SAP IBP, add-in for Microsoft Excel

How to Use the Sample VBA Template in Your Planning Area (SAP Note 1790530)

Version: 2020-06-20

Contents

Introduction	2
Changes compared to previous versions	2
USER GUIDE	3
The Advantage of Using VBA Templates	3
Working with a Template	4
Sharing SAP IBP Planning Views with non-SAP IBP Users (Working in Offline Mode	. 10
Formatting a Template	11
Adjusting the Controls	11
Hiding the Controls	
Extending the Header	12
Changing the Background	
ADMINISTRATOR'S GUIDE	14
Installing a Template	
Creating a Template from ChartVBA*.xlsm	15
Adapting a Template	17
ADDENDIY: EODMATTING VEV EIGHDES THAT ADE EDITARI E IN THE DAST	22



Introduction

Planning View Template development is an important task during an SAP IBP implementation.

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

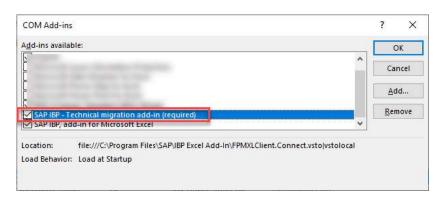
The VBA Samples provided by SAP are samples that can be used to kickstart your implementation of SAP IBP. Please feel free to customize and enhance the VBA code to your liking and towards the needs of your end users.



This version of the VBA samples can only be used in combination with SAP IBP, add-in for Microsoft Excel versions 2005.2.0 and above.

Changes compared to previous versions

- Migration from the older EPM formatting sheet to the IBP formatting sheet.
- Updates to the formatting rules in the IBP formatting sheet (bugfixes).
- Updated the older EPM user defined functions (e.g. EPMUser())with the new IBP user defined functions (e.g. IBPUser()).
- Updated of the VBA code with the ibpAutomationObject so that the template can also run without the SAP IBP technical migration add-in which is currently handling the migration of "old" VBA code (from SAP IBP Excel add-in 2005.2.0 onwards).



- The VBA code and the SAP IBP Chart Feeder were enhanced to support the Offline Mode so that users without an SAP IBP, add-in for Microsoft Excel can also use the chart and controls when in offline mode.
- Some smaller bug fixes in different areas to improve stability.

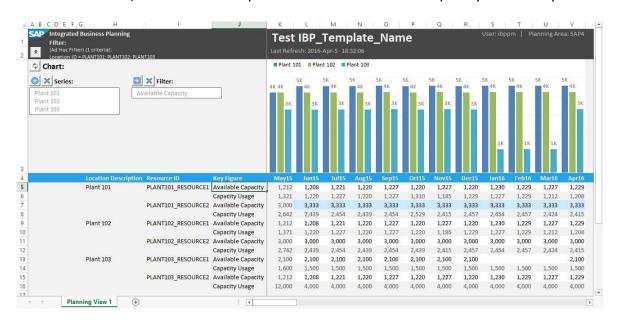


USER GUIDE

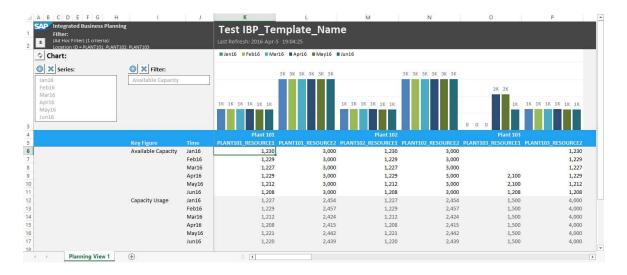
The Advantage of Using VBA Templates

VBA-driven charts are more flexible than our previously published formula-based charts. It can handle non-standard layouts and can plot attributes other than key figures or versions.

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



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



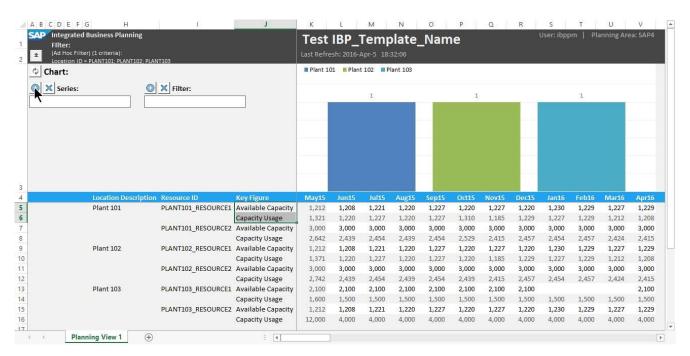




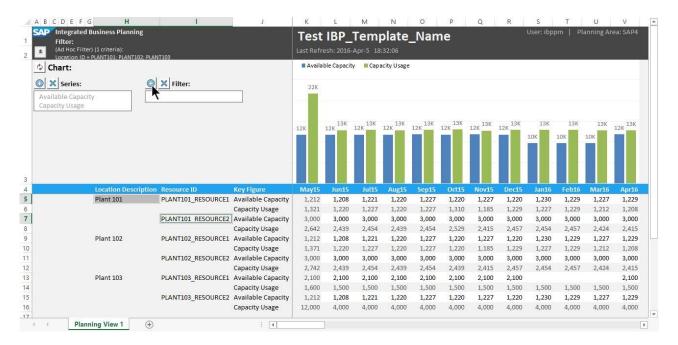
This flexibility makes it easy to add apples to oranges. If, for example, we hadn't filtered by Available Capacity in above chart, the chart would have summed up capacity and usage.

Working with a Template

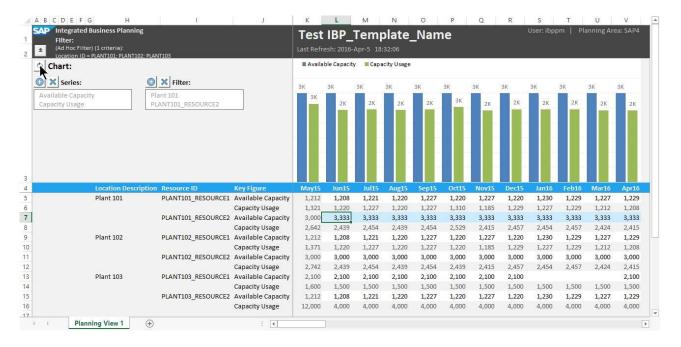
To define your chart's content, select the key figures you want to plot in your row axis and press the 🕒 button next to Series:



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 next to *Filter*:

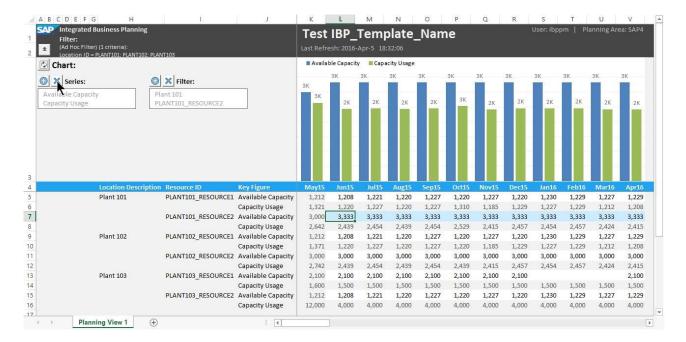


The system will note your selection in the box below and adjust the chart. To refresh the chart after manual edits, press the button next to *Chart*:

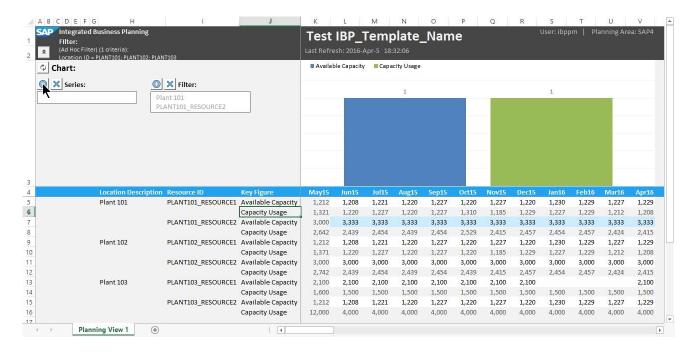




The system will recalculate the chart:

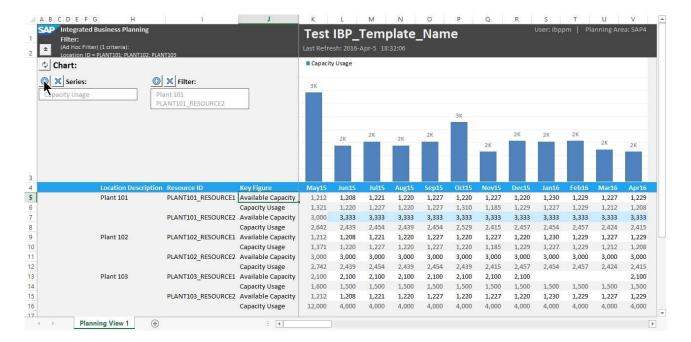


It is not necessary to select all values at once. If values aren't next to each other or you prefer a different sequence in your series, you select the values one by one. In our example we start over by clearing the *Series* with \times . Then we select one key figure and add it with \odot :

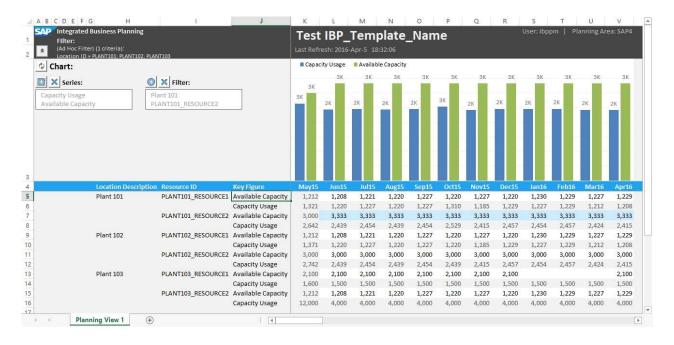




Then you add the second key figure in the same way:



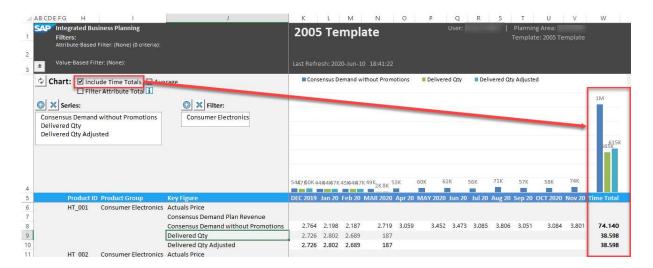
Now the sequence in the series is reversed compared to the sequence in the axis:



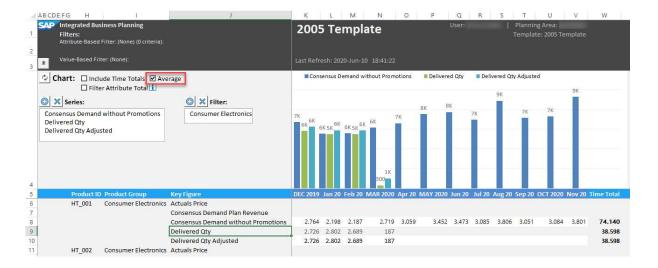


i Notes

- 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.
- When local members are added to series, filters will not work.
- The sequence in Filter 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.

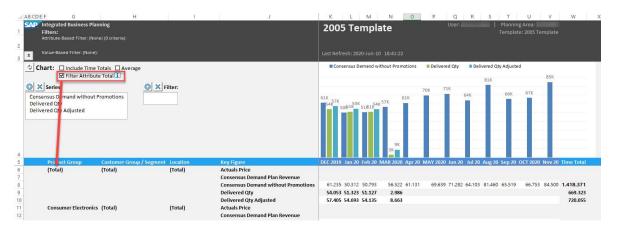


• if you select the *Average* checkbox, the chart plots the averages of the values that fit to the selection criteria.





• 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.



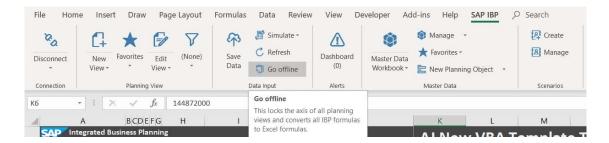
- The "Filter attribute total" behavior is the following:
 - o In case there are no filters defined in the *Filter* list, then the chart shows the grand total for the selected series (the row is being taken into account in which you can see the most TOTAL attributes)
 - o In case there is one or more filters in the *Filter* list, then those rows are being taken into account 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 X.
- You can remove individual items from the Filter by selecting the items you want to remove before clicking X. Selecting none or all items clears the control completely.
- The chart updates automatically after you choose Save Data, Simulate, Refresh,
 Edit View, and other 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 you need to refresh
 it manually.

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

The Offline Mode that is available as of SAP IBP, add-in for Microsoft Excel version 2005.2.0 can be used to share SAP IBP Planning Views with non-SAP IBP Users. To enable this, when using the Offline Mode in the SAP IBP, add-in for Microsoft Excel, 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.

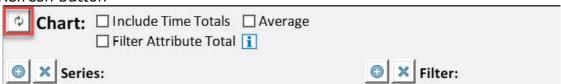
To properly work in offline mode, the VBA Template had to be accordingly redesigned. But the end user experience, like setting series and filters, should remain as before.

Let's assume you would like to share a data report with a user who does not have IBP. You open your report (IBP Favorite/Template), refresh the Planning View (s) you would like to share and 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.
- the system is converting all SAP IBP formulas into readable Excel formulas.
- 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 incl the VBA controls). The plot area in Offline Mode is protected like all other areas outside of the data area and cannot be hidden/unhidden by the user when the workbook is used in offline mode.

Once in Offline Mode you can save your report locally and share it with any other user. In the report the user can still set the chart series, filters and plan data. Once the changes are done, the non-IBP user saves the report and sends it back to the SAP IBP add-in user. Afterwards, the SAP IBP user can then look at the changed data and simulate them in the SAP IBP system. If needed, he can save the data back to SAP IBP.



Formatting a Template

Adjusting the Controls

To render properly the active-x controls, an autofit needs to be done for the columns that show the settings. To also have the possibility to avoid the autofit, the template takes the option "autofit column width" as an indicator. If users do not want the autofit for every column but only for the columns that shows the setting, they can deselect the IBP option, and set the variable <code>GetAutoFitColumnWidth</code> in the VBA coding of the sheet to false.

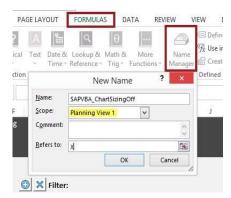
If changes made to a planning view result in changes in its size, the chart adapts to the new width in two ways:

• It adjusts the x-axis of the chart automatically to the new number of columns if you edit the view.

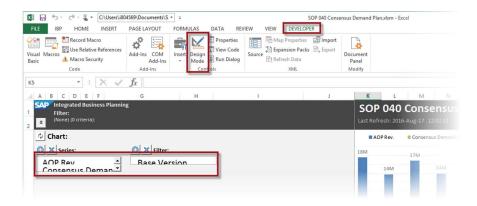
It resizes the width of the chart's plot area to match the width of the planning view

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 in Offline Mode due to the Sheet Protection.

Hiding the Controls

Finally, button 2 / will collapse / expands row 3 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 due to an Excel bug.

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

Deselecting this option might affect the performance of your planning view. You can change this setting for every template.



Caution

Please note that hiding the controls is not possible when in Offline Mode due to the Sheet Protection.

Extending the Header

If more space is needed in the header of the table, an extra row can be inserted there. This action does not cause any issues with the chart.



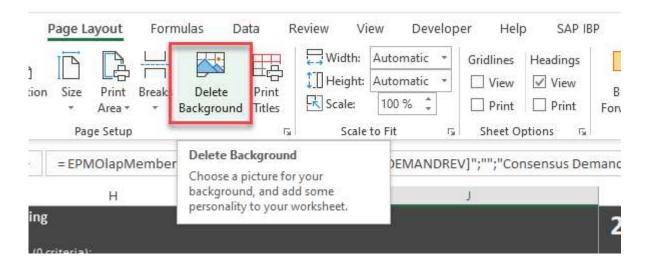
Caution

Please note that inserting rows is not possible when in Offline Mode due to the Sheet Protection.



Changing the Background

If you want a different background for the template, you can first delete the grey background via $Page\ Layout \rightarrow Delete\ Background$ and then insert a new background.





Caution

Please note that changing the Background is not possible when in Offline Mode due to the Sheet Protection.



ADMINISTRATOR'S GUIDE

Installing a Template

The SAP IBP VBA sample template comes in two flavors:

Embedded

With this flavor, all VBA code is embedded in the template itself.

The advantage is that the end user does not need to install an additional add-in to run the templates.

The downside is that the code is replicated in every template and favorite, which makes it impossible to correct a bug in all copies in the code. 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 note 1790530 and proceed with the next chapter.

With VBA add-in (separate .xlam file)

With this flavor, the central code resides in the VBA add-in, only control events, IBP-events, and control manipulation code are located directly in the template.

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

To enable the VBA add-in, do the following:

- 1. Extract SAP_IBP_Chart.xlam from the archive delivered with note 1790530 and place it in %appdata%\Roaming\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 Manage: Excel Add-ins -> Go
- 4. Check the checkbox for Sap_Ibp_Chart and confirm with OK.

i Note

Specific Excel versions will not load the Excel Add-in unless it is in a trusted location. Go to File > Options > Trust Center > Trust Center Settings... > Trusted Locations > Add new location... > Browse and copy

%appdata%\Microsoft\AddIns into the path field. Press *OK* until you are done.



i Note

Regardless of the flavor you opt for, specific Microsoft Excel versions might not load templates properly unless the file path for the related workbook is added as a Trusted Location. To do so, go to File > Options > Trust Center > Trust Center Settings... > Trusted Locations > Add new location... > Browse. Copy

C:\Users\<User Name>\Documents\SAP_IBP (which is by default file path for SAP IBP workbooks) and paste it into the path field. Press OK until you are done.

Do not place the file SAP_IBP_Chart.xlam in the *C:\Users\<User Name>\Documents\SAP_IBP* folder.

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.

If your workbook file path is not at above mentioned location, then you can find it in the *Settings* option under SAP IBP Excel Add-in. For this, open *Microsoft Excel* > *Logon to SAP IBP* > *Go to Settings...* > *File Path for Workbooks* and add the mentioned location as a Trusted Location.

Your VBA add-in is installed and ready to use. To update it, replace it with a newer version. You don't need to re-install it. Note that all users of the VBA templates will have to install the VBA add-in once.

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

Creating a Template from ChartVBA*.xlsm

To create a template, do the following:

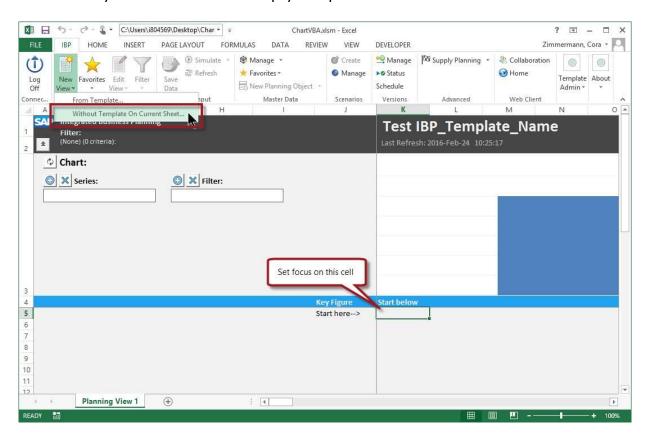
- 1. Open ChartVBA_AddIn.xlsm *or* ChartVBA_Embedded.xlsm (refer to previous chapter to make the choice)
- 2. Choose the worksheet with the chart you prefer you can delete the other one.
- 3. If desired, adjust the controls, chart, header section, and VBA code as you see fit (refer to chapter "A Look Under the Hood" for more information).



Note

The **S&OP 3.0** add-in does not support the SOP_Filter_* fields – to avoid confusion, please delete those fields from A1/A2.

- 2. If you need more than 10 columns for the row headers, insert more columns and adjust the header section accordingly; repair the headers in what is initially row 4.
- 3. For a multi sheet template, copy the sheet as many times as you want. Remember:
 - a. You can only copy the sheet as long as it has no planning view in it deleting extra sheets later is no problem.
 - b. if you copy an extra sheet 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 SAP IBP.
- 8. Focus on the cell next to *Start here* and below *Start below* and create a planning view using *New View > Without Template On Current Sheet...*.



- 9. When the planning view is rendered, follow the instructions in chapter "User Guide: Working with the Chart" to set meaningful defaults in *Series* and *Filter*.
- 10. Repeat the last three steps for every sheet.
- 11.Adjust the formatting. 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, upload your template, and save a versioned copy locally.

Adapting a Template

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

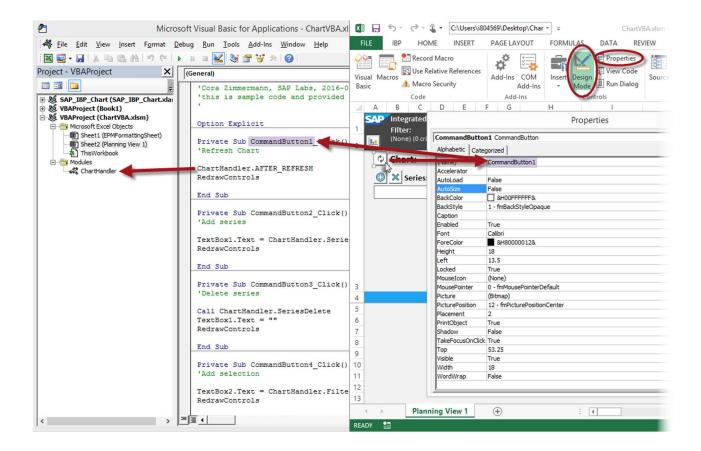
Note that the intent of delivering sample templates is to jumpstart the template development within the implementation project. The delivered templates cannot eliminate this very important task, they can only make it easier. Therefore, changes or enhancements to the code by the implementation team is expected.

Support: 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.

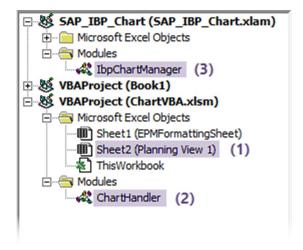
To view the code, use Alt+F11 or $DEVELOPER \rightarrow Visual\ Basic$ (you may have to customize your ribbon to make the DEVELOPER tab visible).

To adjust the controls, use $DEVELOPER \rightarrow Design\ Mode$, then select the control you want to adjust. Click $DEVELOPER \rightarrow Properties$ to change names or captions.





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





(1) In the sheet with the chart and planning view:

Both flavors have a slim code layer behind the Excel worksheet. This layer provides two kinds of services:

CommandButton<n>_Click()

These subroutines handle the user's button click. They call the logic in the central code and fill or clear the corresponding text box. Two routines collapse and expand the chart-line (line 3).

Except for the collapse subroutine, they then call the second service:

RedrawControls()

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).

If you make any adjustments to these controls, you may have to change the corresponding code.

This sheet-dependent code will be copied with the sheet if you create a multisheet template.

(2) In ChartHandler module:

In the embedded flavor of the template, this module contains the VBA-hook for AFTER_REFRESH() and all central code.

In the add-in flavor of the template this module contains only the VBA -hook AFTER REFRESH(). The central code is in the add-in (point 3).

AFTER_REFRESH() is a function called by the SAP IBP add-in's layer after every rendering of a planning view, no matter how it is triggered (refresh, simulate, save data, new view, edit view, open favorite/template, sheet options, ...).

In the VBA templates, the AFTER_REFRESH-hook is used to auto-update the chart.

(3) In SAP_IBP_Chart, module IbpChartManager:

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).

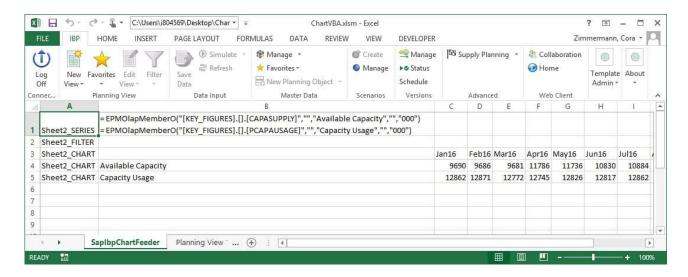


The *IbpChartManager* module is almost identical to *ChartHandler* in the embedded version. The only difference is the missing AFTER_REFRESH(), which needs to be in the active workbook to be found by IBP.

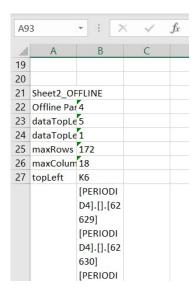
The templates use a helper sheet named *SapIbpChartFeeder*. In case the sheet is deleted, you will have to re-select your *Series* and *Filter* - the system will regenerate the sheet automatically when you do.

SapIbpChartFeeder serves all chart-sheets in the workbook.

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



It uses the code name of the sheet (here *Sheet2* is the code name of *Planning View 1*), so renaming your planning view sheet doesn't break the link between the sheets.



The SapIbpChartFeeder also remembers the data needed for the Offline Mode.

Initially those data are not available in the template. As soon as a planning view is created in the template, the offline data are being created.

They are written as extension to the existing Series and Filter selections using code name of the sheet as identification.

They are only used in case of an offline mode.

Additionally, 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.

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 senses the position of the planning view, so adding columns to accommodate more planning levels 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: FORMATTING KEY FIGURES THAT ARE EDITABLE IN THE PAST

The checkbox *Inputable* (using property *CALC* with values Y/N) works on entire rows and columns. A cell is editable only if all column and row attribute values relevant for the cell are editable.

Assume you have key figures in rows and time periods in columns. Each key figure and time bucket only has one *CALC* flag: Either editable or not editable. Past time buckets are always flagged as "not editable". They cannot be flagged as editable for one key figure and as not editable for a different key figure.

Therefore, flexible key figure editability cannot be formatted this way. SAP IBP provides these two properties:

Property Editable Indicator for dimension Key Figures with these values:

- ALL
- FUTURE
- PAST
- NONE

Property Past Current Future Indicator for time dimensions with these values:

- **P** (past all time dimensions)
- **C** (current lowest granularity)
- **F** (future all time dimensions)
- **PC** (past and current aggregated time dimensions, e.g. O1 in March)
- **PCF** (past, current, and future aggregated time dimensions, e.g. Q1 in February)
- **CF** (current and future aggregated time dimensions, e.g. Q1 in January)

The system will allow editing for these property value combinations:

Editable Indicator (Key Figures)	Past Current Future Indicator (Time)
ALL	(irrelevant)
FUTURE	С
FUTURE	CF
FUTURE	F
PAST	P

To give special formatting to editable cells, the formatting instructions need to match those property combinations for all aggregation levels in your time profile that are relevant for your template.



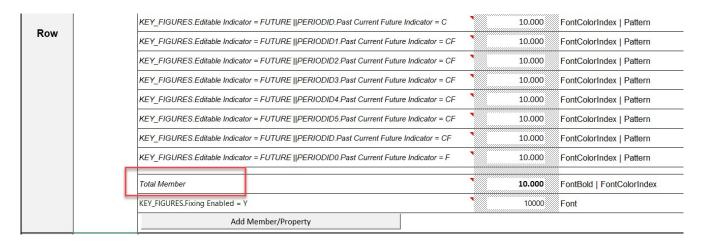
Below are the formatting instructions delivered with this note. They are maintained for a six-tier time profile including PERIODID for flexible time axis. It is for example safe to remove all instructions for PERIODID5 if your time profile only has five tiers:

▼ Apply	Formatting on Specific Member/Property:		
	KEY_FIGURES.Editable Indicator = ALL	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID0.Past Current Future Indicator = C	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID.Past Current Future Indicator = C	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID1.Past Current Future Indicator = CF	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID2.Past Current Future Indicator = CF	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID3.Past Current Future Indicator = CF	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID4.Past Current Future Indicator = CF	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID5.Past Current Future Indicator = CF	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID.Past Current Future Indicator = CF	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODIDO.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID1.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID2.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID3.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID4.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID5.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = FUTURE PERIODID.Past Current Future Indicator = F	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODIDO.Past Current Future Indicator = P	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODID1.Past Current Future Indicator = P	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODID2.Past Current Future Indicator = P	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODID3.Past Current Future Indicator = P	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODID4.Past Current Future Indicator = P	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODID5.Past Current Future Indicator = P	10.000	FontColorIndex Pattern
	KEY_FIGURES.Editable Indicator = PAST PERIODID.Past Current Future Indicator = P	10.000	FontColorIndex Pattern

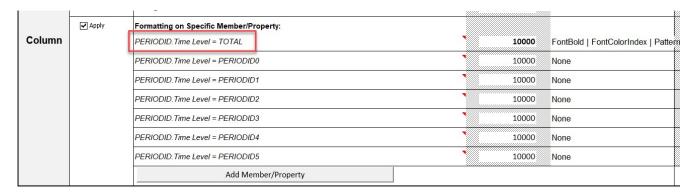
If you need to maintain these instructions yourself, note that depending on the time you do this, not all indicators are sent for all time periods. For example, the CF indicator for a year is only sent in January, since as of February the year already contains a past month and the current year is sent with indicator PCF, all other years either with P or F. At this point in time, you cannot set the CF-instruction for the year.

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

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



Formatting rule for time-based totals was inserted in Column section.



To introduce these rules from this new IBP formatting sheet into an existing template or vice versa, you can copy the formatting sheet into a template workbook. Rename and delete the formatting sheets as needed. Use the Options to make sure all planning view sheets refer to the correct formatting sheet ($About \rightarrow Settings \rightarrow Options... \rightarrow Formatting$):

