Combination-Based Planning with Changing the Status

If planners are in charge of, for example, several product location combinations, they might check and adjust the key figure values for each combination one by one. Therefore, it would be beneficial to have only one combination in the “Planning” worksheet, where the data is reviewed and adjusted. Using the button “Save and Next” it’s possible to save the changes and get the next combination shown in the worksheet directly.

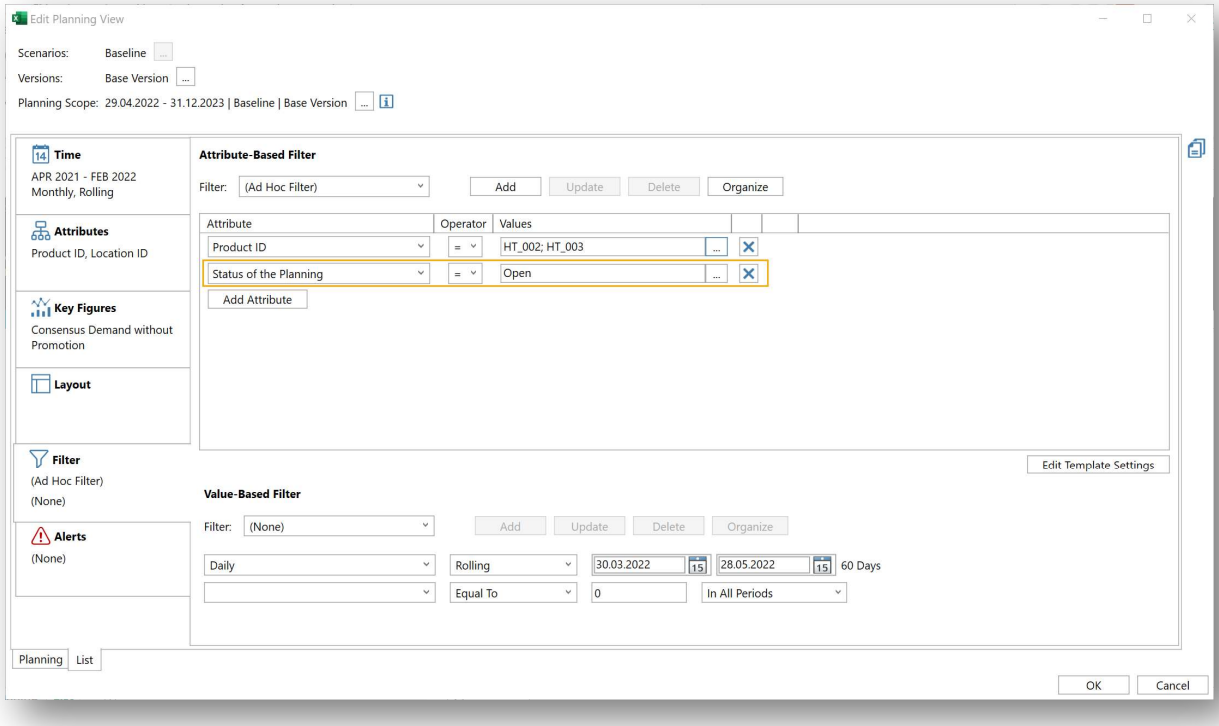
Configure a master data attribute to document the status of the planning process for a combination. Change the value of the attribute ***Status*** of the master data type ***Location Product*** from ***Open*** to ***For Review*** directly in the planning view when going through the combinations one by one, as described in our sample document “Combination-Based Planning” that we provided with release 2202. In this way, other planners can directly see which combinations have already been adjusted/planned and which have not.

1. HOW TO START

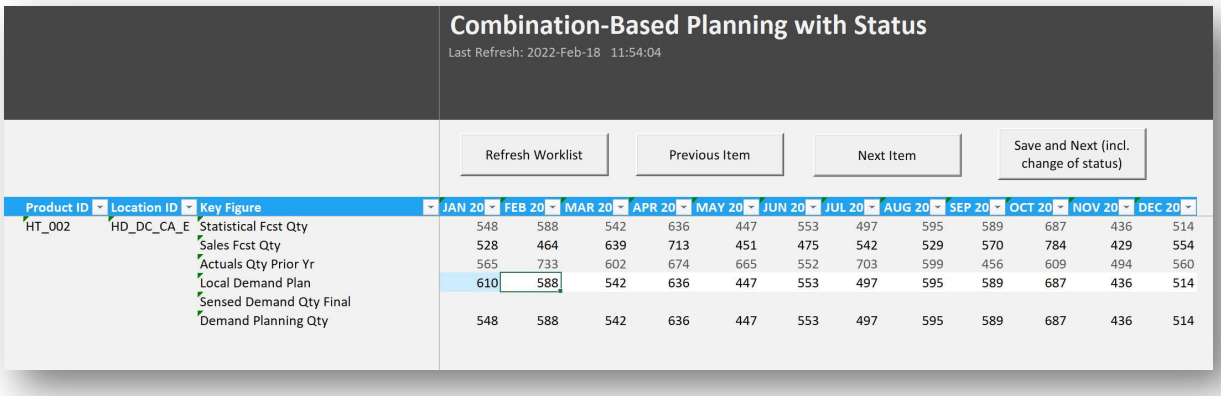
Please make sure to read the tutorial for “Combination-Based Planning” we provided with release 2202 first. In this document, we only describe the additional changes of the sample code to support the status change.

In our example, we created the attribute ***PLANSTATUS*** (“Status of the Planning”) and included it in the master data type ***Location Product*** of the planning area. As a starting point, (start of the planning) set the status for all location product combinations to ***Open***.

To make sure that only combinations with status ***Open*** are displayed in the list of location product combinations which need to be planned, adjust the filter of the planning view you created for the working list (worksheet named “List”):



In the worksheet “Planning” you might adjust the caption of the button ***Save and Next*** to ***Save and Next (incl. change of status)***, so that the user knows that the status is getting changed automatically.



Product ID	Location ID	Key Figure	JAN 20	FEB 20	MAR 20	APR 20	MAY 20	JUN 20	JUL 20	AUG 20	SEP 20	OCT 20	NOV 20	DEC 20
HT_002	HD_DC_CA_E	Statistical Fcst Qty	548	588	542	636	447	553	497	595	589	687	436	514
		Sales Fcst Qty	528	464	639	713	451	475	542	529	570	784	429	554
		Actuals Qty Prior Yr	565	733	602	674	665	552	703	599	456	609	494	560
		Local Demand Plan	610	588	542	636	447	553	497	595	589	687	436	514
		Sensed Demand Qty Final												
		Demand Planning Qty	548	588	542	636	447	553	497	595	589	687	436	514

The code for clicking ***Save and Next*** needs to be adjusted slightly, as marked in yellow:

```
Private IBPAutomationObject As Object
Private position As Range
Private combination() As String

Private Sub SaveAndNext_Click()
    If position Is Nothing Then
        MsgBox "Please first refresh the worklist, to initialize the planning view.", vbOKOnly, "Microsoft Excel:
            Custom VBA code"
        Exit Sub
    End If

    On Error GoTo ErrorHandling:
        If IBPAutomationObject Is Nothing Then Set IBPAutomationObject =
            Application.COMAddIns("IBPXLClient.Connect").Object

        Call IBPAutomationObject.SaveData

        'Change the Status to "For Review"
        Call IBPAutomationObject.UpdateSingleMasterData("Location Product", combination,
            Array("[PLANSTATUS].[].[For Review]"))
        NextItem_Click
        Exit Sub

ErrorHandling:
    'Implement an error handling to help the user to understand what went wrong
    MsgBox Err.Description, vbOKOnly, "Microsoft Excel: Custom VBA code"

End Sub
```

Code explained row by row:

Private combination() As String

To change the value of the master data attribute ***PLANSTATUS***, the product location combination is needed. Therefore, the combination needs to be declared as a global variable outside the function, so that it can be used across the VBA methods that are implemented for the worksheet “Planning”. In the respective sample of 2202, we used the variable “attributes” instead, and declared it locally in the functions needed.

Call IBPAutomationObject.UpdateSingleMasterData("Location Product", combination, Array("[PLANSTATUS].[].[For Review]"))
Call the API UpdateSingleMasterData to adjust the value of ***PLANSTATUS*** to ***For Review*** for the respective location product combination. You need to pass the ID or name of the master data type (which is ***Location Product***), the key attribute values defined by the ***Location ID*** and the ***Product ID***, (those are retrieved in the right format by calling GetAttributeValues in the function “RefreshList_Click”, “PreviousItem_Click” or “NextItem_Click”), and the non-key attribute values that you want to change (Array("[PLANSTATUS].[].[For Review]")).

Please note: Both parameters keyValues and nonKeyValues are from type String[]. Therefore you need to pass a one-dimensional string array, even if only one key attribute value or non-key attribute value is passed. Please make sure to only pass non-key attribute values that are different from the actual saved value, otherwise an error message will be shown. See UpdateSingeMasterData on the SAP Help Portal for further information.

Further comments to the code:

- If you have additional attributes other than ***Product ID*** and ***Location ID*** included in the planning view of the worksheet “List”, the retrieved one-dimensional string array will include further attribute values as well apart from the key attribute values needed. In this case, it cannot be directly used as input parameter keyValues for UpdateSingleMasterData to change the status of the combination.

Follow us

www.sap.com/contactsap