# Interactive Solution for Complex Calculations in DP Planning Book Using Macro BAdIs

## Applies to:

Supply Chain Management 5.0 component SCM-APO-FCS (Demand Planning).
For more information, visit the Supply Chain Management homepage.

## Summary

Perform complex calculations in planning book using Macro BAdIs. The BAdI will have the planning book data to play around.

**Author:** Srinivas Maddineni

**Company:** Intelligroup Inc.,

**Created on:** 11 November 2008

## Author Bio

Srinivas Maddineni has 6 years of IT experience and has worked multiple SAP Implementations. Srini has hands on experience in SAP ABAP on R/3, CRM and APO modules. Srini has a Masters in Information Technology and Management from Periyar University TN India and a Bachelor degree in Business Management from Nagarjuna University AP India and is currently working for Intelligroup Inc., USA.

.

## Table of Contents

Scenario....3
Requirement Example....4
Solution Steps....5
Assign BAdI to Macro:....11
Source Code:....13
Related Content....18
Disclaimer and Liability Notice....19

## Scenario

Safety Stock calculation generally happens in Supply Network Planning. Safety stock needs to be calculated based on Forecast and certain parameters and needs to be interactive. A generic algorithm will be built in demand planning.

**Requirement**

- **Step 1:**

  Build plan (ZBUILDPLN) calculated in the weekly planning book is present here

- **Step 2:**

  Calculate Safety Stock Average Calculation weeks (ZSSAVGWK) (Manual Entry)
  In the below work sheet the value entered is 4. Now the program calculation would be (w1+w2+w3+w4) / 4.
  The result would be (100 + 200 + 150 + 150) / 4 = 150.

- **Step 3:**

  Calculated Safety Stock (ZCALSS), when the user inputs a value in the cell Number of weeks stock required (ZSSWKREQ) then the result would be ZCALSS = ZAVGWKFC x ZSSWKREQ that is 150 x 3 = 300.

- **Step 4:**

  Calculate Total Safety Stock (ZTOTSS), User inputs a value in the cell Fixed safety stock required (ZFIXSS) then calculate ZTOTSS = ZCALSS + ZFIXSS that is 300 + 75 = 375.

- **Step 5:**

  Calculate safety stock quantity (ZSAFSTK), User inputs a value in the cell Safety Stock Allocation weeks (ZSSALLWK) then ZSAFSTK = ZTOTSS / ZSSALLWK that is 375 / 3 = 125

- **Step 6:**

  Start printing the resulting values from the desired Cell. User inputs a value in Safety Stock Start Week (ZNALLSTWK) then the Safety Stock Quantity (ZSAFSTK) should print from the input value week starting from the current week.

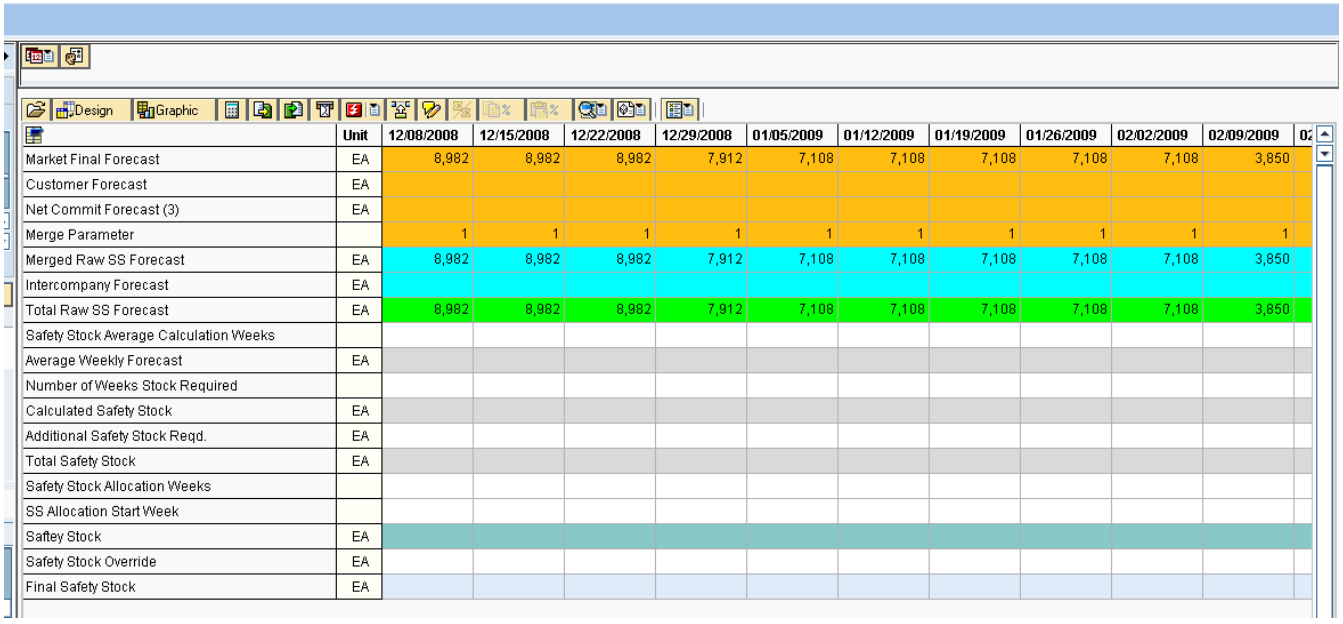
## Requirement Example

| Key Figure | Description | W-4 | W-3 | W-2 | W-1 | W0 | W1 | W2 | W3 | W4 | W…n |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ZBUILDPLN | Build Plan | X | X | X | X | 100 | 200 | 150 | 150 | 500 | X |
| ZSSAVGWK | SS Avg Calculation Weeks | X | X | X | X | 4 | | | | | |
| ZAVGWKFC | Avg Weekly Forecast | X | X | X | X | 150 | | | | | |
| ZSSWKREQ | Number of Weeks Stock Required | X | X | X | X | 2 | | | | | |
| ZCALSS | Calculated Saftey Stock | | | | | 300 | | | | | |
| ZFIXSS | Fixed Safety Stock Reqd. | X | X | X | X | 75 | | | | | |
| ZTOTSS | Total Safety Stock | | | | | 375 | | | | | |
| ZSSALLWK | SS Allocation Weeks | X | X | X | X | 3 | | | | | |
| ZNALLSTWK | Safety Stock Start Week | X | X | X | X | 2 | | | | | |
| ZSAFSTK | Safety Stock Quantity | X | X | X | X | | 125 | 125 | 125 | 0 | X |

Dervived from Monthly forecast
Calculated by program
Updateable columns

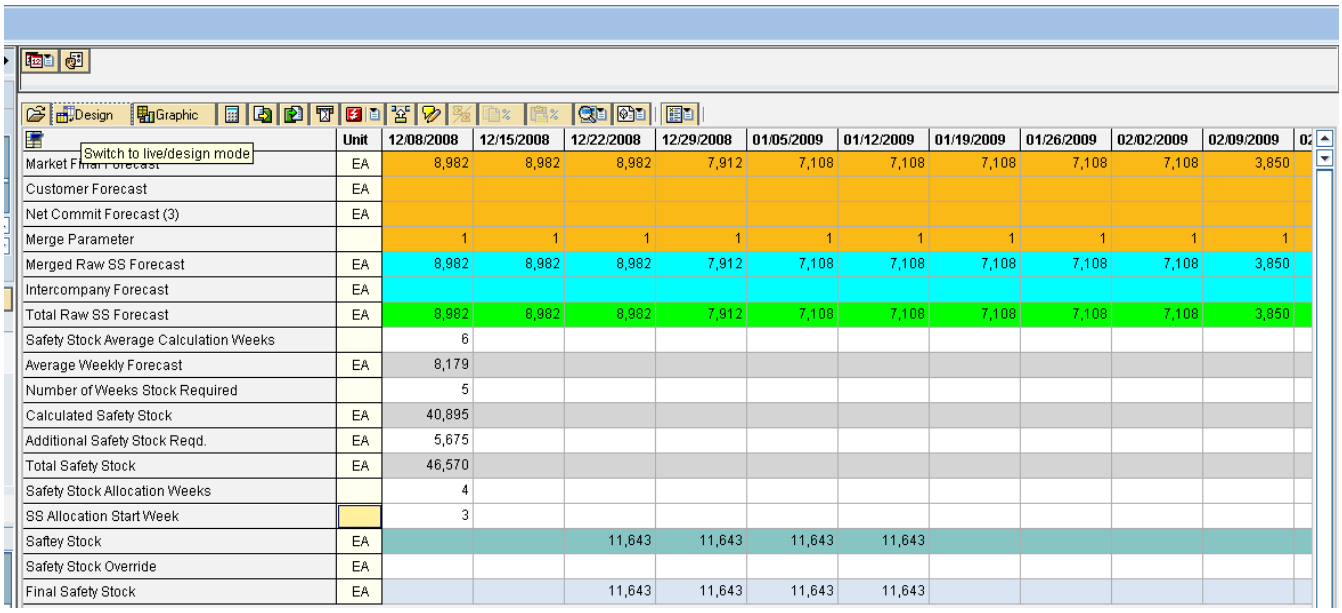
**Planning Book Initial Screen:**

Here the values for Total Raw SS forecast are populated. All the white cells are manual inputs and grey ones are calculated outputs.




| | Unit | 12/08/2008 | 12/15/2008 | 12/22/2008 | 12/29/2008 | 01/05/2009 | 01/12/2009 | 01/19/2009 | 01/26/2009 | 02/02/2009 | 02/09/2009 | 02 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Market Final Forecast | EA | 8,982 | 8,982 | 8,982 | 7,912 | 7,108 | 7,108 | 7,108 | 7,108 | 7,108 | 3,850 | |
| Customer Forecast | EA | | | | | | | | | | | |
| Net Commit Forecast (3) | EA | | | | | | | | | | | |
| Merge Parameter | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| Merged Raw SS Forecast | EA | 8,982 | 8,982 | 8,982 | 7,912 | 7,108 | 7,108 | 7,108 | 7,108 | 7,108 | 3,850 | |
| Intercompany Forecast | EA | | | | | | | | | | | |
| Total Raw SS Forecast | EA | 8,982 | 8,982 | 8,982 | 7,912 | 7,108 | 7,108 | 7,108 | 7,108 | 7,108 | 3,850 | |
| Safety Stock Average Calculation Weeks | | | | | | | | | | | | |
| Average Weekly Forecast | EA | | | | | | | | | | | |
| Number of Weeks Stock Required | | | | | | | | | | | | |
| Calculated Safety Stock | EA | | | | | | | | | | | |
| Additional Safety Stock Reqd. | EA | | | | | | | | | | | |
| Total Safety Stock | EA | | | | | | | | | | | |
| Safety Stock Allocation Weeks | | | | | | | | | | | | |
| SS Allocation Start Week | | | | | | | | | | | | |
| Saftey Stock | EA | | | | | | | | | | | |
| Safety Stock Override | EA | | | | | | | | | | | |
| Final Safety Stock | EA | | | | | | | | | | | |

**The Result Screen:**

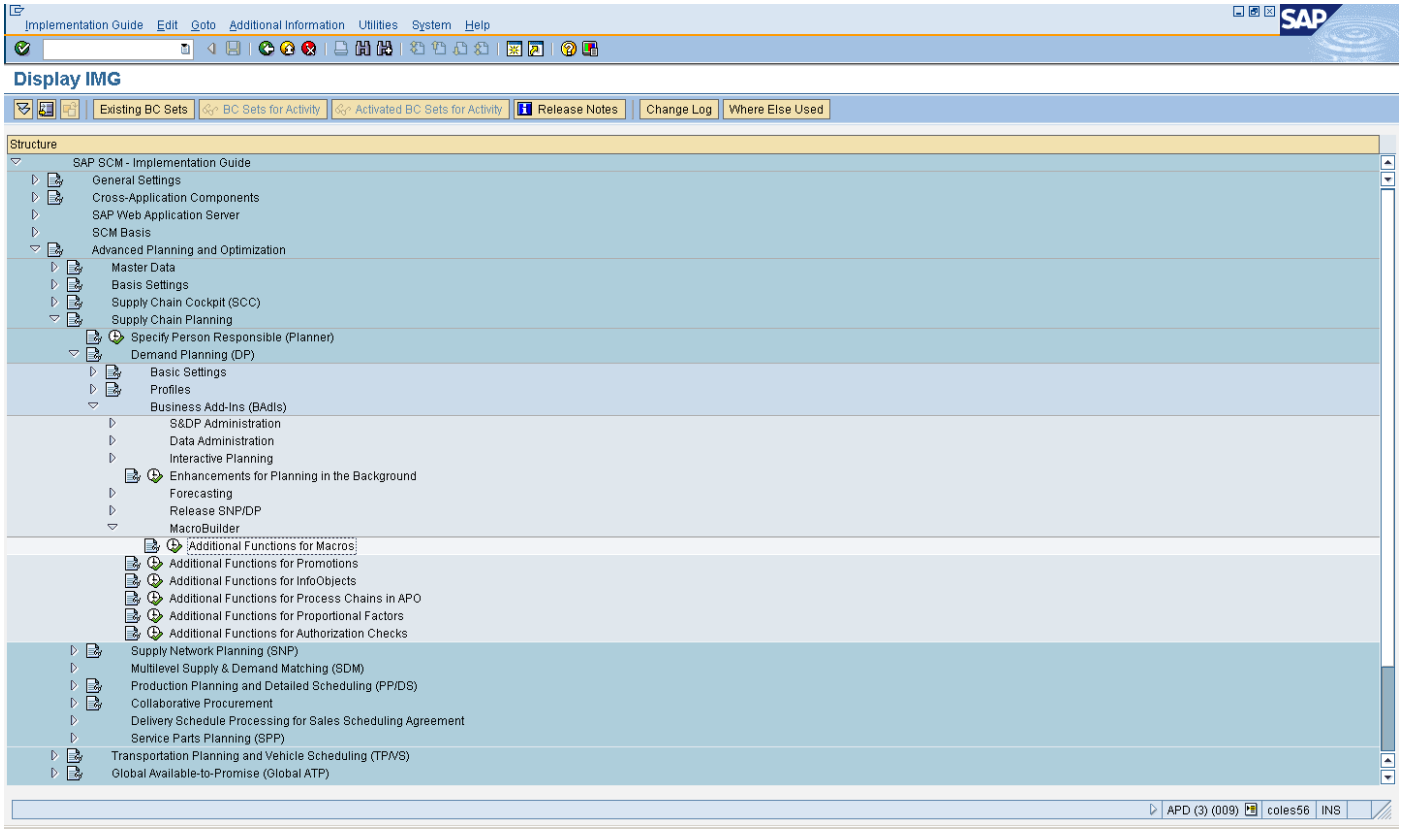



| | Unit | 12/08/2008 | 12/15/2008 | 12/22/2008 | 12/29/2008 | 01/05/2009 | 01/12/2009 | 01/19/2009 | 01/26/2009 | 02/02/2009 | 02/09/2009 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Market Final Forecast | EA | 8,982 | 8,982 | 8,982 | 7,912 | 7,108 | 7,108 | 7,108 | 7,108 | 7,108 | 3,850 |
| Customer Forecast | EA | | | | | | | | | | |
| Net Commit Forecast (3) | EA | | | | | | | | | | |
| Merge Parameter | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Merged Raw SS Forecast | EA | 8,982 | 8,982 | 8,982 | 7,912 | 7,108 | 7,108 | 7,108 | 7,108 | 7,108 | 3,850 |
| Intercompany Forecast | EA | | | | | | | | | | |
| Total Raw SS Forecast | EA | 8,982 | 8,982 | 8,982 | 7,912 | 7,108 | 7,108 | 7,108 | 7,108 | 7,108 | 3,850 |
| Safety Stock Average Calculation Weeks | | 6 | | | | | | | | | |
| Average Weekly Forecast | EA | 8,179 | | | | | | | | | |
| Number of Weeks Stock Required | | 5 | | | | | | | | | |
| Calculated Safety Stock | EA | 40,895 | | | | | | | | | |
| Additional Safety Stock Reqd. | EA | 5,675 | | | | | | | | | |
| Total Safety Stock | EA | 46,570 | | | | | | | | | |
| Safety Stock Allocation Weeks | | 4 | | | | | | | | | |
| SS Allocation Start Week | | 3 | | | | | | | | | |
| Saftey Stock | EA | | | 11,643 | 11,643 | 11,643 | 11,643 | | | | |
| Safety Stock Override | EA | | | | | | | | | | |
| Final Safety Stock | EA | | | 11,643 | 11,643 | 11,643 | 11,643 | | | | |

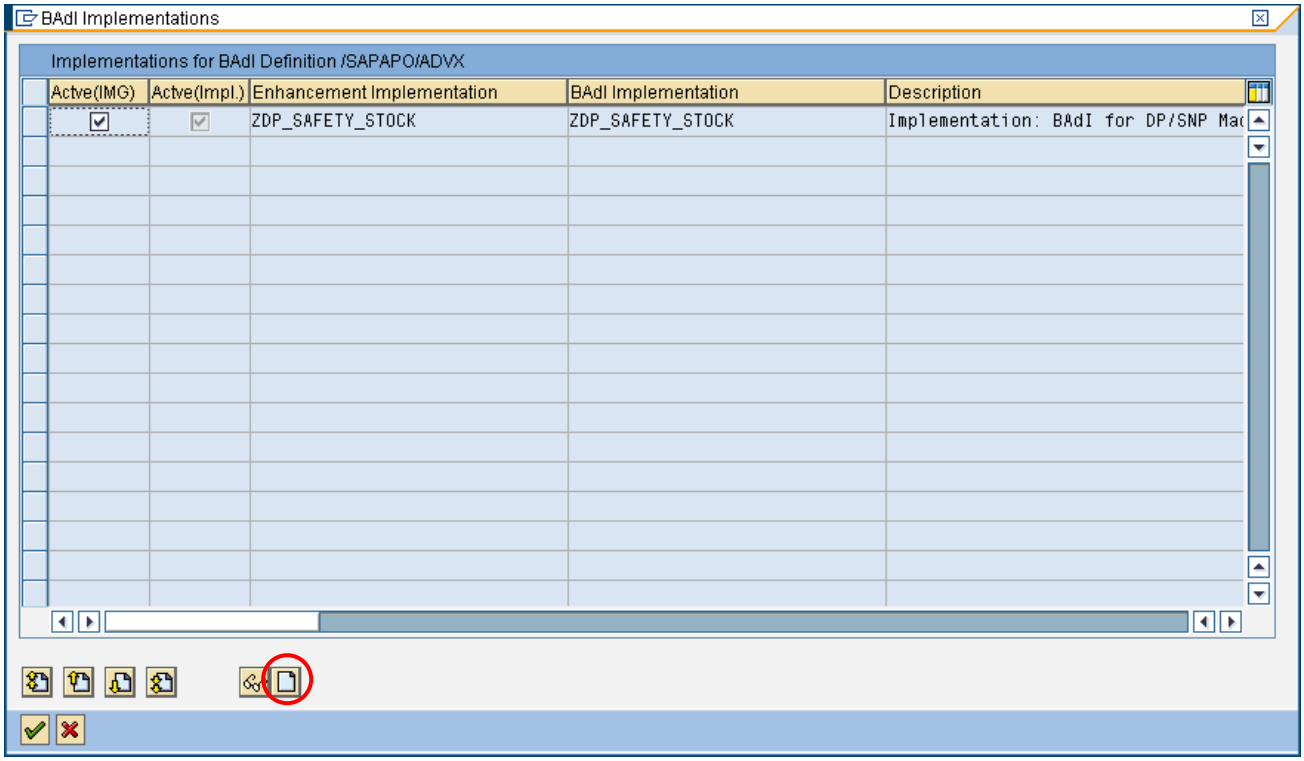
## Solution Steps

The procedure to implement a BADI is:

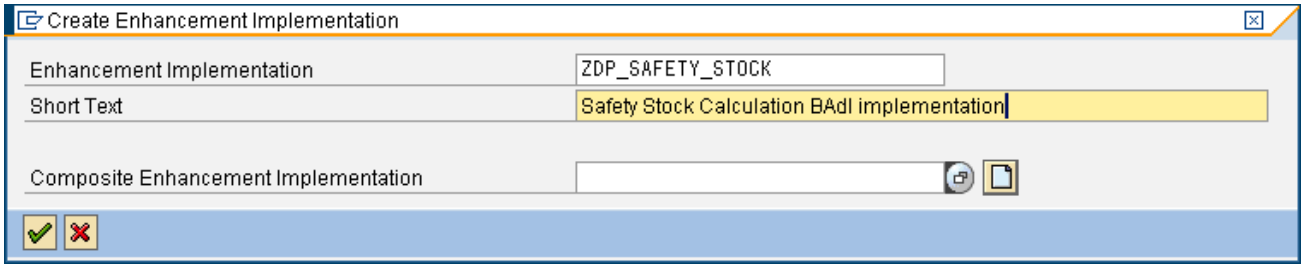
1. SPRO --> SAP SCM - Implementation Guide --> Advanced Planning and Optimization --> Supply Chain Planning --> Demand Planning (DP) --> Business Add-Ins (BAdIs) --> MacroBuilder --> Additional Functions for Macros.



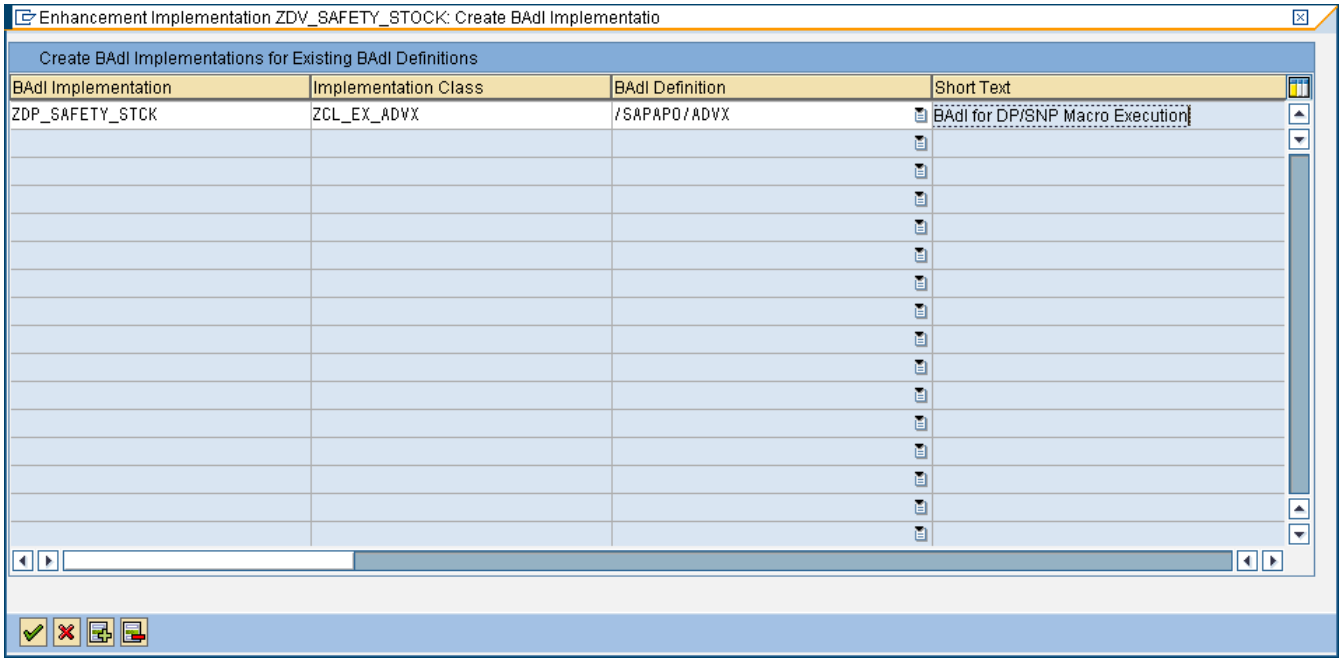
2. Create Implementation button. You will not see a line in the screen if you haven't implemented this BAdI before.



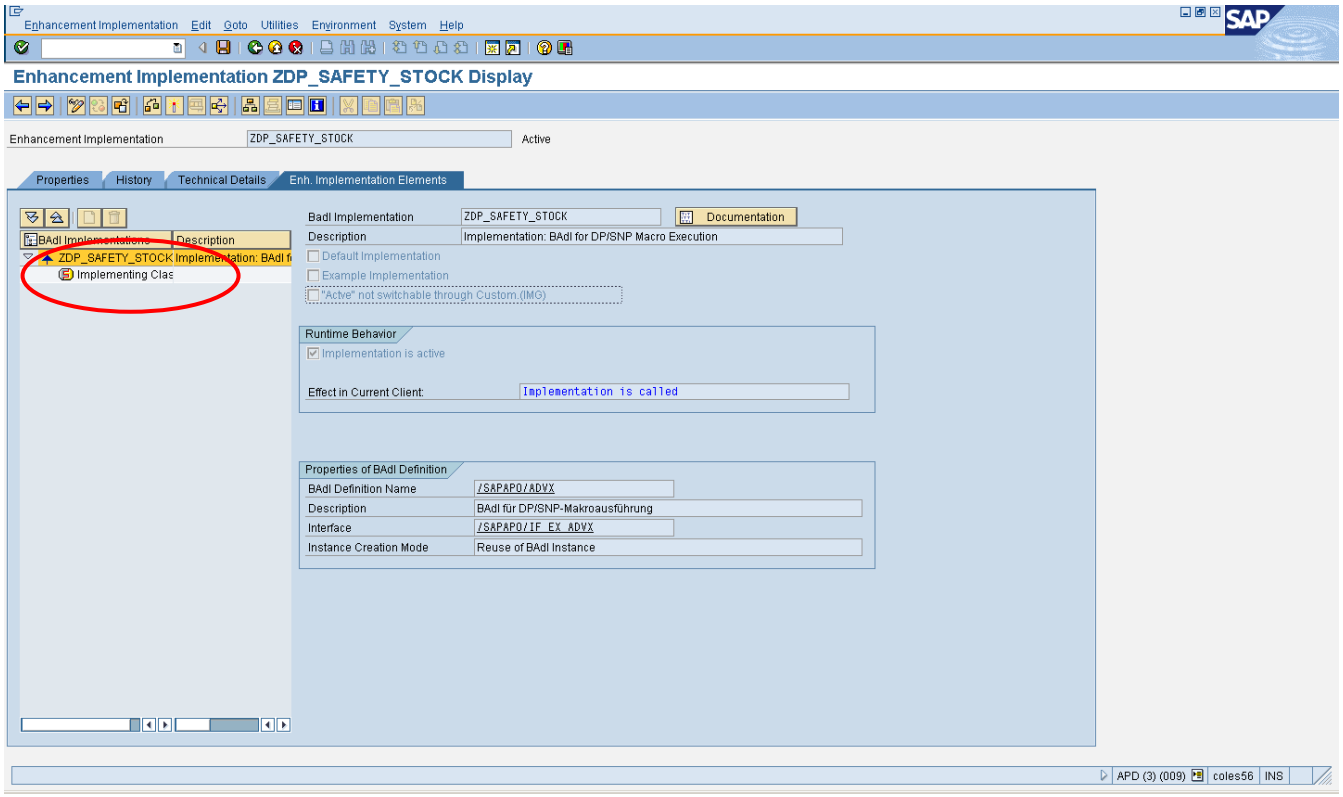

3. Give a name for your implementation, here I mention as ZDP_SAFETY_STOCK, when you press enter key it will prompt you for save. Save the BAdI implementation to appropriate project.



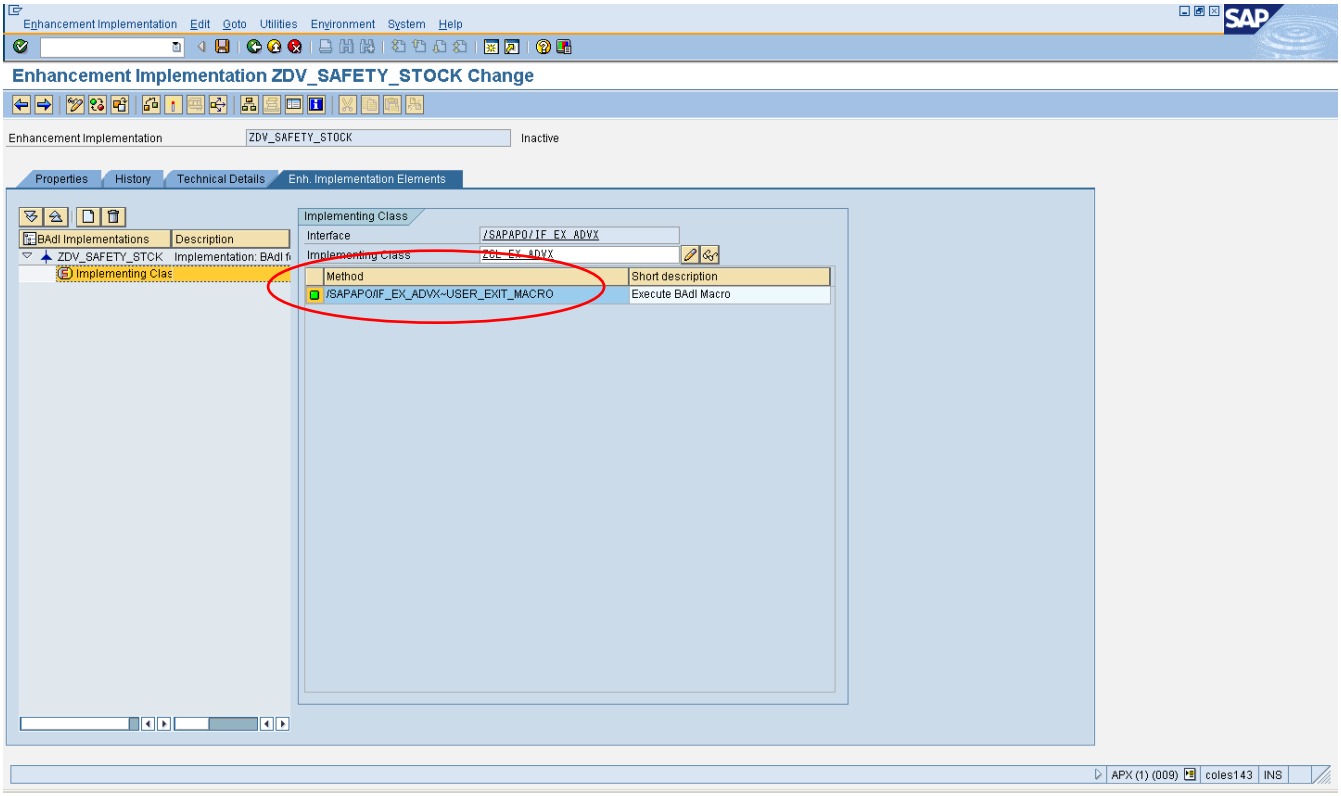
4. Create a new implementation by copying the class for the BADI definition '/SAPAPO/ADVX' and write your own code in the method '/SAPAPO/IF_EX_ADVX~USER_EXIT_MACRO '.



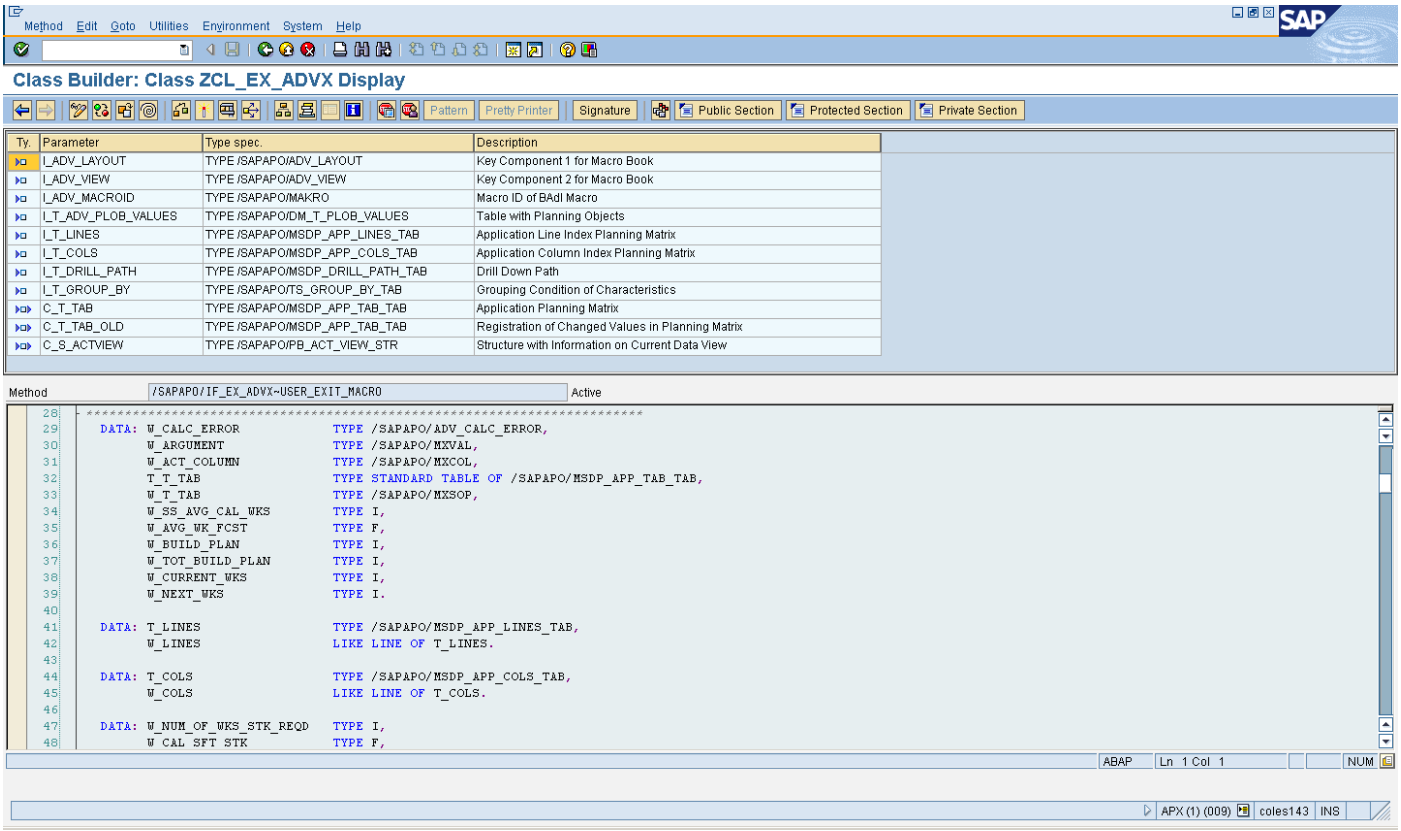

5. You will come to this screen, here drop down implementation name and double click on implementing class



6. You will see a method name; double click on the method to browse through code.

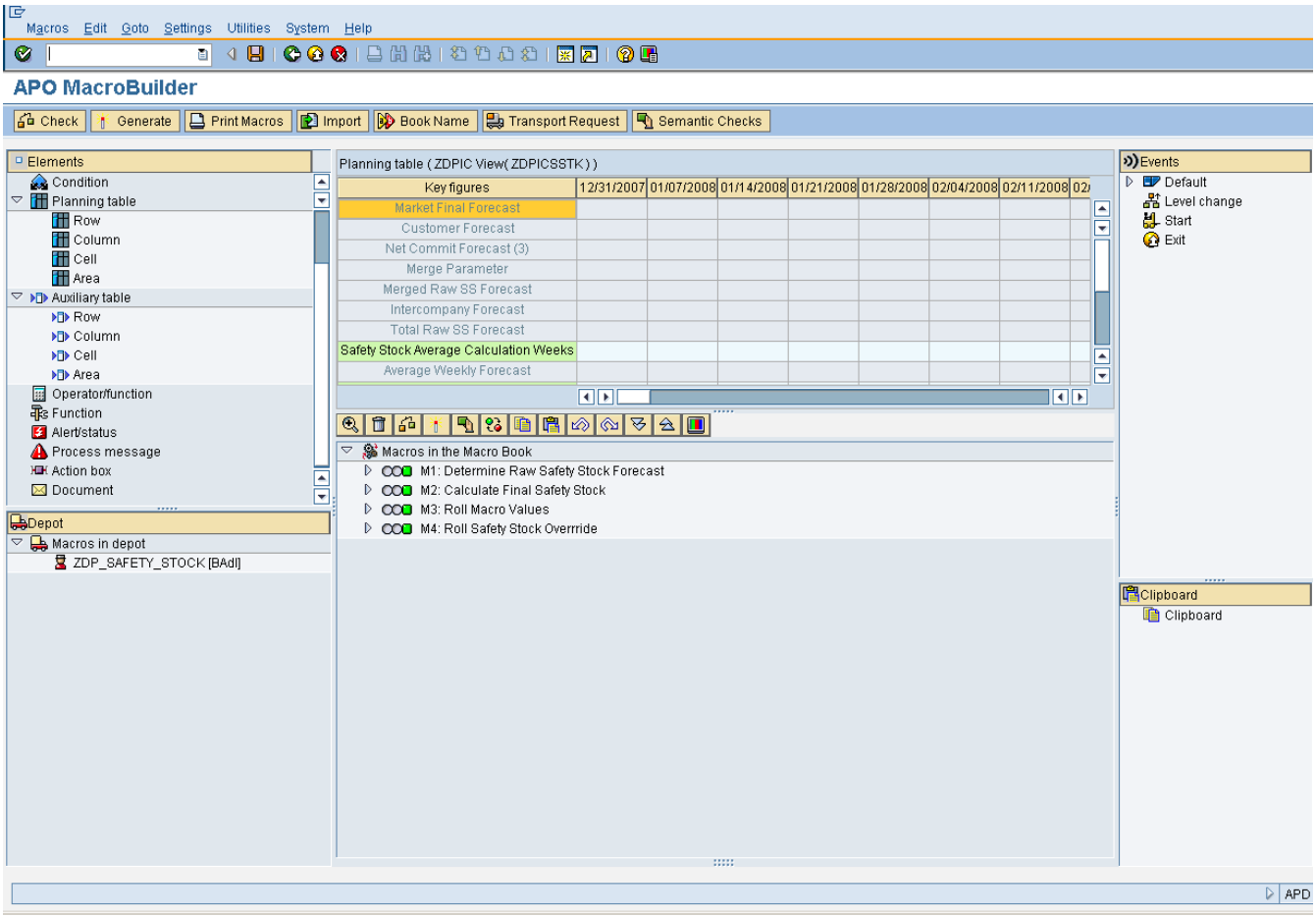


7. Start writing the BAdI code in this method. To close the parameters sub screen click Signature button on the application tool bar. . There is a sample code and proceedure explaining how to handle the data in the internal tables C_T_TAB and C_T_TAB_OLD. the calculations can be made with help of I_T_LINES, I_T_COLS which are rows and columns tables. find out the row and columns of the grid to be read and do calculation and then put the result in the desired cell.



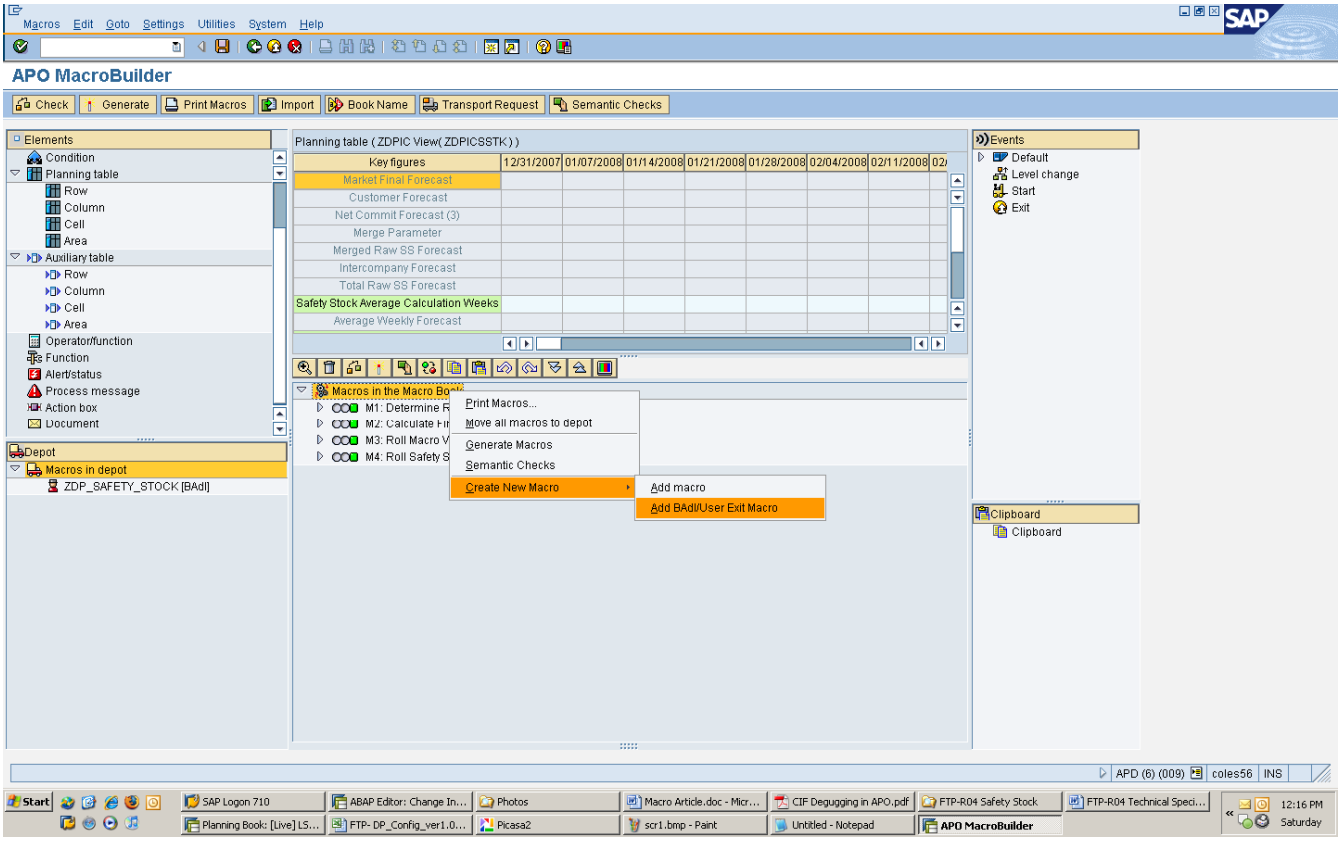
8. After writing the code, save and activate BAdI implementation.

9. Go to transaction /SAPAPO/ADVM select the appropriate planning book / data view.

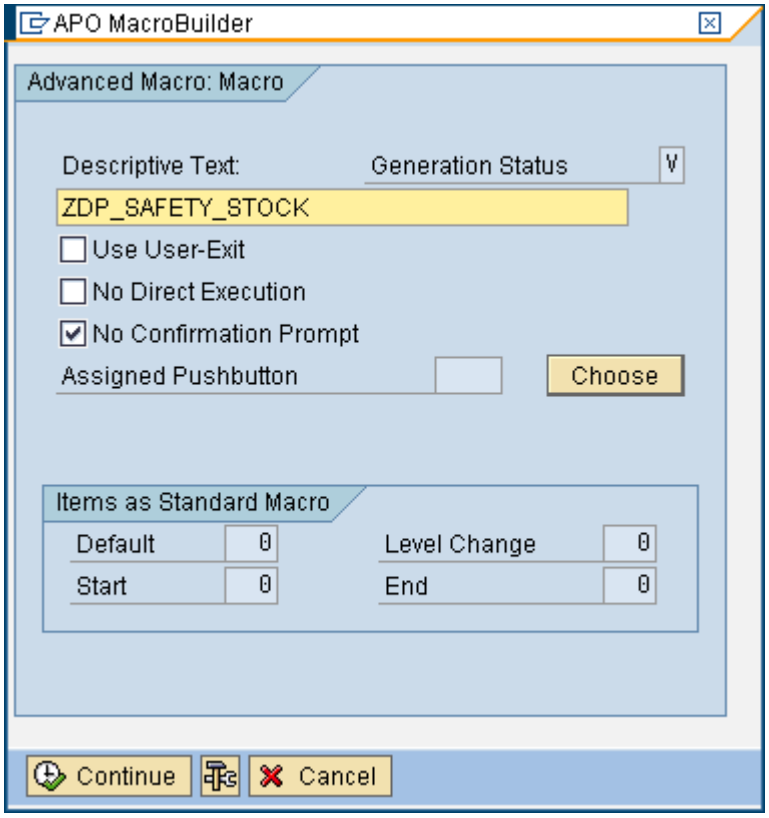


## Assign BAdI to Macro:

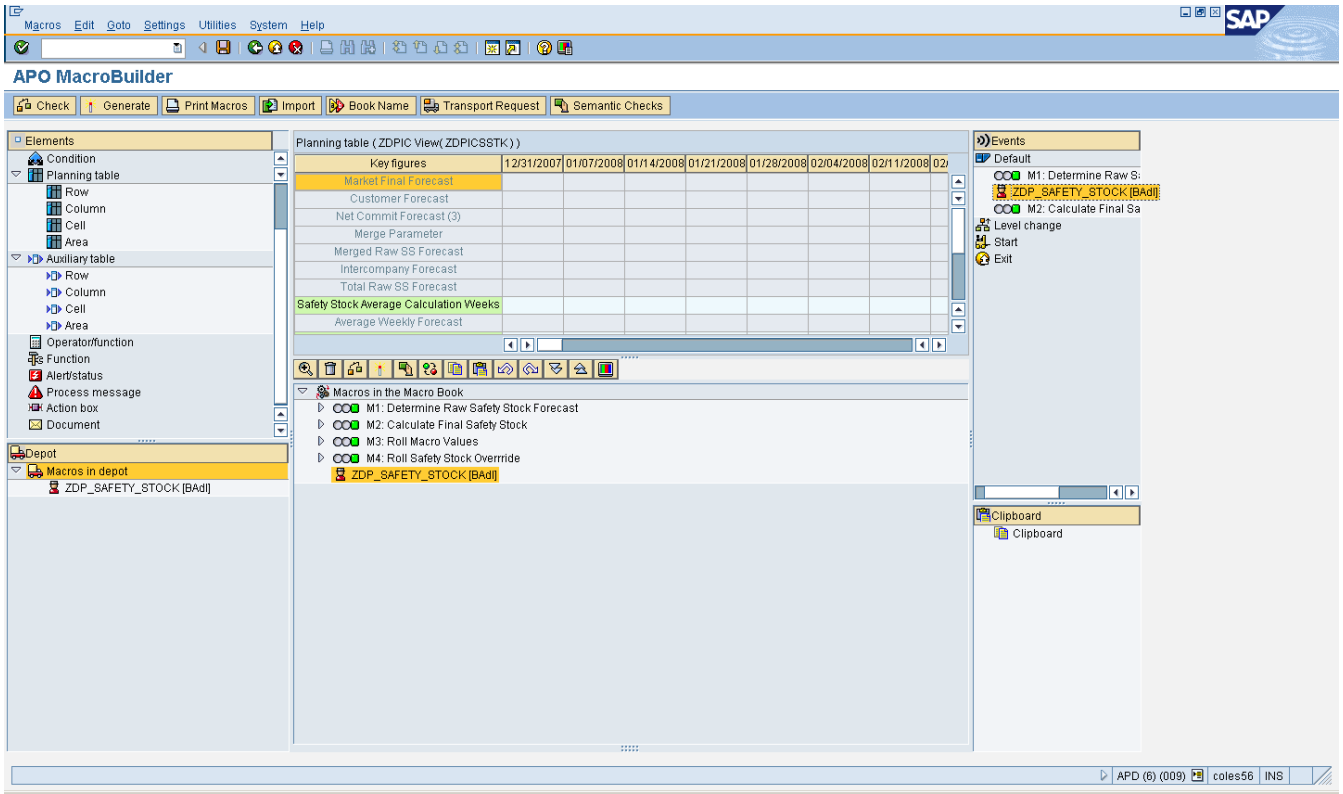
1. Right click on the macro book select new Macro → Add BAdI/User Exit Macro option.



2. Give the BAdI implementation name and click continue




3. Add the Macro BAdI to the default event to trigger for all the changes done in the screen. To do this just drag and drop the BAdI on to event sub screen. Then save and generate the Macro Builder.



## Source Code:

This is the code used in my program to achieve the requirement.

```
METHOD /SAPAPO/IF_EX_ADVX~USER_EXIT_MACRO.
************************************************************************
*                          V A R I A B L E S                           *
************************************************************************
  DATA: W_CALC_ERROR            TYPE /SAPAPO/ADV_CALC_ERROR,
        W_ARGUMENT              TYPE /SAPAPO/MXVAL,
        W_ACT_COLUMN            TYPE /SAPAPO/MXCOL,
        T_T_TAB                 TYPE STANDARD TABLE OF /SAPAPO/MSDP_APP_TAB_TAB,
        W_T_TAB                 TYPE /SAPAPO/MXSOP,
        W_SS_AVG_CAL_WKS        TYPE I,
        W_AVG_WK_FCST           TYPE F,
        W_BUILD_PLAN            TYPE I,
        W_TOT_BUILD_PLAN        TYPE I,
        W_CURRENT_WKS           TYPE I,
        W_NEXT_WKS              TYPE I.

  DATA: T_LINES                 TYPE /SAPAPO/MSDP_APP_LINES_TAB,
        W_LINES                 LIKE LINE OF T_LINES.

  DATA: T_COLS                  TYPE /SAPAPO/MSDP_APP_COLS_TAB,
        W_COLS                  LIKE LINE OF T_COLS.

  DATA: W_NUM_OF_WKS_STK_REQD   TYPE I,
        W_CAL_SFT_STK           TYPE F,
        W_FIXED_SFT_STK_REQD    TYPE I,
        W_TOT_SFT_STK           TYPE F,
        W_SFT_STK_ALC_WKS       TYPE I,
        W_SFT_STK               TYPE F,
        W_START_WK              TYPE I,
        W_COUNTER               TYPE I.

  FIELD-SYMBOLS:
        <FS_S_AGC_CLMN>         LIKE LINE OF C_S_ACTVIEW-AGC_GRID1.


************************************************************************
*                              M A I N                                 *
************************************************************************
*** Run the BADI only when the data in the grid is changed.
  IF C_T_TAB_OLD[] IS NOT INITIAL OR
     SY-UCOMM = 'TCHANGE' OR
     SY-UCOMM = 'SICH'.

** Collect the grid data like Rows and Columns into local internal tables.
    T_LINES[] = I_T_LINES[].
    T_COLS[] = I_T_COLS[].
    SORT T_LINES[] BY ZPOS.

* Read Current work week.
    READ TABLE T_COLS INTO W_COLS WITH KEY FIRSTFUTURE = 'X'.
    W_CURRENT_WKS = W_COLS-COLUMN.
```

```
*** Average Weekly Forecast.
* Read cell value of SS Avg. Cal. Weeks (2nd Row)
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZNSSAVGWK'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    W_SS_AVG_CAL_WKS = W_T_TAB-V.

* Read Row Matrix - Build Plan.
    CLEAR W_LINES.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQRTINTCO'.

* Read Buildpaln value and SUM upto Safety Stock Average calculation value.
* For example if SS Avg. Cal. Weeks are 5 then SUM buildplan values upto 5 columns.
    DO W_SS_AVG_CAL_WKS TIMES.
      CLEAR: W_T_TAB, W_BUILD_PLAN.
      READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
      IF SY-SUBRC = 0.
        W_BUILD_PLAN     = W_T_TAB-V.
        W_TOT_BUILD_PLAN = W_TOT_BUILD_PLAN + W_BUILD_PLAN.
      ENDIF.
        W_CURRENT_WKS    = W_CURRENT_WKS + 1.
    ENDDO.




* Calaulate Average Weekly forecast
* SUM of buildplan devided by SS Avg. Cal. Weeks value.
    W_AVG_WK_FCST = W_TOT_BUILD_PLAN / W_SS_AVG_CAL_WKS.

* Clear workarea and get current workweek.
    CLEAR: W_T_TAB, W_CURRENT_WKS.
    W_CURRENT_WKS = W_COLS-COLUMN.

* Insert the calculated value into the cell.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQAVGWKFC'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    IF SY-SUBRC NE 0.
      W_T_TAB-Z = W_LINES-LINE.
      W_T_TAB-C = W_CURRENT_WKS.
      W_T_TAB-V = W_AVG_WK_FCST.
      INSERT W_T_TAB INTO TABLE C_T_TAB.
    ELSE.
      W_T_TAB-V = W_AVG_WK_FCST.
      MODIFY TABLE C_T_TAB FROM W_T_TAB TRANSPORTING V.
    ENDIF.


*** Calculated Safety Stock.
* Read cell value of SS Avg. Cal. Weeks (4th Row in Planning book ZDP_SAFETY_STOCK)
    CLEAR : W_LINES, W_T_TAB.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZNSSWKREQ'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    W_NUM_OF_WKS_STK_REQD = W_T_TAB-V.

* Check the user enter a value in "Number of weeks stock required" cell.
*    IF NOT W_NUM_OF_WKS_STK_REQD IS INITIAL.
```

```
* Calaulated Safety Stock
    W_CAL_SFT_STK = W_NUM_OF_WKS_STK_REQD * W_AVG_WK_FCST.

* Insert the calculated safety stock value into the cell.
    CLEAR : W_LINES, W_T_TAB.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQCALSS'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    IF SY-SUBRC NE 0.
      W_T_TAB-Z = W_LINES-LINE.
      W_T_TAB-C = W_CURRENT_WKS.
      W_T_TAB-V = W_CAL_SFT_STK.
      INSERT W_T_TAB INTO TABLE C_T_TAB.
    ELSE.
      W_T_TAB-V = W_CAL_SFT_STK.
      MODIFY TABLE C_T_TAB FROM W_T_TAB TRANSPORTING V.
    ENDIF.
*    ENDIF.


*** Total Safety Stock
* Read cell value of Fixed Safety Stock required (6th Row in Planning book ZDP_SAFETY
_STOCK)
    CLEAR : W_LINES, W_T_TAB.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQFIXSS'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    W_FIXED_SFT_STK_REQD = W_T_TAB-V.

* Check if the user enter a value in "Fixed Safety Stock required" cell.
*    IF NOT W_FIXED_SFT_STK_REQD IS INITIAL.
* Calaulated Safety Stock
    W_TOT_SFT_STK = W_CAL_SFT_STK + W_FIXED_SFT_STK_REQD.

* Insert the calculated safety stock value into the cell.
    CLEAR : W_LINES, W_T_TAB.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQTOTSS'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    IF SY-SUBRC NE 0.
      W_T_TAB-Z = W_LINES-LINE.
      W_T_TAB-C = W_CURRENT_WKS.
      W_T_TAB-V = W_TOT_SFT_STK.
      INSERT W_T_TAB INTO TABLE C_T_TAB.
    ELSE.
      W_T_TAB-V = W_TOT_SFT_STK.
      MODIFY TABLE C_T_TAB FROM W_T_TAB TRANSPORTING V.
    ENDIF.
*    ENDIF.


*** Safety Stock
* Read cell value of Safety Stock allocation weeks
* (Row in Planning book ZDP_SAFETY_STOCK)
    CLEAR : W_LINES, W_T_TAB.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZNSSALLWK'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    W_SFT_STK_ALC_WKS = W_T_TAB-V.
```

```
* Check if the user enter a value in "Safety Stock allocation weeks" cell.
*    IF NOT w_sft_stk_alc_wks IS INITIAL.
* Safety Stock
    IF NOT W_SFT_STK_ALC_WKS IS INITIAL.
      W_SFT_STK = W_TOT_SFT_STK / W_SFT_STK_ALC_WKS.
    ELSE.
      CLEAR W_SFT_STK.
    ENDIF.
* Get the Safety Stock allocation start week.
    CLEAR : W_LINES, W_T_TAB, W_START_WK.
    READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZNALLSTWK'.
    READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_CURRENT_WKS.
    IF W_T_TAB-V IS INITIAL.
      W_START_WK = W_CURRENT_WKS.
    ELSE.
      W_START_WK = W_T_TAB-V - 1.
      W_START_WK = W_CURRENT_WKS + W_START_WK.
    ENDIF.

*** Clear all the cells which has the values less than start week.
*   This wil refresh the old values if any for the columns less than start week value
.
    W_NEXT_WKS = W_CURRENT_WKS.
*      DO w_t_tab-v TIMES.
    DO 53 TIMES.
      READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQSAFSTK'.
      READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_NEXT_WKS.
      IF SY-SUBRC = 0.
        W_T_TAB-V = 0.
        MODIFY TABLE C_T_TAB FROM W_T_TAB TRANSPORTING V.
        W_T_TAB-V = 1.
        INSERT W_T_TAB INTO TABLE C_T_TAB_OLD.
      ENDIF.
      W_NEXT_WKS = W_NEXT_WKS + 1.
      CLEAR W_T_TAB.
    ENDDO.

* Insert the calculated safety stock value into the cell.
    CLEAR : W_LINES, W_T_TAB, W_NEXT_WKS.
    W_NEXT_WKS = W_START_WK.
    DO W_SFT_STK_ALC_WKS TIMES.
      READ TABLE T_LINES INTO W_LINES WITH KEY FELDH = 'ZQSAFSTK'.
      READ TABLE C_S_ACTVIEW-
AGC_GRID1 ASSIGNING <FS_S_AGC_CLMN> WITH KEY AGC_INDEX = W_NEXT_WKS.
      <FS_S_AGC_CLMN>-MODIFIED = 'X'.
      READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_NEXT_WKS.
      IF SY-SUBRC NE 0.
        W_T_TAB-Z = W_LINES-LINE.
        W_T_TAB-C = W_NEXT_WKS.
        W_T_TAB-V = W_SFT_STK.
        INSERT W_T_TAB INTO TABLE C_T_TAB.
        W_T_TAB-V = 1.
        INSERT W_T_TAB INTO TABLE C_T_TAB_OLD.
      ELSE.
        W_T_TAB-V = W_SFT_STK.
        MODIFY TABLE C_T_TAB FROM W_T_TAB TRANSPORTING V.
```

```
        W_T_TAB-V = 1.
        INSERT W_T_TAB INTO TABLE C_T_TAB_OLD.
      ENDIF.
      W_NEXT_WKS = W_NEXT_WKS + 1.
    ENDDO.

* Refresh the all the cells where data is not available.
* This will refresh the column data which is more than start week + SSt calc weeks.
    CLEAR: W_NEXT_WKS, W_COUNTER.
    W_NEXT_WKS = W_START_WK.
    DO.
      W_COUNTER = W_COUNTER + 1.
      READ TABLE C_T_TAB INTO W_T_TAB WITH KEY Z = W_LINES-LINE C = W_NEXT_WKS.
      IF SY-SUBRC = 0.
        IF W_COUNTER > W_SFT_STK_ALC_WKS.
          W_T_TAB-V = 0.
          MODIFY TABLE C_T_TAB FROM W_T_TAB TRANSPORTING V.
          W_T_TAB-V = 1.
          INSERT W_T_TAB INTO TABLE C_T_TAB_OLD.
        ENDIF.
        W_NEXT_WKS = W_NEXT_WKS + 1.
        CLEAR W_T_TAB.
      ELSE.
        EXIT.
      ENDIF.
    ENDDO.
*    ENDIF.
  ENDIF.
ENDMETHOD.
```

## Related Content

Planning Book

Standard Macros

Advanced Macros

For more information, visit the Supply Chain Management homepage.

## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.