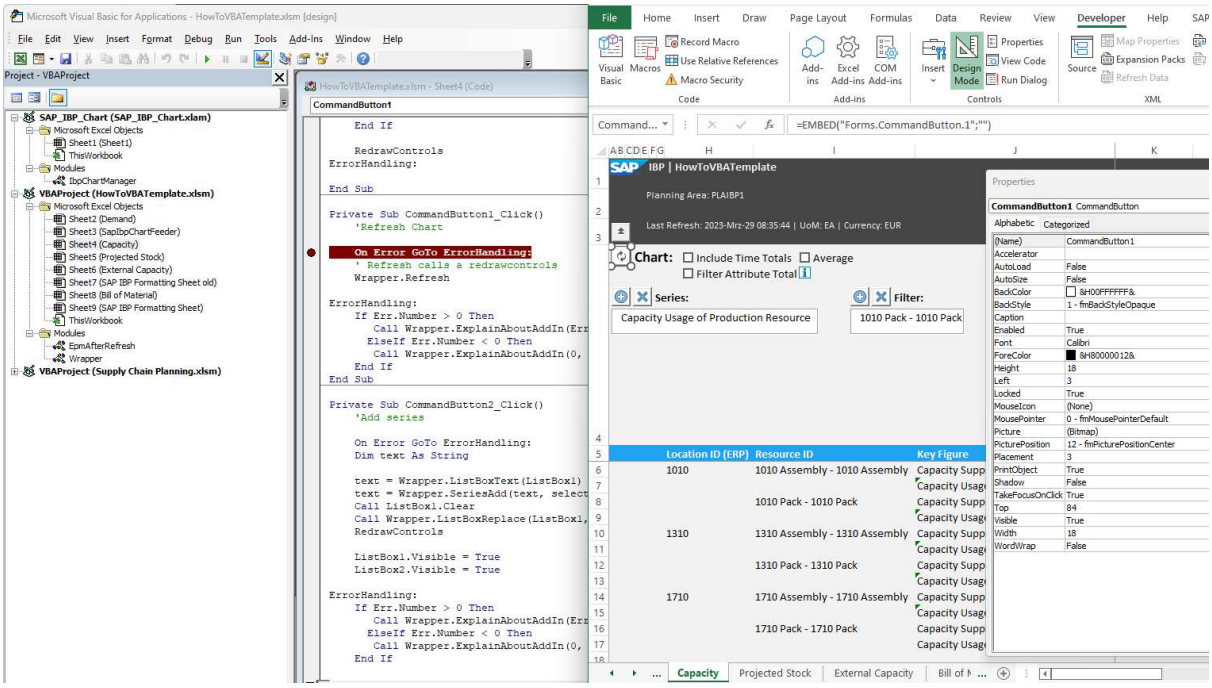
How to adjust the VBA code

This section addresses template administrators who want to change or enhance the sample VBA code provided with the templates.

Note that the intent of delivering sample templates is to jumpstart the template development within the implementation project. The delivered templates cannot eliminate this very important task, they can only make it easier. Therefore, changes or enhancements to the code by the implementation team is expected. The code is provided “as is” and support may be extended as a courtesy until the code is stable. Any changed code is solely your responsibility.

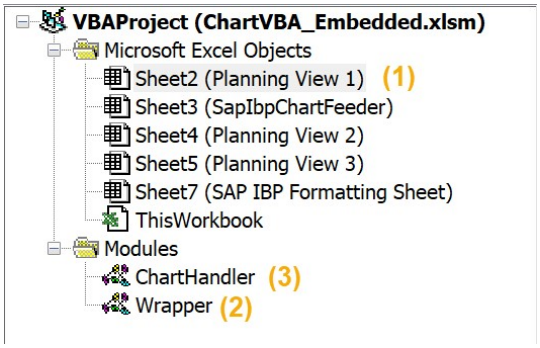
To view the code, select the ***Developer*** tab in ***Microsoft Excel*** and press the button ***Visual Basic*** (you may have to customize your ribbon to make the ***Developer*** tab visible).

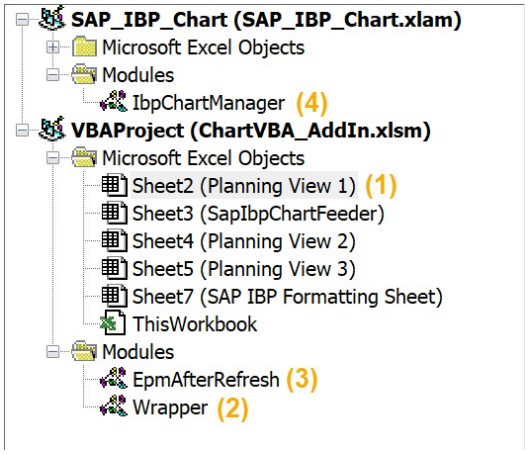
To adjust the controls, choose ***Design Mode*** in the ***Controls*** group of the ***Developer*** tab, then select the control you want to adjust. Click ***Properties*** in the ***Controls*** group of the ***Developer*** tab to change names or captions.



Included VBA Code

Depending on whether you are using the embedded version or the VBA add-in, you will find VBA code in three or four places:





- In the worksheet with the chart and the planning view (1):
  Both template versions (Embedded and VBA add-in) have a slim code layer behind the Excel worksheet. This layer provides different kinds of procedures:
  - Worksheet_Activate()
    Standard Microsoft VBA event, that occurs when a workbook or worksheet is activated. It is calling the RedrawControls procedure.
  - Checkbox<n>_Click()
    These subroutines handle the user's checkbox click. The relevant procedures are called and RedrawControls is triggered.
  - CommandButton<n>_Click()
    These subroutines handle the user's button click for refreshing the chart, adding and deleting ***Series***, adding and deleting ***Filter*** and collapsing and expanding the chart area. The relevant procedures are called and RedrawControls is triggered if needed.
  - RedrawControls() and RedrawControlsFast()
    This subroutine resizes and repositions all controls except collapse and expand to repair any movement or distortion and to adjust to the width of the text box content. The central code also calls back into this method (if you rename it, adjust those calls, too).

Note: *If you make any adjustments to these controls, you may have to change the corresponding code. The worksheet-dependent code will be copied together with the worksheet if you copy one of the worksheets within the VBA template by using the copy function of* ***Microsoft Excel****.*

- In the ***Wrapper*** module (2):
  This module contains the different procedures that are called by the procedures in the worksheet (1) and are calling the central code procedures located in either the ***ChartHandler*** or the ***IbpChartManager*** module depending on the version of the VBA template.
- In the ***ChartHandler*** module / ***EpmAfterRefresh*** module (3):
  - In the embedded version of the template, this module contains the VBA hook for AFTER_REFRESH() and all central code.
  - In the VBA add-in version of the template the module ***EPMAfterRefresh*** contains only the VBA hook ***AFTER_REFRESH()***. The central code is in the module ***IbpChartManager*** of the ***SAP_IBP_Chart*** (4).

  In the VBA templates, the SAP IBP hook (VBA event) ***AFTER_REFRESH*** is used to automatically update the chart.

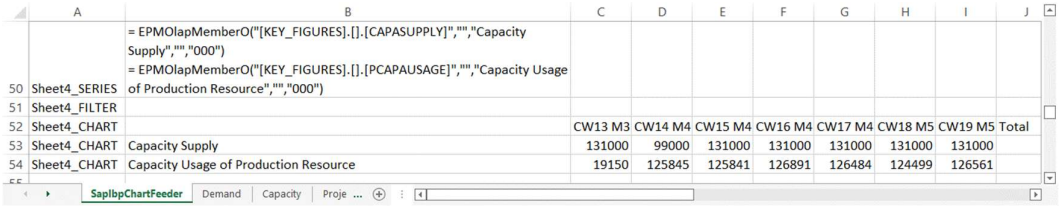
Note: *AFTER_REFRESH() is a function called by the Excel add-in after having refreshed the data and rewritten the planning view, no matter how it is triggered (refresh, simulate, save data, new view, edit view, open favorite/template, changed options, …), see SAP Help at AFTER_REFRESH.*

- In the ***SAP_IBP_Chart***, module ***IbpChartManager*** (4):
  - This is the VBA add-in. It is an xlam add-in, which is basically an empty workbook with VBA code and a special extension (.xlam).
  - The ***IbpChartManager*** module is almost identical to ***ChartHandler*** in the embedded version. The only difference is the missing ***AFTER_REFRESH()***, which needs to be in the active workbook (module ***EPMAfterRefresh***) to be handled accordingly by the Excel add-in.

SapIbpChartFeeder

The templates are using a helper worksheet named ***SapIbpChartFeeder***. In case the worksheet is deleted, you will have to re-select your ***Series*** and ***Filter*** - the system will regenerate the worksheet automatically when you do. The ***SapIbpChartFeeder*** serves all worksheets in the workbook with charts included.

It remembers ***Series*** and ***Filter*** selections per worksheet and holds the numbers to plot:

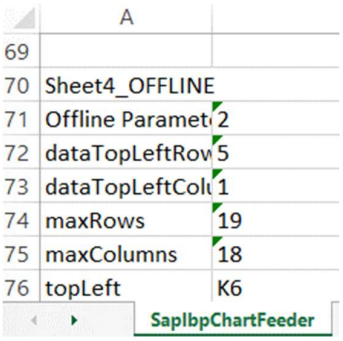


	A	B	C	D	E	F	G	H	I	J
50	Sheet4_SERIES	= EPMOlapMemberO("[KEY_FIGURES].[].[CAPASUPPLY]","","Capacity Supply","","000") = EPMOlapMemberO("[KEY_FIGURES].[].[PCAPAUSAGE]","","Capacity Usage of Production Resource","","000")								
51	Sheet4_FILTER									
52	Sheet4_CHART		CW13 M3	CW14 M4	CW15 M4	CW16 M4	CW17 M4	CW18 M5	CW19 M5	Total
53	Sheet4_CHART	Capacity Supply	131000	99000	131000	131000	131000	131000	131000	
54	Sheet4_CHART	Capacity Usage of Production Resource	19150	125845	125841	126891	126484	124499	126561	

SapIbpChartFeeder | Demand | Capacity | Proje ...

It uses the code name of the worksheet (here Sheet2 is the code name of Planning View 1), so renaming your planning view worksheet doesn't break the link to the worksheets.

The ***SapIbpChartFeeder*** also remembers the data needed for the Offline Mode. Initially the data is not available in the template. As soon as a planning view is created in the template, the offline data is being created. The data is included as extension to the existing ***Series*** and ***Filter*** selections using the code name of the worksheet as identification. They are only used in case of offline mode. When switching to offline mode or switching back to online mode, the existing ***Series*** and ***Filter*** selections are adjusted with the formulas used in the corresponding mode.



	A	
69		
70	Sheet4_OFFLINE	
71	Offline Paramet	2
72	dataTopLeftRow	5
73	dataTopLeftCol	1
74	maxRows	19
75	maxColumns	18
76	topLeft	K6

SapIbpChartFeeder

The VBA code in ***ChartHandler*** or ***IbpChartManager*** links the chart to its data, so extending the series or the width of the planning view does not require manual intervention.

ChartHandler and ***IbpChartManager*** also recognizes the position of the planning view, so adding additional attributes and therefore columns does not require any manual intervention.

If the ***Series*** labels in the chart's legend don't update correctly, right-click the chart, chose ***Select Data…*** and extend the selection to cover column B, where the system puts the series labels. Doing this once should fix the issue permanently.

APPENDIX: SAP IBP FORMATTING SHEET

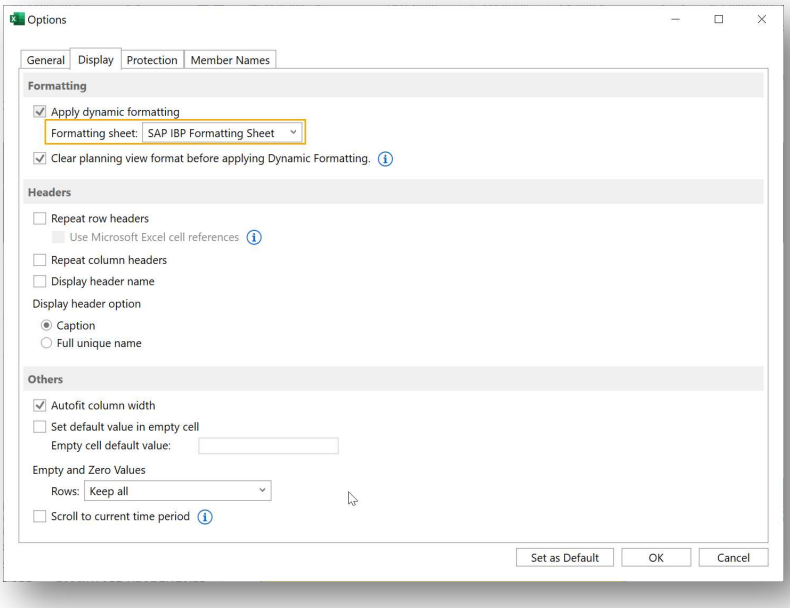
To format your planning view, dependent on the planning view definition (time periods, attributes, key figures, etc.) the ***SAP IBP Formatting Sheet*** can be used. For more information about the ***SAP IBP Formatting Sheet*** and how to use it please see SAP Help at SAP IBP Formatting Sheet.

In the sample VBA templates provided, an ***SAP IBP Formatting Sheet*** is included with some basic formatting settings. Feel free to check the settings and adjust them to your needs. In the sections below you will find further details about the settings that have been added.

If you want to use the ***SAP IBP Formatting Sheet*** provided with the sample VBA templates in one of your existing planning view templates, you can do so.

Please follow the steps below:

1. Delete the existing worksheet with the ***SAP IBP Formatting Sheet*** from your planning view template if you do not want to use it anymore. (You can also leave it; you just would need to make sure that the names of the worksheets are unique).
2. Copy the worksheet with the ***SAP IBP Formatting Sheet*** from the sample VBA template to your planning view template using the standard ***Move or Copy*** function of ***Microsoft Excel***.
3. Choose ***Settings*** > ***Options*** in the ***About*** group and make sure the new formatting sheet is selected in the ***Formatting sheet*** dropdown in the ***Display*** section for all worksheets.



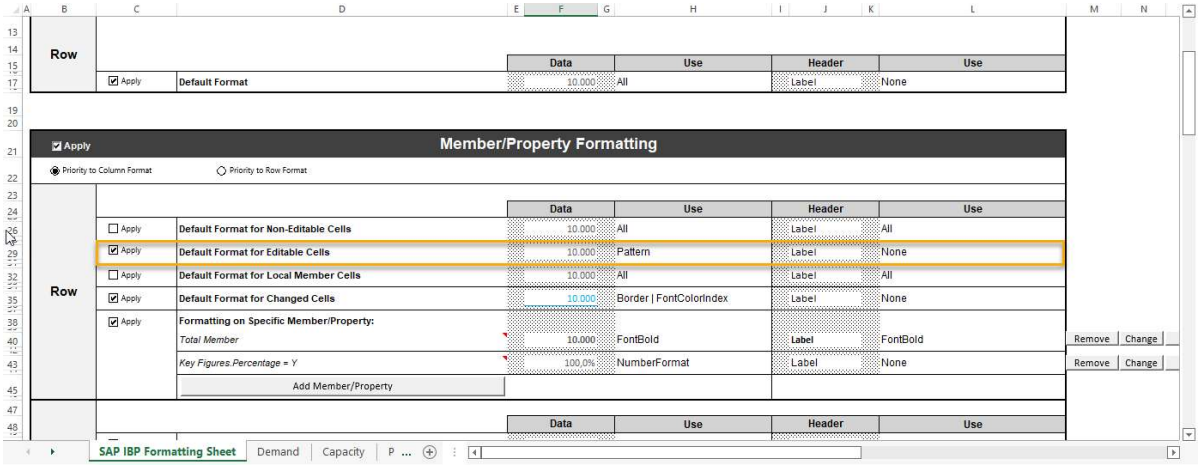
Formatting rule to highlight key figure editability

SAP IBP Formatting Sheets created with Excel add-in version 2105.2.0 and newer offer new formatting settings to visualize the editability of key figures in the planning views, see SAP Help at Formatting Rules to Visualize Key Figure Editability and Editability of Key Figure Values in the Planning View.

You can use the options ***Default Format for Editable Cells*** and ***Default Format for Non-Editable Cells*** to set-up formatting rules how to highlight key figures based on their editability.

Note: *The formatting rules based on the key figure property* ***Editable Indicator*** *that were included in previous versions of the VBA templates are no longer needed and can be deleted. We recommend using the new options to apply formatting to editable or non-editable cells.*

In the ***SAP IBP Formatting Sheet*** that is included in the sample VBA templates those options are already preconfigured.

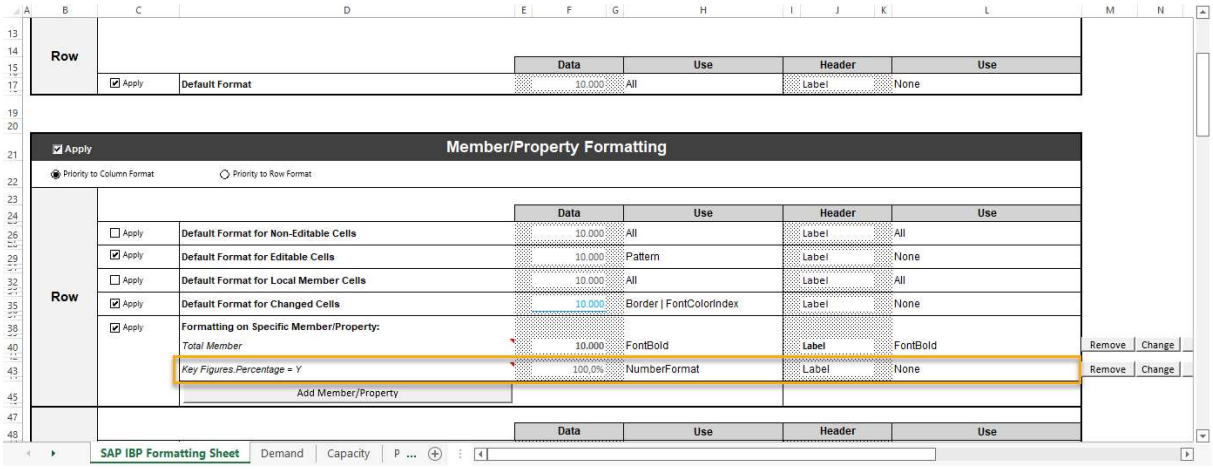


Formatting rule to format percentage key figures

When configuring a key figure, it is possible to define whether a key figure should be displayed as a percentage or not. This can be done in the ***Planning Areas*** app by activating the checkbox ***Display as Percentage*** in the ***Display settings*** of a key figure.

If you want the key figure values in your planning views to be formatted as percentage, you need to add a formatting rule to ***Formatting on Specific Member/Property*** in the ***Row*** box for key figures with the property ***Percentage*** set to ***Y***. See SAP Help at Formatting Key Figure Values as Percentage.

In the ***SAP IBP Formatting Sheet*** that is included in the sample VBA templates such a formatting rule is already included.

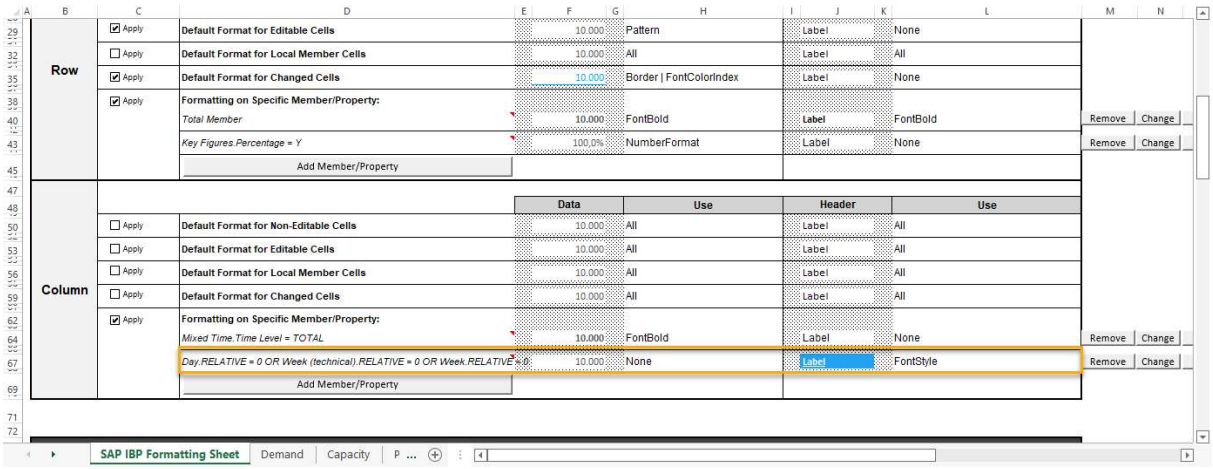


Formatting rule based on the relative property

You can format headers and cells according to their relative time period (as an example for the current month the ***Relative*** property is set to 0; next month is 1…), see SAP Help at Formatting Based on the RELATIVE Property for further details how to set up such a formatting rule.

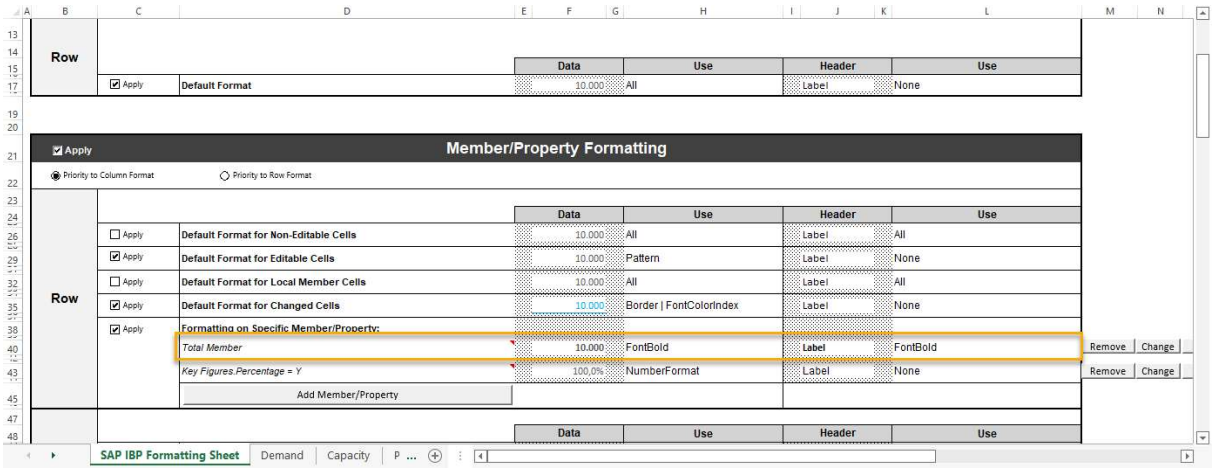
Those kind of formatting rules can be used to highlight the current time period in your planning view or a specific time horizon relative to the actual date. The advantage is that dependent on the actual date the formatting is automatically adjusted.

In the ***SAP IBP Formatting Sheet*** that is included in the sample VBA templates, formatting rules have been set up that format the header of current time period for each time level, independent which one is selected.

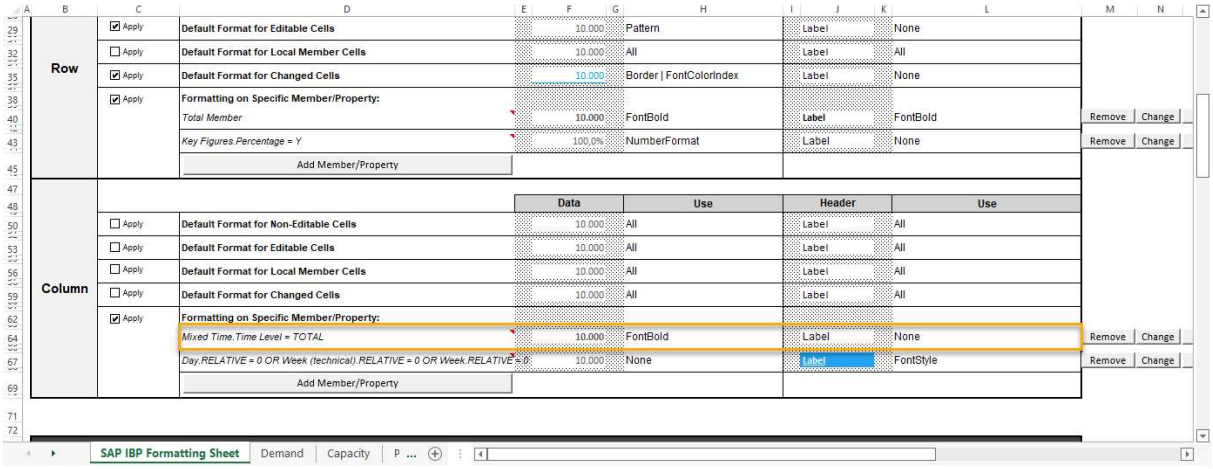


Formatting rules for attribute-based totals and time-based totals

Formatting rule for attribute-based totals inserted in the ***Row*** section:



Formatting rule for time-based totals inserted in the ***Column*** section:



Follow us

www.sap.com/contactsap