**LIT ASSIGNMENT-DAY 7**

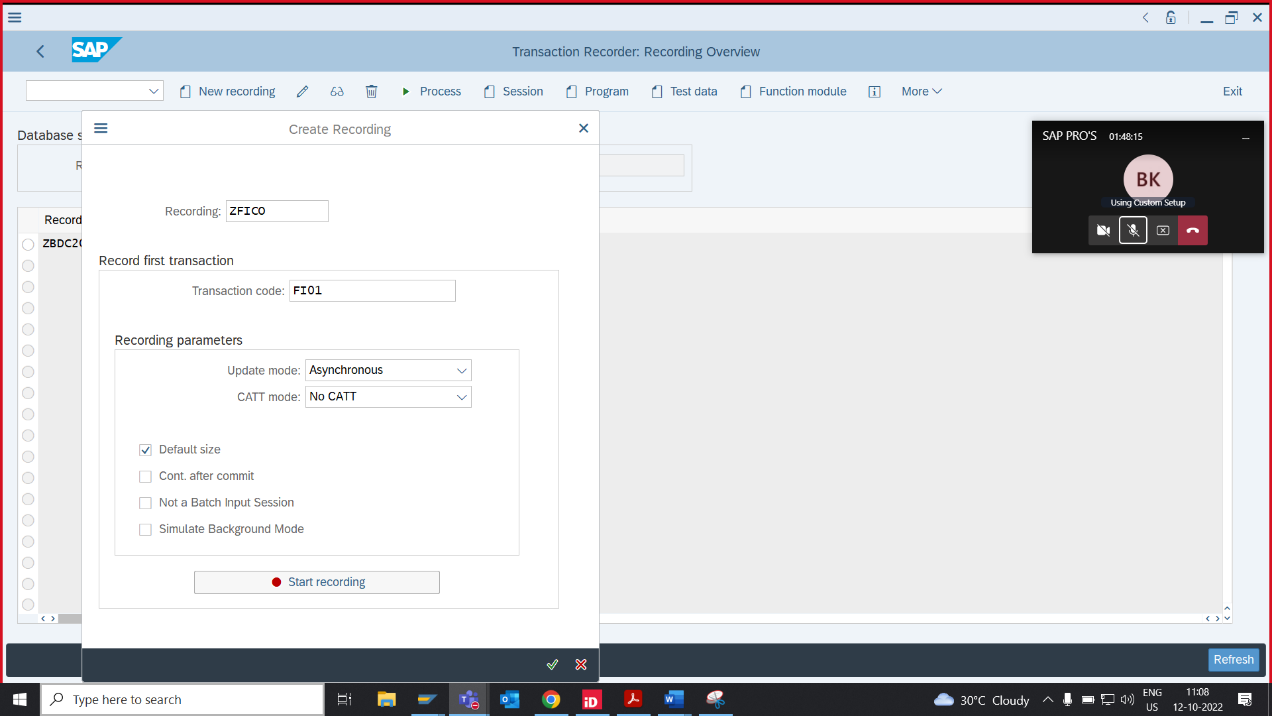
DATE : 09th September,2022  Batch: SAP ABAP ON HANA

EMP ID:46255236

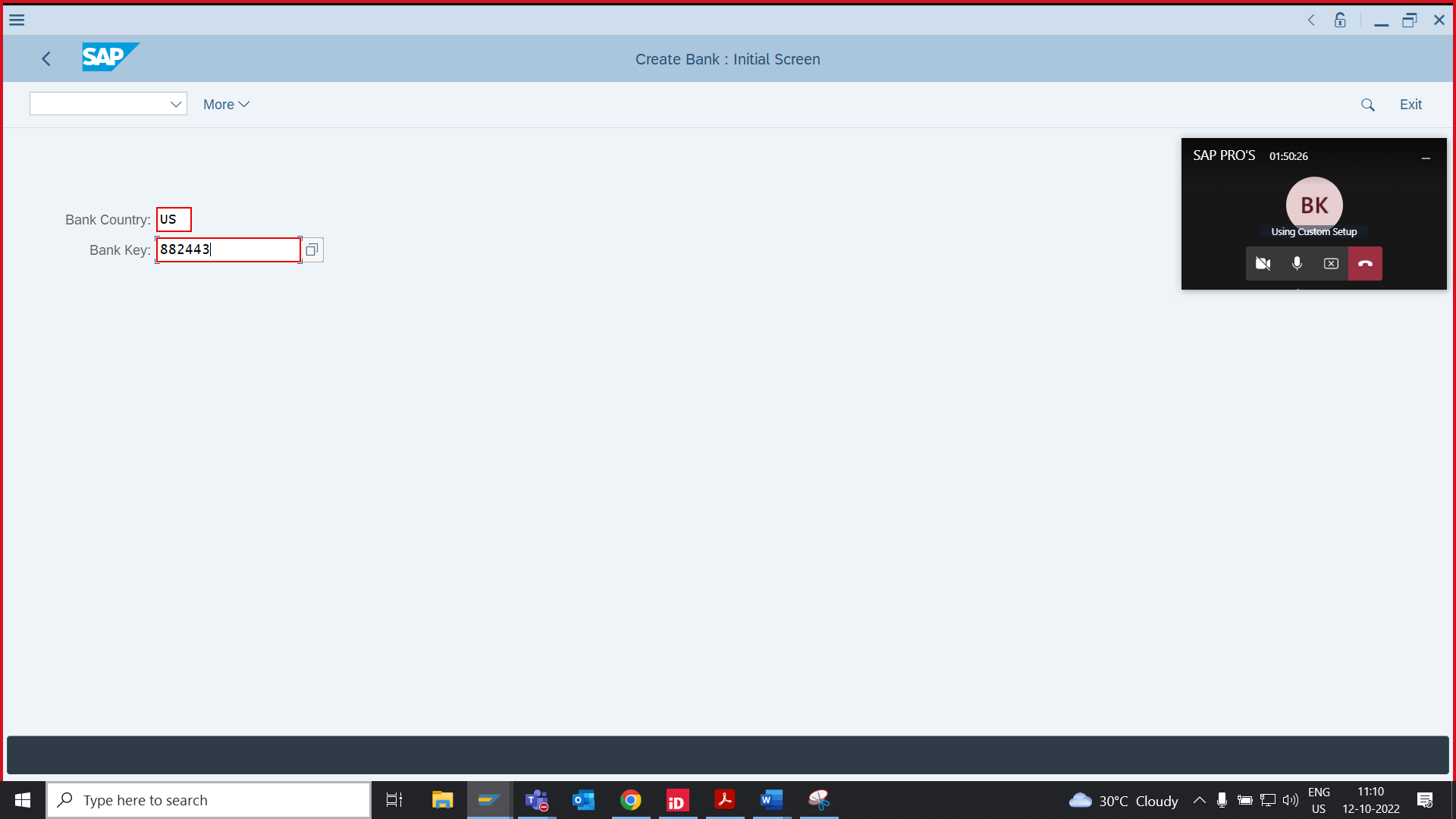
**Q1. Create an Upload program Using BDC Call Transaction to update data in VA01(Sales Order) .**

* **UPLOAD THE DATA USING FI01 TRANSACTION CODE.BEACAUSE** **WHEN WE USE TCODE(VA01) ,RUN TIME ERROR IS OCCURED DUE TO SERVER ISSUE - CHANGE IN KNA1 TABLE**

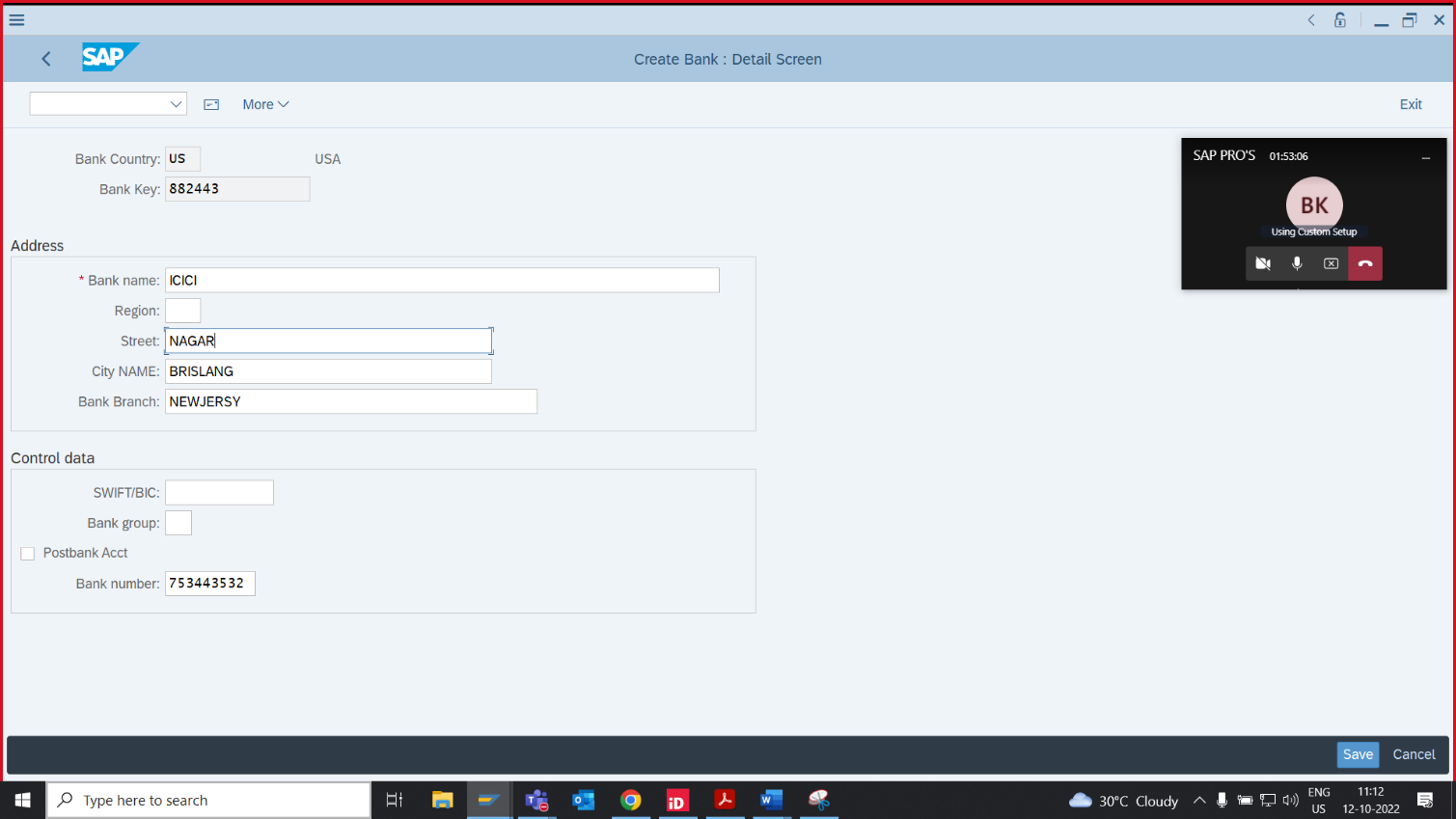
**STEP 1:** CREATE A NEW RECORDING IN SHDB.



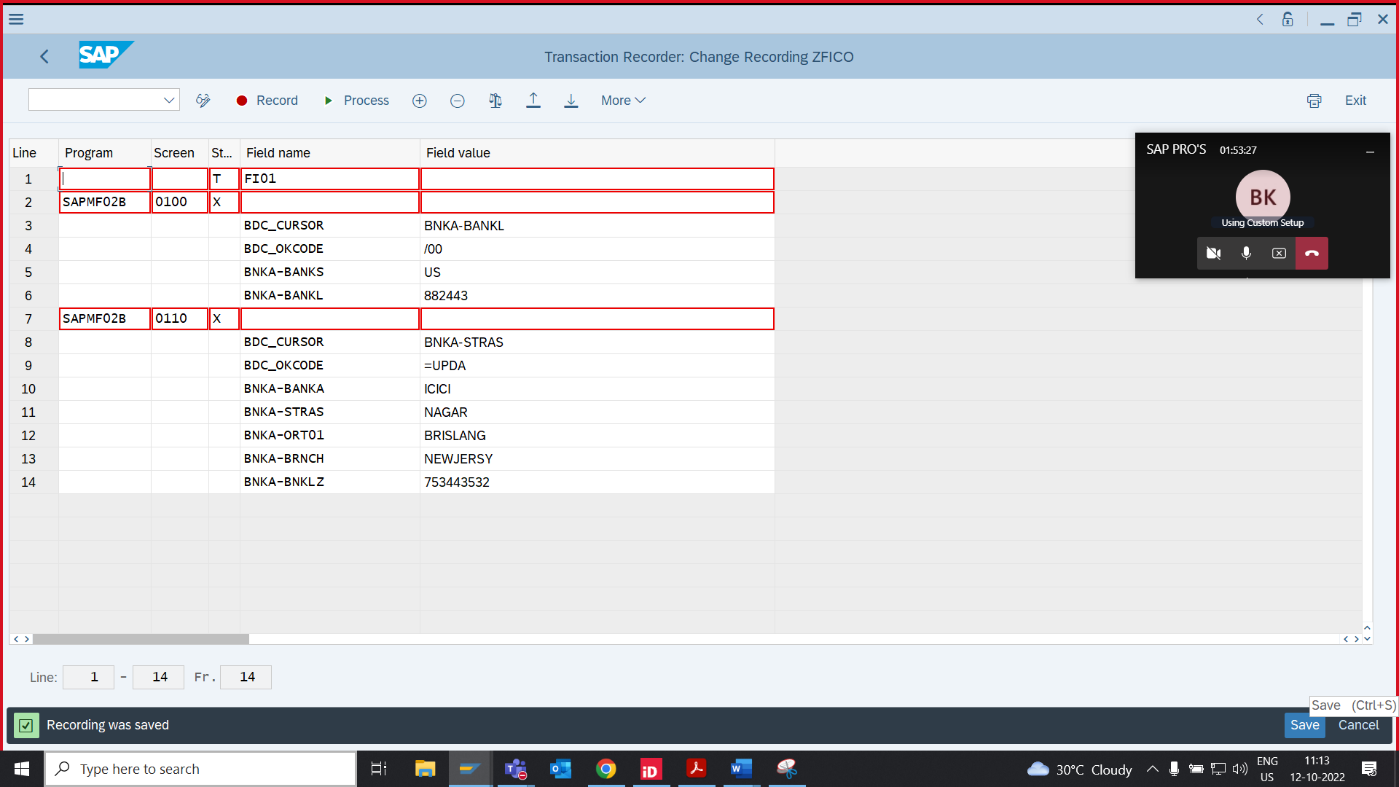
* Enter the Bank country and Key



* Now enter the Bank name ,City ,country and street.



Now Save the Recording.

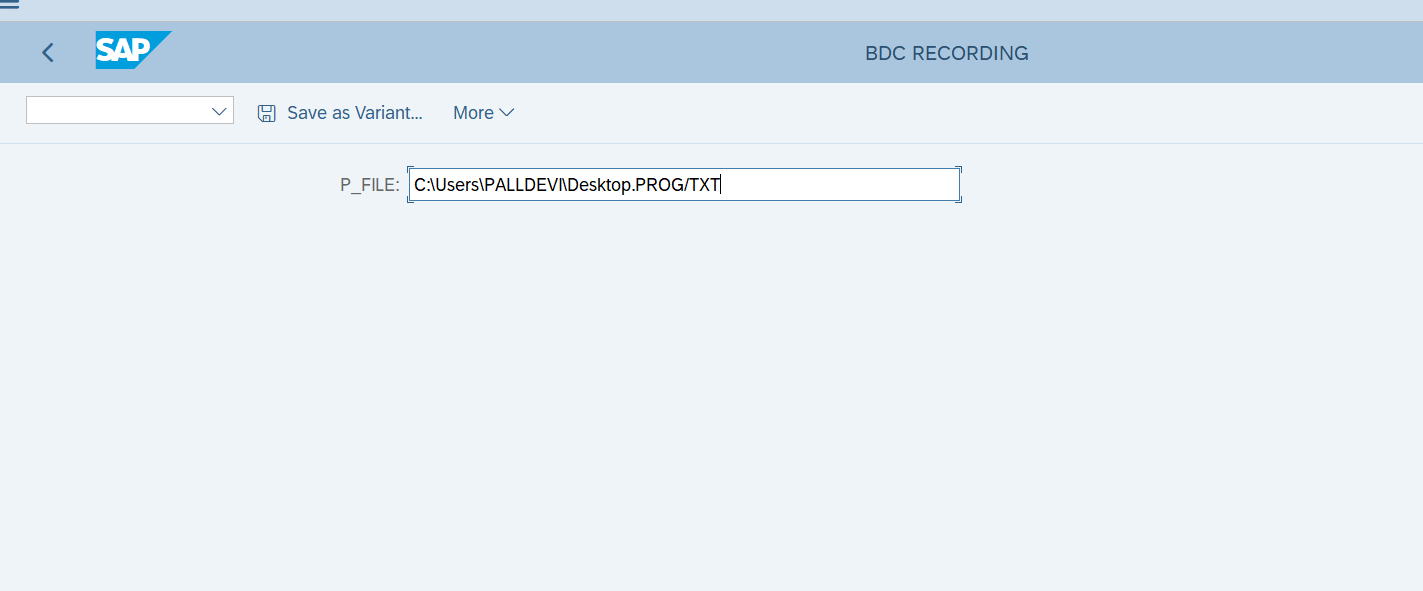


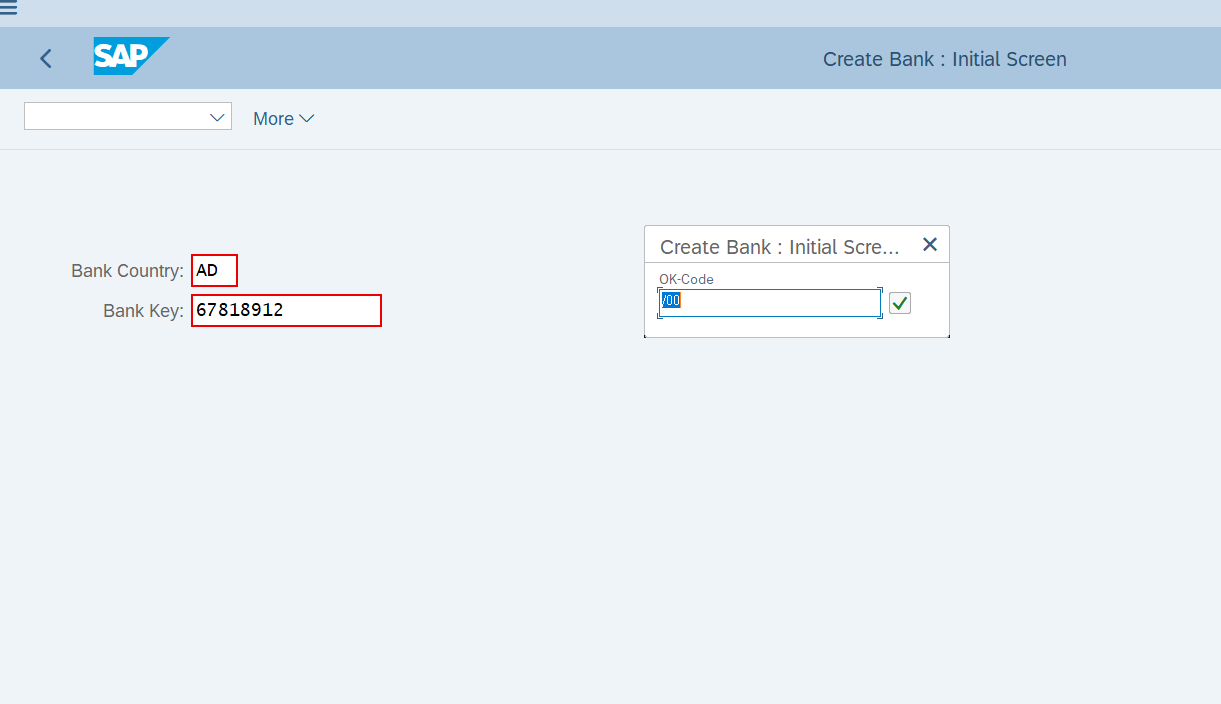
* Now select the created record and created a Program.
* Now BDC program will be created automatically and you can find the structures and dynpro and dynfield subroutines in Include BDCRECX1 main program.Copy them and paste in your source code.

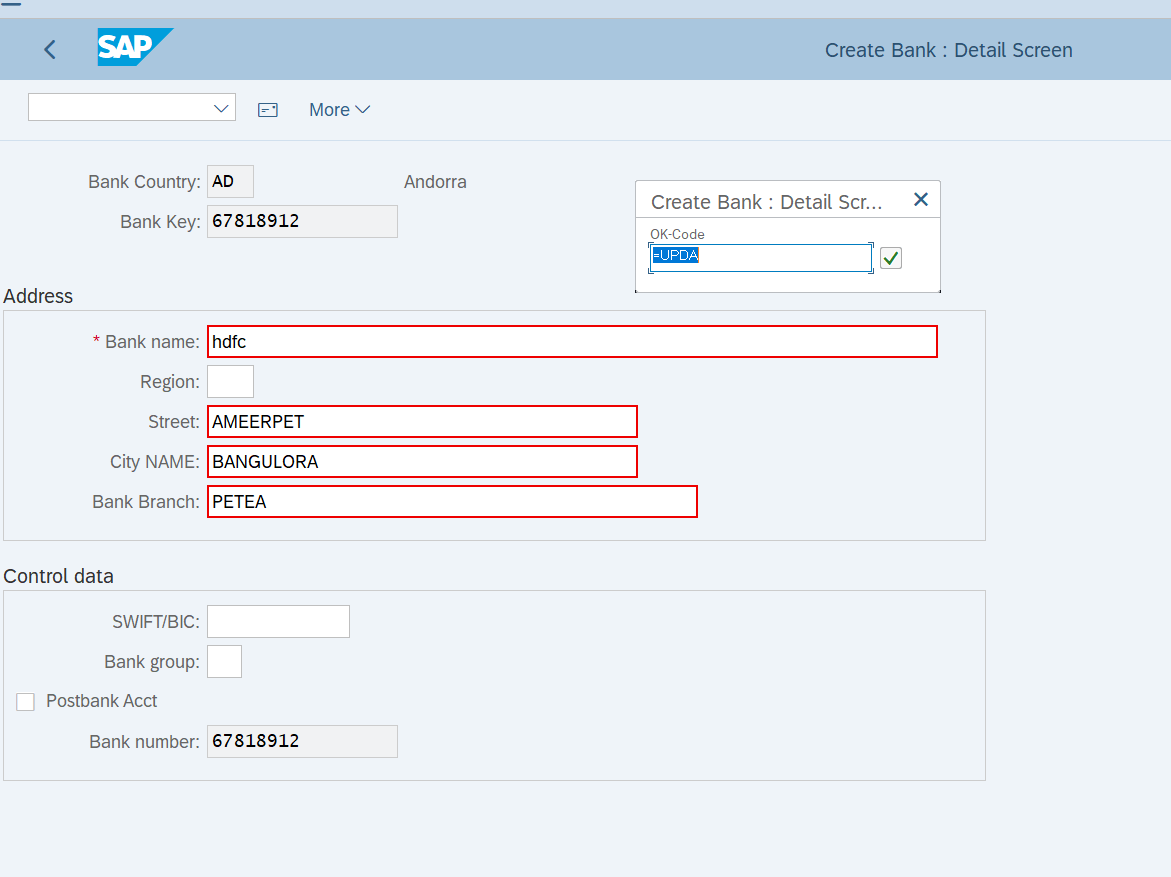
**REPORT:**

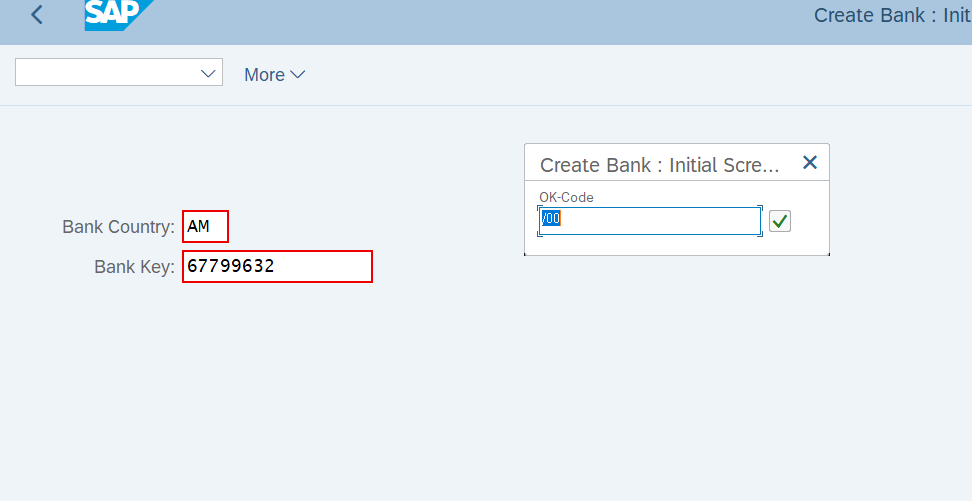
\*&---------------------------------------------------------------------\*  
\*& Report Z5236\_FICO\_BDC  
\*&---------------------------------------------------------------------\*  
\* Description:                  BDC Call Transacion to update data in FI01     
\*                                                                      \*  
\* Author:              ANJALI DEVI PALLAPU                                
\*                                                                      \*  
\* Create date:          12/10/2022                                                                                                                  
  
REPORT Z5236\_FICO\_BDC.  
  
types : begin of ty\_bnka,  
         banks type bnka-banks,  
         bankl type bnka-bankl,  
         banka type bnka-banka,  
        stras type bnka-stras,  
        ort01 type bnka-ort01,  
        BRNCH type bnka-BRNCH,  
        BNKLZ type bnka-BNKLZ,  
         end of ty\_bnka.  
  
         data : it\_bnka type table of ty\_bnka,  
                wa\_bnka type ty\_bnka.  
         data : it\_bdcdata type table of bdcdata,  
                wa\_bdcdata type bdcdata.  
  
         data : bdcmsg type table of bdcmsgcoll.  
         data : file type string.  
  
         parameters : p\_file type rlgrap-filename.  
  
  
start-of-selection.  
  
         if  p\_file is not initial.  
           file = p\_file.  
           ENDIF.  
  
CALL FUNCTION 'GUI\_UPLOAD'  
  EXPORTING  
    filename                      = file  
   FILETYPE                      = 'ASC'  
   HAS\_FIELD\_SEPARATOR           = 'X'  
  
  tables  
    data\_tab                      = it\_bnkA  
  
 EXCEPTIONS  
   FILE\_OPEN\_ERROR               = 1  
   FILE\_READ\_ERROR               = 2  
   NO\_BATCH                      = 3  
   GUI\_REFUSE\_FILETRANSFER       = 4  
   INVALID\_TYPE                  = 5  
   NO\_AUTHORITY                  = 6  
   UNKNOWN\_ERROR                 = 7  
   BAD\_DATA\_FORMAT               = 8  
   HEADER\_NOT\_ALLOWED            = 9  
   SEPARATOR\_NOT\_ALLOWED         = 10  
   HEADER\_TOO\_LONG               = 11  
   UNKNOWN\_DP\_ERROR              = 12  
   ACCESS\_DENIED                 = 13  
   DP\_OUT\_OF\_MEMORY              = 14  
   DISK\_FULL                     = 15  
   DP\_TIMEOUT                    = 16  
   OTHERS                        = 17  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
\*perform open\_group.  
LOOP AT IT\_BNKA INTO WA\_BNKA.  
  REFRESH : bdcmsg,IT\_BDCDATA.  
  
perform bdc\_dynpro      using 'SAPMF02B' '0100'.  
perform bdc\_field       using 'BDC\_CURSOR'  
                              'BNKA-BANKL'.  
perform bdc\_field       using 'BDC\_OKCODE'  
                              '/00'.  
perform bdc\_field       using 'BNKA-BANKS' WA\_BNKA-BANKS.  
\*                              'US'  
perform bdc\_field       using 'BNKA-BANKL' WA\_BNKA-BANKL.  
\*                              '882443'  
perform bdc\_dynpro      using 'SAPMF02B' '0110' .  
perform bdc\_field       using 'BDC\_CURSOR'  
                              'BNKA-STRAS'.  
perform bdc\_field       using 'BDC\_OKCODE'  
                              '=UPDA'.  
perform bdc\_field       using 'BNKA-BANKA' WA\_BNKA-BANKA.  
\*                              'ICICI'.  
perform bdc\_field       using 'BNKA-STRAS'  WA\_BNKA-STRAS.  
\*                              'NAGAR'.  
perform bdc\_field       using 'BNKA-ORT01' WA\_BNKA-ORT01.  
\*                              'BRISLANG'.  
perform bdc\_field       using 'BNKA-BRNCH' wa\_BNKA-BRNCH.  
\*                              'NEWJERSY'.  
perform bdc\_field       using 'BNKA-BNKLZ' WA\_BNKA-BNKLZ.  
\*                              '753443532'.  
\*perform bdc\_transaction using 'FI01'.  
  
  
   CALL TRANSACTION 'FI01' USING IT\_BDCDATA MODE 'A' UPDATE 'A' MESSAGES INTO BDCMSG.  
  
\*perform close\_group.  
ENDLOOP.  
  
  
DATA : WA\_BDCMSG LIKE LINE OF BDCMSG.  
IF BDCMSG IS NOT INITIAL.  
  LOOP AT BDCMSG INTO WA\_BDCMSG.  
    WRITE:/ WA\_BDCMSG-TCODE,  
           WA\_BDCMSG-MSGTYP,  
           WA\_BDCMSG-MSGV1,  
            WA\_BDCMSG-FLDNAME.  
    CLEAR WA\_BDCMSG.  
    ENDLOOP.  
    ENDIF.  
  
FORM BDC\_DYNPRO USING PROGRAM DYNPRO.  
  CLEAR WA\_BDCDATA.  
  wa\_BDCDATA-PROGRAM  = PROGRAM.  
  wa\_BDCDATA-DYNPRO   = DYNPRO.  
  wa\_BDCDATA-DYNBEGIN = 'X'.  
  APPEND WA\_BDCDATA TO IT\_BDCDATA.  
ENDFORM.  
  
\*----------------------------------------------------------------------\*  
\*        Insert field                                                  \*  
\*----------------------------------------------------------------------\*  
FORM BDC\_FIELD USING FNAM FVAL.  
\*  IF FVAL <> NODATA.  
    CLEAR wa\_BDCDATA.  
    wa\_BDCDATA-FNAM = FNAM.  
    wa\_BDCDATA-FVAL = FVAL.  
    APPEND wa\_BDCDATA TO IT\_BDCDATA.  
\*  ENDIF.  
ENDFORM.

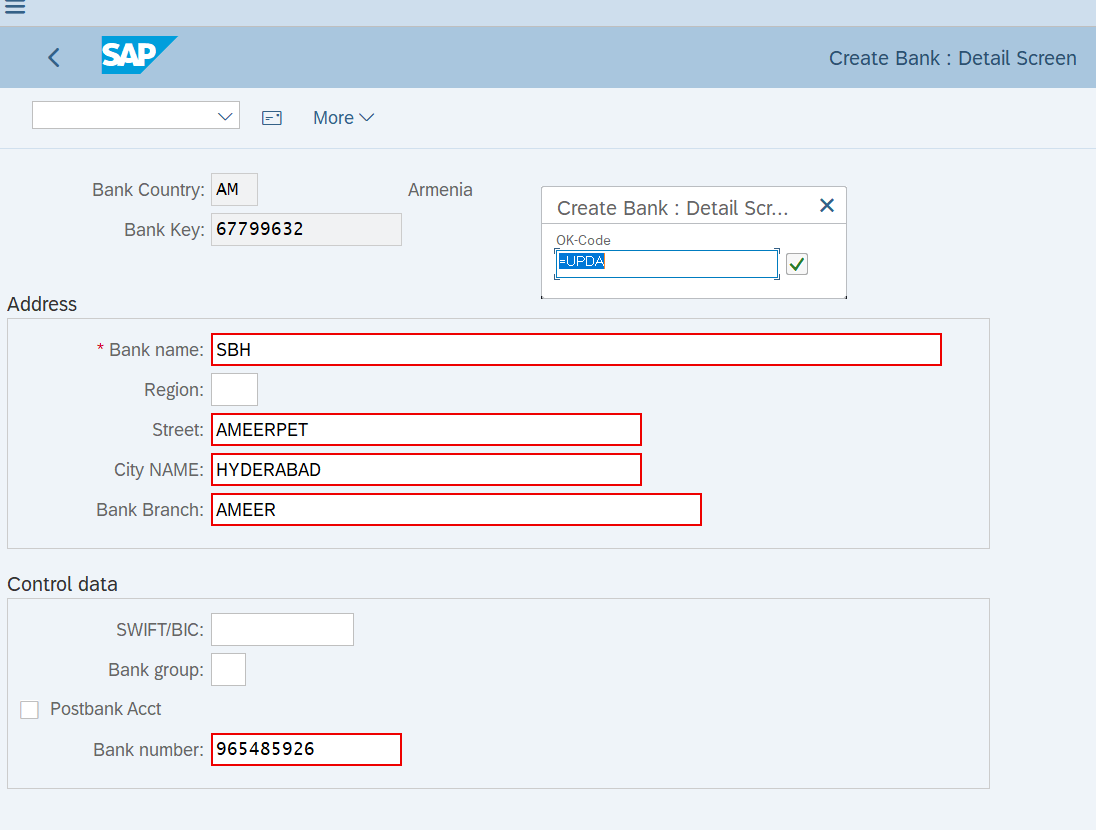
OUTPUT: Read data from the flat file

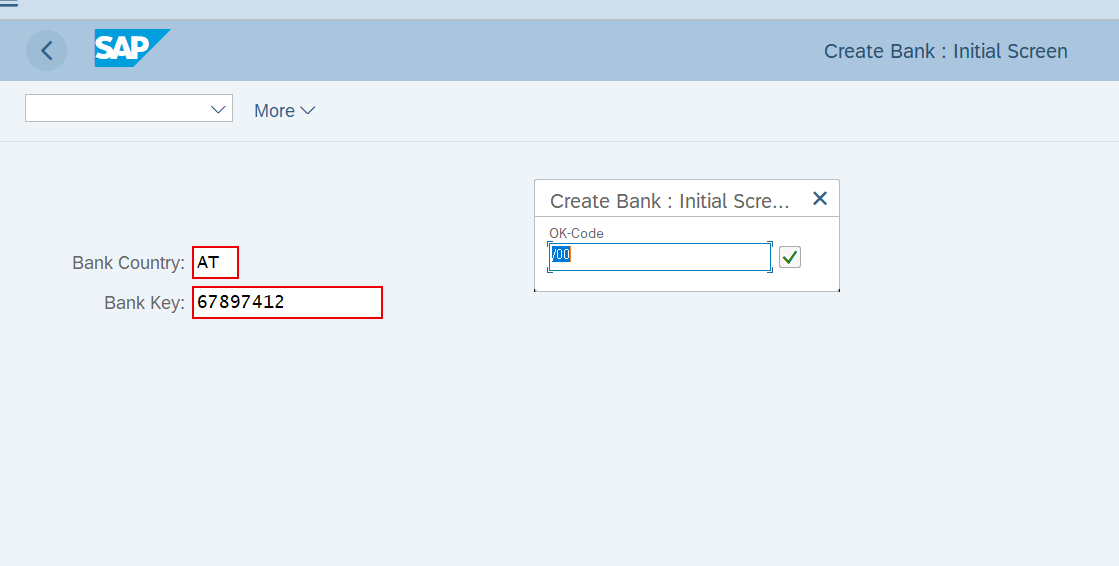


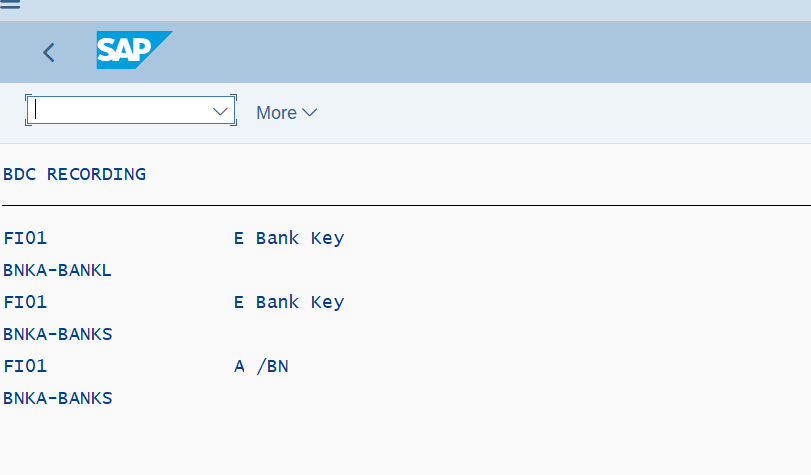






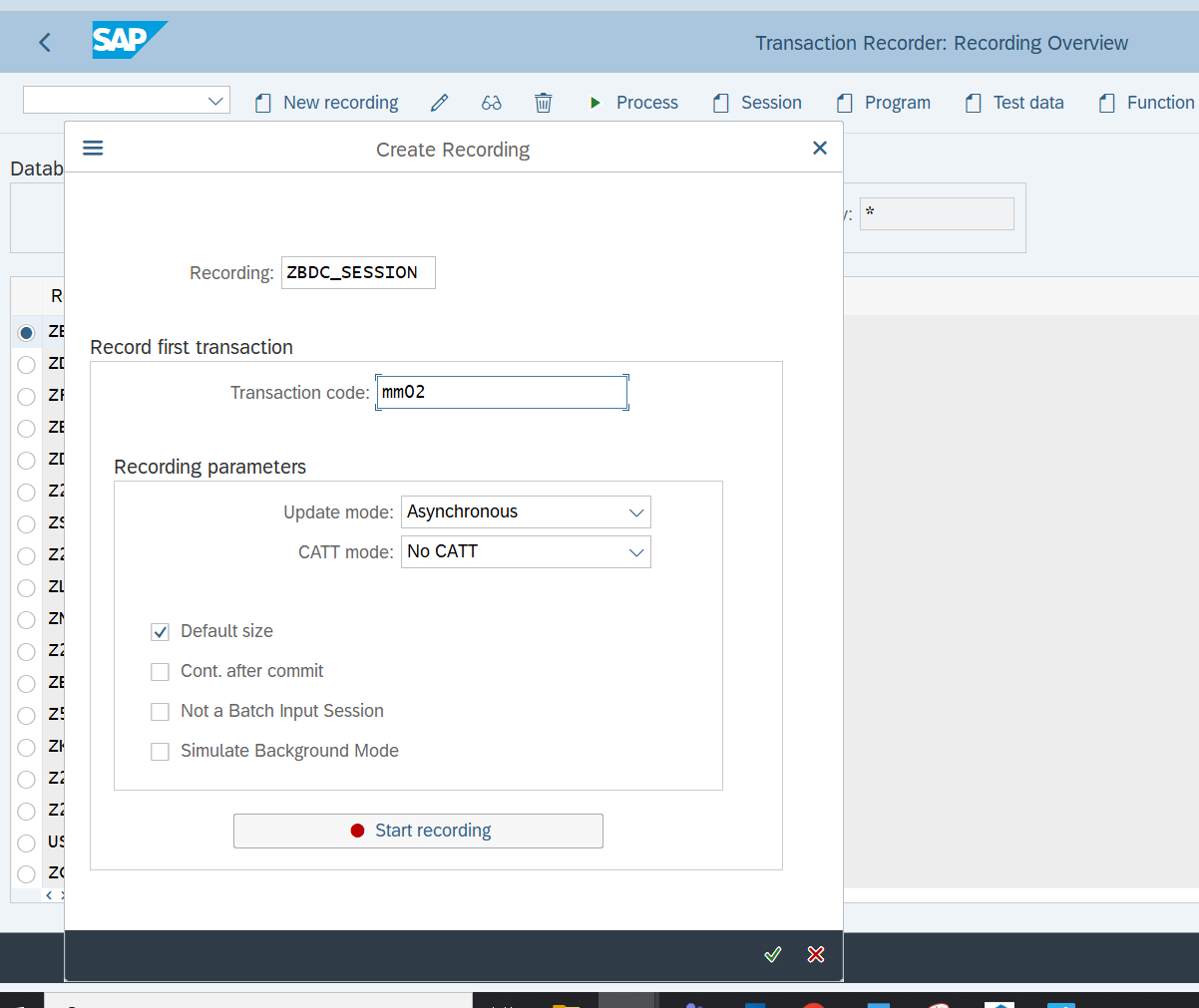




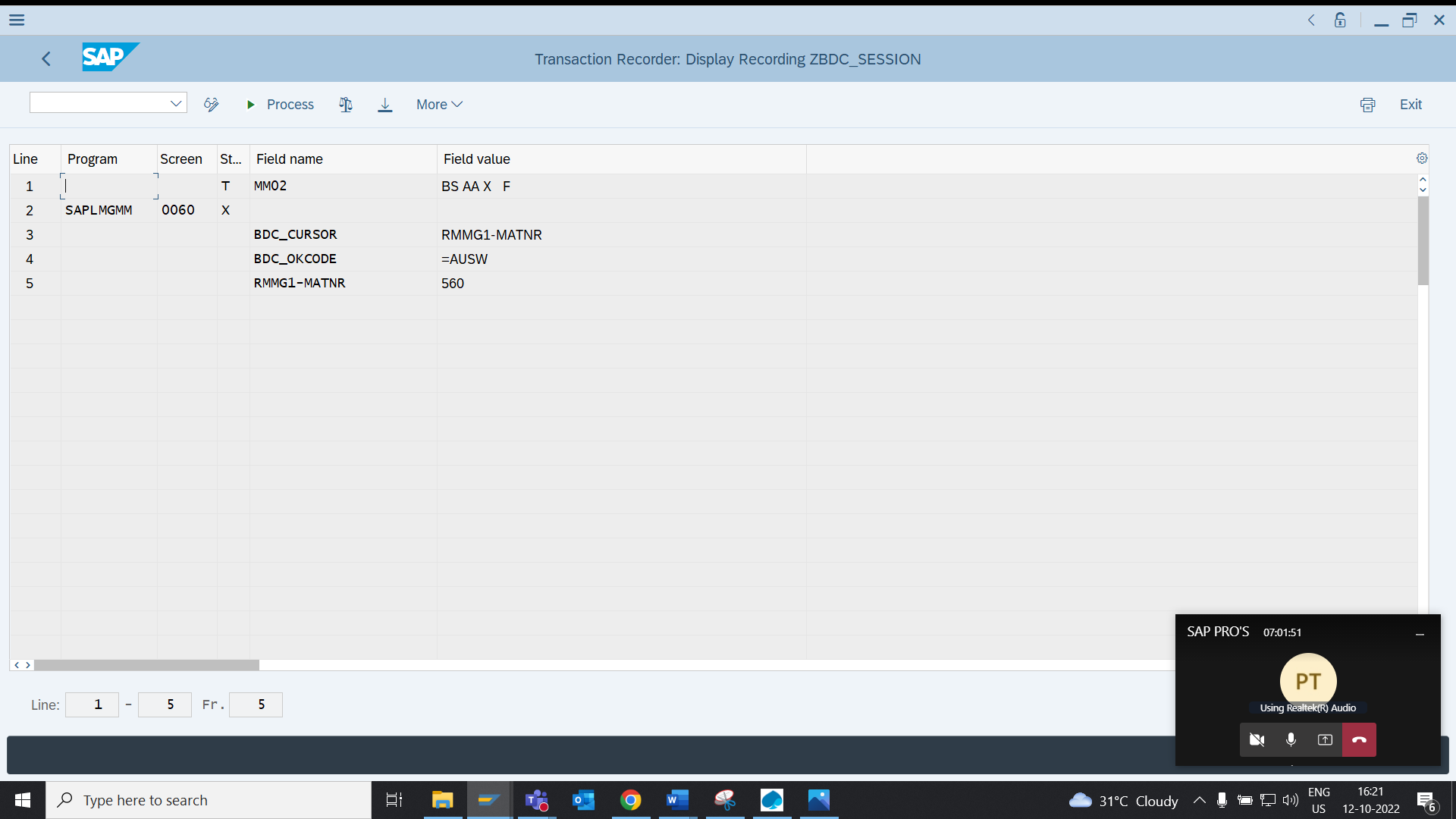


**Q2**. **Create an Upload program Using BDC Session Method to update data in MM02(Material Change).**

**STEP 1: created the new record in SHDB.**

****

Now Save the Recording .



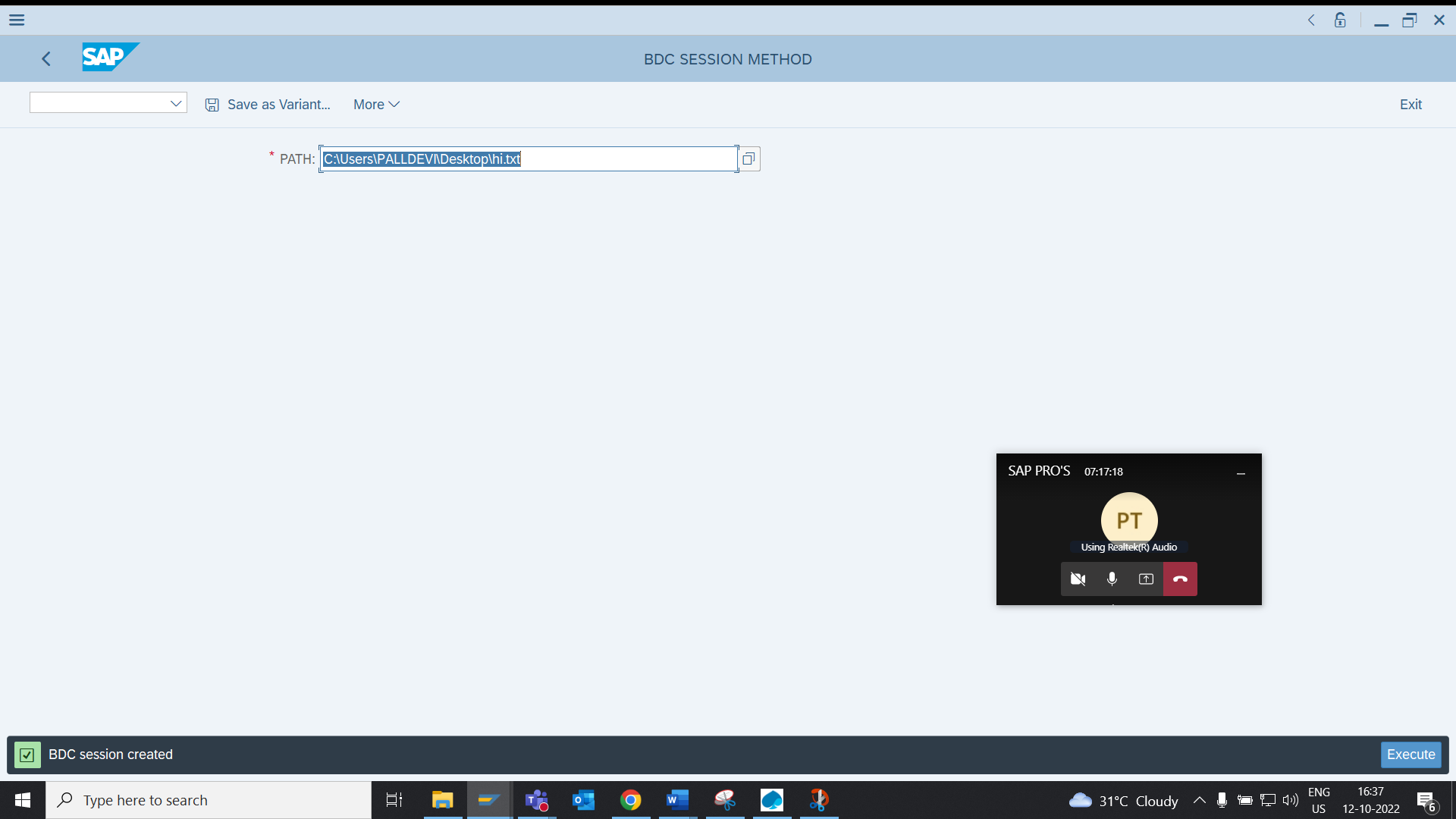
* Now select the created record and click on Program and create the program.
* Then BDC program will be created automatically and you can find the structures and dynpro and dynfield subroutines in Include BDCRECX1 main program.Copy them and paste in your source code.

**REPORT:**

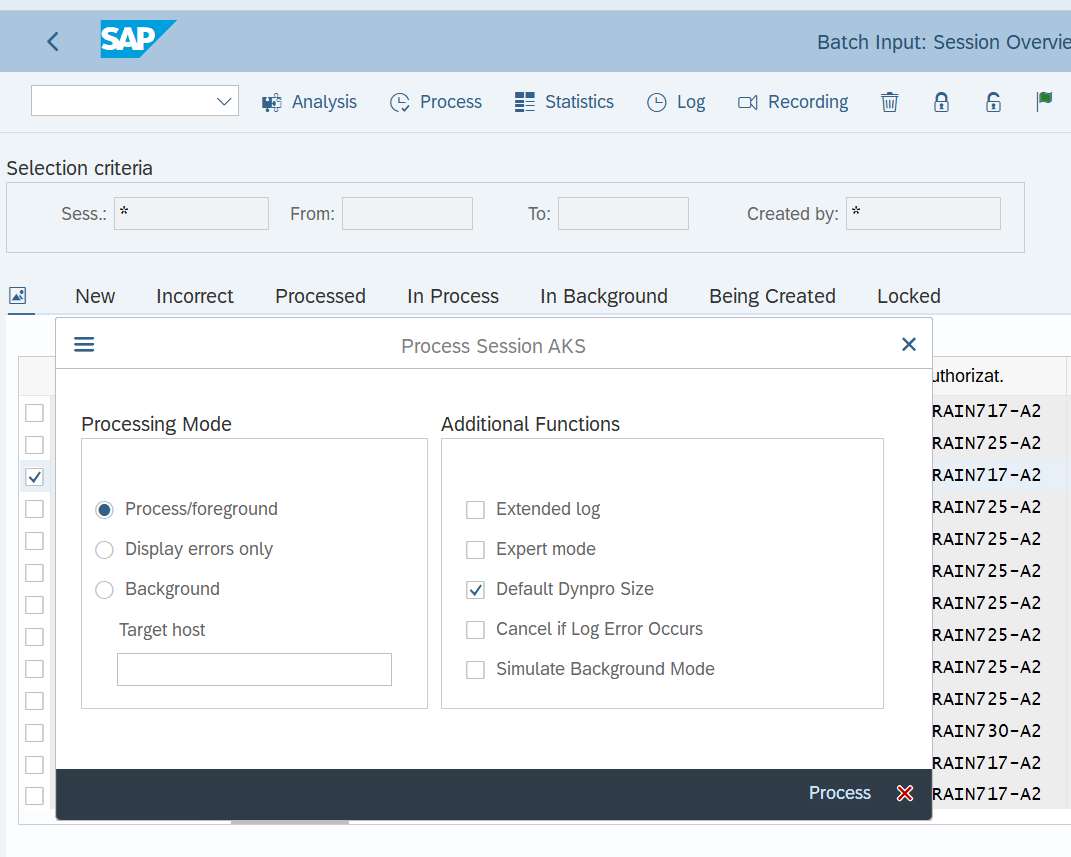
\*&---------------------------------------------------------------------\*  
\*& Report Z5236\_BDC\_SESSION  
\*&---------------------------------------------------------------------\*  
\* Description:           BDC SESSION METHOD USING MM02                                              \*  
\*                                                                      \*  
\* Author:             ANJALI DEVI PALLAPU                                                 \*  
\*                                                                      \*  
\* Create date:                                                         \*  
\*                                                                      \*  
\*&---------------------------------------------------------------------\*  
  
REPORT Z5236\_BDC\_SESSION.  
  
TYPES:BEGIN OF TY\_MAT,  
       MATNR TYPE MATNR,  
       END OF TY\_MAT.  
  
DATA: IT\_MA TYPE TABLE OF TY\_MAT,  
      WA\_MA TYPE TY\_MAT,  
  F\_NAME TYPE STRING.  
  
  
DATA:   BDCDATA LIKE BDCDATA .   
\*       messages of call transaction  
DATA:   MESSTAB LIKE BDCMSGCOLL.  
  
  
PARAMETERS : path TYPE rlgrap-filename OBLIGATORY.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR path .  
  
  CALL FUNCTION 'F4\_FILENAME'  
  
    EXPORTING  
      program\_name  = syst-cprog  
      dynpro\_number = syst-dynnr  
      field\_name    = 'PATH'  
    IMPORTING  
      file\_name     = path.  
  
START-OF-SELECTION.  
F\_NAME = PATH.  
CALL FUNCTION 'GUI\_UPLOAD'  
  EXPORTING  
    filename                      = F\_NAME  
   FILETYPE                      = 'ASC'  
   HAS\_FIELD\_SEPARATOR           = 'X'  
  tables  
    data\_tab                      = IT\_MA  
\* CHANGING  
\*   ISSCANPERFORMED               = ' '  
\* EXCEPTIONS  
\*   FILE\_OPEN\_ERROR               = 1  
\*   FILE\_READ\_ERROR               = 2  
\*   NO\_BATCH                      = 3  
\*   GUI\_REFUSE\_FILETRANSFER       = 4  
\*   INVALID\_TYPE                  = 5  
\*   NO\_AUTHORITY                  = 6  
\*   UNKNOWN\_ERROR                 = 7  
\*   BAD\_DATA\_FORMAT               = 8  
\*   HEADER\_NOT\_ALLOWED            = 9  
\*   SEPARATOR\_NOT\_ALLOWED         = 10  
\*   HEADER\_TOO\_LONG               = 11  
\*   UNKNOWN\_DP\_ERROR              = 12  
\*   ACCESS\_DENIED                 = 13  
\*   DP\_OUT\_OF\_MEMORY              = 14  
\*   DISK\_FULL                     = 15  
\*   DP\_TIMEOUT                    = 16  
\*   OTHERS                        = 17  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
CALL FUNCTION 'BDC\_OPEN\_GROUP'  
 EXPORTING  
   CLIENT                    = SY-MANDT  
\*   DEST                      = FILLER8  
   GROUP                     = 'AKS'  
\*   HOLDDATE                  = FILLER8  
   KEEP                      = 'X'  
   USER                      = sy-uname   
   PROG                      = SY-CPROG  
=  
 EXCEPTIONS  
   CLIENT\_INVALID            = 1  
   DESTINATION\_INVALID       = 2  
   GROUP\_INVALID             = 3  
   GROUP\_IS\_LOCKED           = 4  
   HOLDDATE\_INVALID          = 5  
   INTERNAL\_ERROR            = 6  
   QUEUE\_ERROR               = 7  
   RUNNING                   = 8  
   SYSTEM\_LOCK\_ERROR         = 9  
   USER\_INVALID              = 10  
   OTHERS                    = 11  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
  
  
LOOP AT IT\_MA INTO WA\_MA.  
  REFRESH : MESSTAB, BDCDATA.  
  
perform bdc\_dynpro      using 'SAPLMGMM' '0060'.  
perform bdc\_field       using 'BDC\_CURSOR'  
                              'RMMG1-MATNR'.  
perform bdc\_field       using 'BDC\_OKCODE'  
                              '=AUSW'.  
perform bdc\_field       using 'RMMG1-MATNR'  
                               WA\_MA-MATNR.  
  
  
  
CALL FUNCTION 'BDC\_INSERT'  
 EXPORTING  
   TCODE                  = 'MM02'  
  TABLES  
    DYNPROTAB              = BDCDATA  
 EXCEPTIONS  
   INTERNAL\_ERROR         = 1  
   NOT\_OPEN               = 2  
   QUEUE\_ERROR            = 3  
   TCODE\_INVALID          = 4  
   PRINTING\_INVALID       = 5  
   POSTING\_INVALID        = 6  
   OTHERS                 = 7  
          .  
IF SY-SUBRC <> 0.  
\* Implement suitable error handling here  
ENDIF.  
clear : WA\_MA.  
endloop.  
  
CALL FUNCTION 'BDC\_CLOSE\_GROUP'  
 EXCEPTIONS  
   NOT\_OPEN          = 1  
   QUEUE\_ERROR       = 2  
   OTHERS            = 3  
          .  
IF SY-SUBRC <> 0.  
\* Implement suitable error handling here  
  message 'Error while creating bdc session' TYPE 'E'.  
ELSE.  
   MESSAGE 'BDC session created' TYPE 'S'.  
  
ENDIF.  
  
FORM BDC\_DYNPRO USING PROGRAM DYNPRO.  
  CLEAR BDCDATA.  
  BDCDATA-PROGRAM  = PROGRAM.  
  BDCDATA-DYNPRO   = DYNPRO.  
  BDCDATA-DYNBEGIN = 'X'.  
  APPEND BDCDATA.  
ENDFORM.  
  
\*----------------------------------------------------------------------\*  
\*        Insert field                                                  \*  
\*----------------------------------------------------------------------\*  
FORM BDC\_FIELD USING FNAM FVAL.  
  
    CLEAR BDCDATA.  
    BDCDATA-FNAM = FNAM.  
    BDCDATA-FVAL = FVAL.  
    APPEND BDCDATA.  
  
ENDFORM.

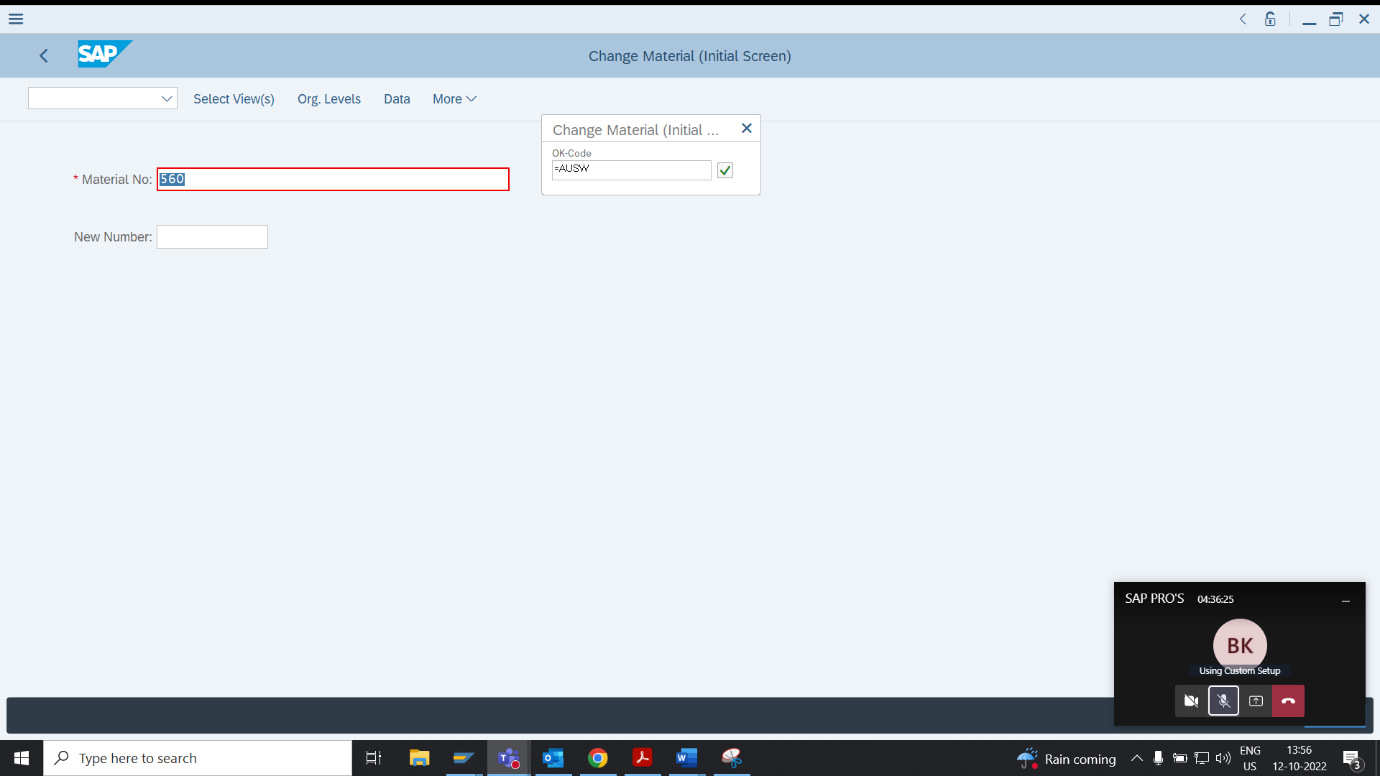
**OUTPUT:**

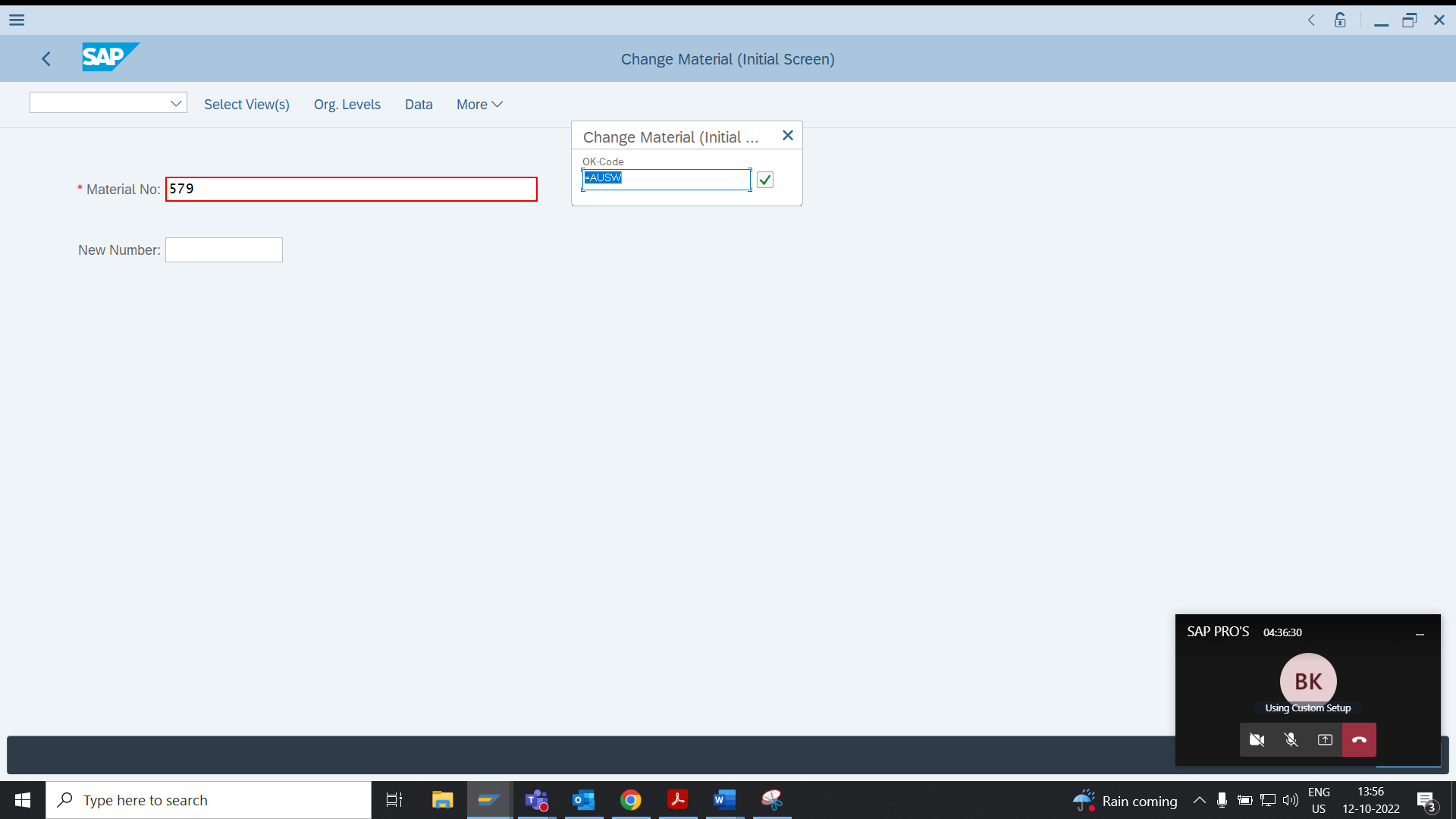
**selection screen:**

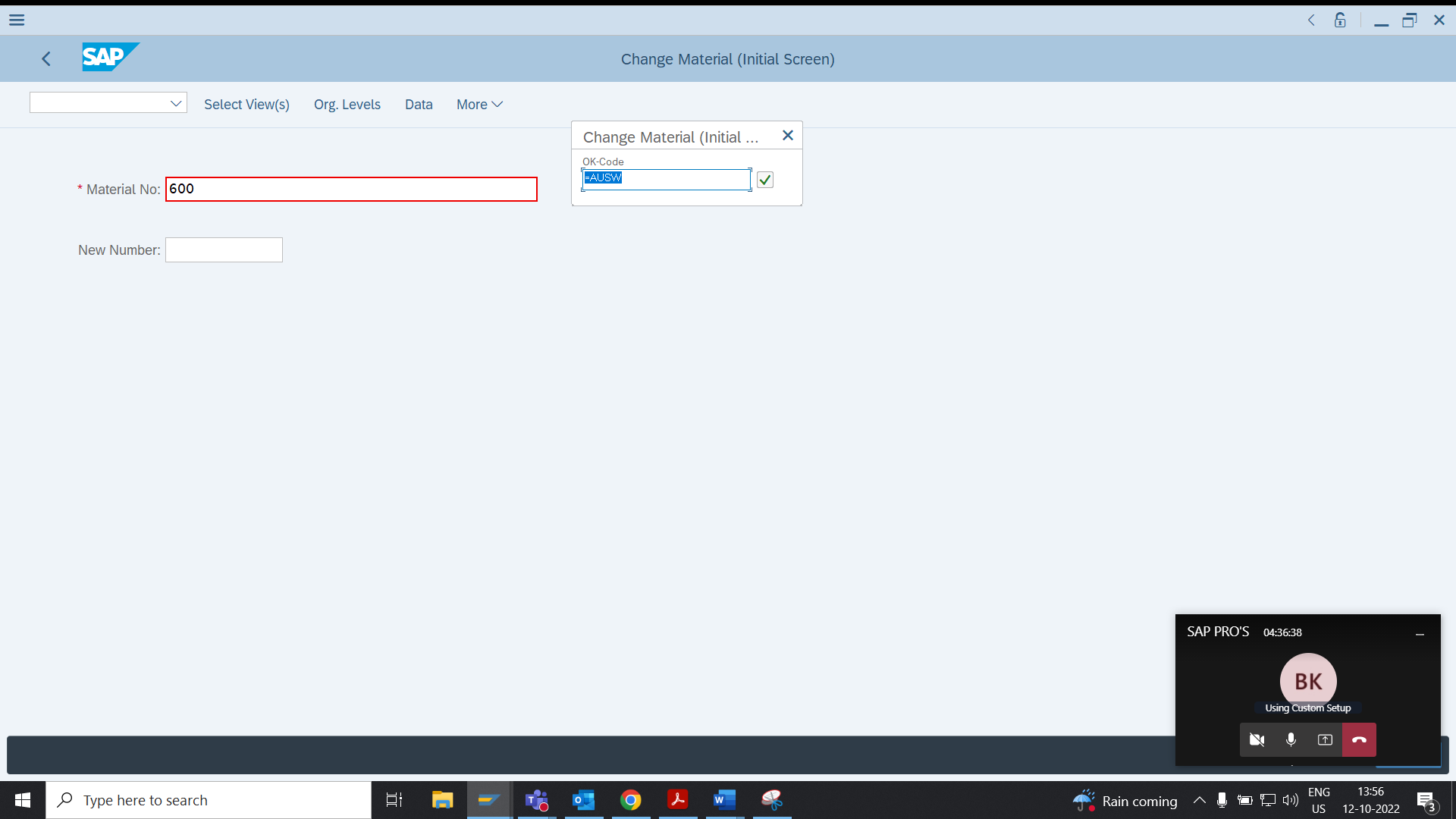


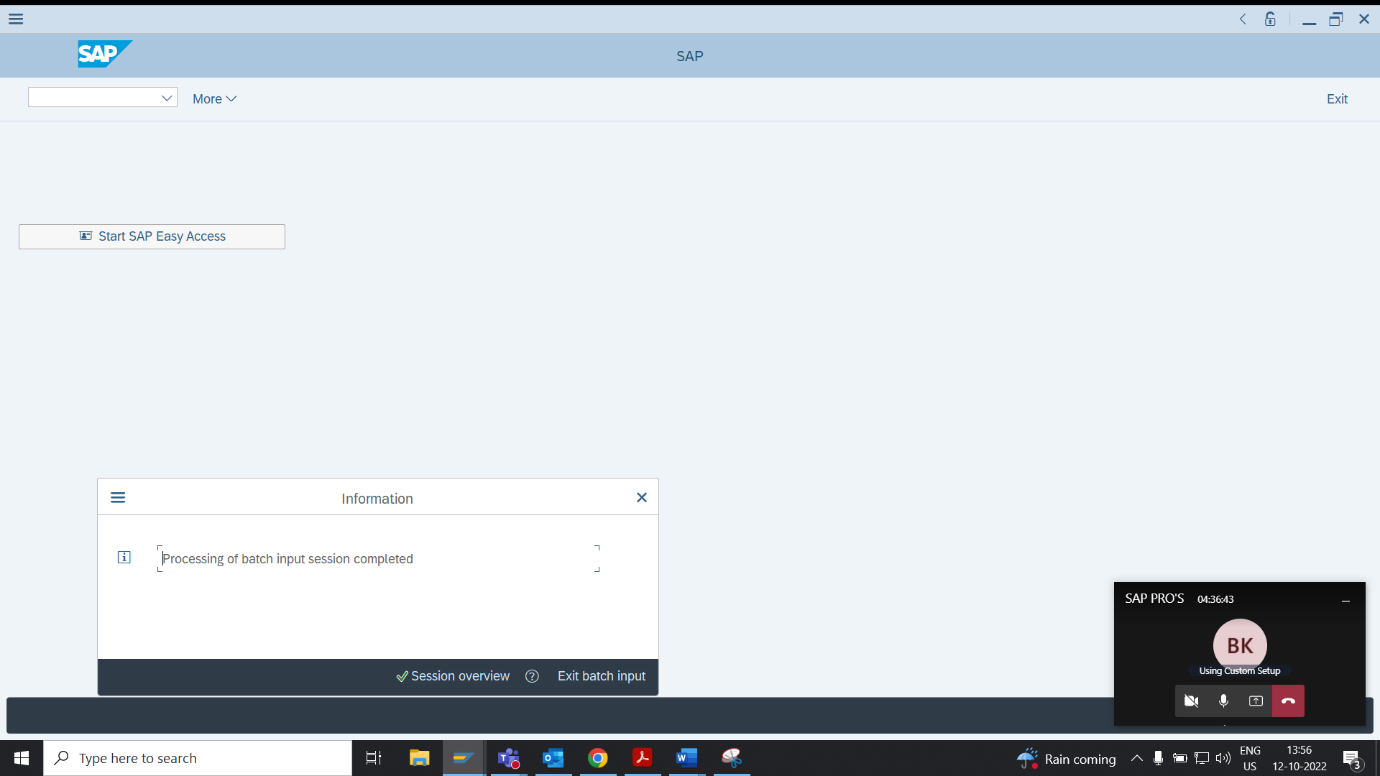
* Now program successfully transferred the data through two recorded screens of FI01 transaction code .









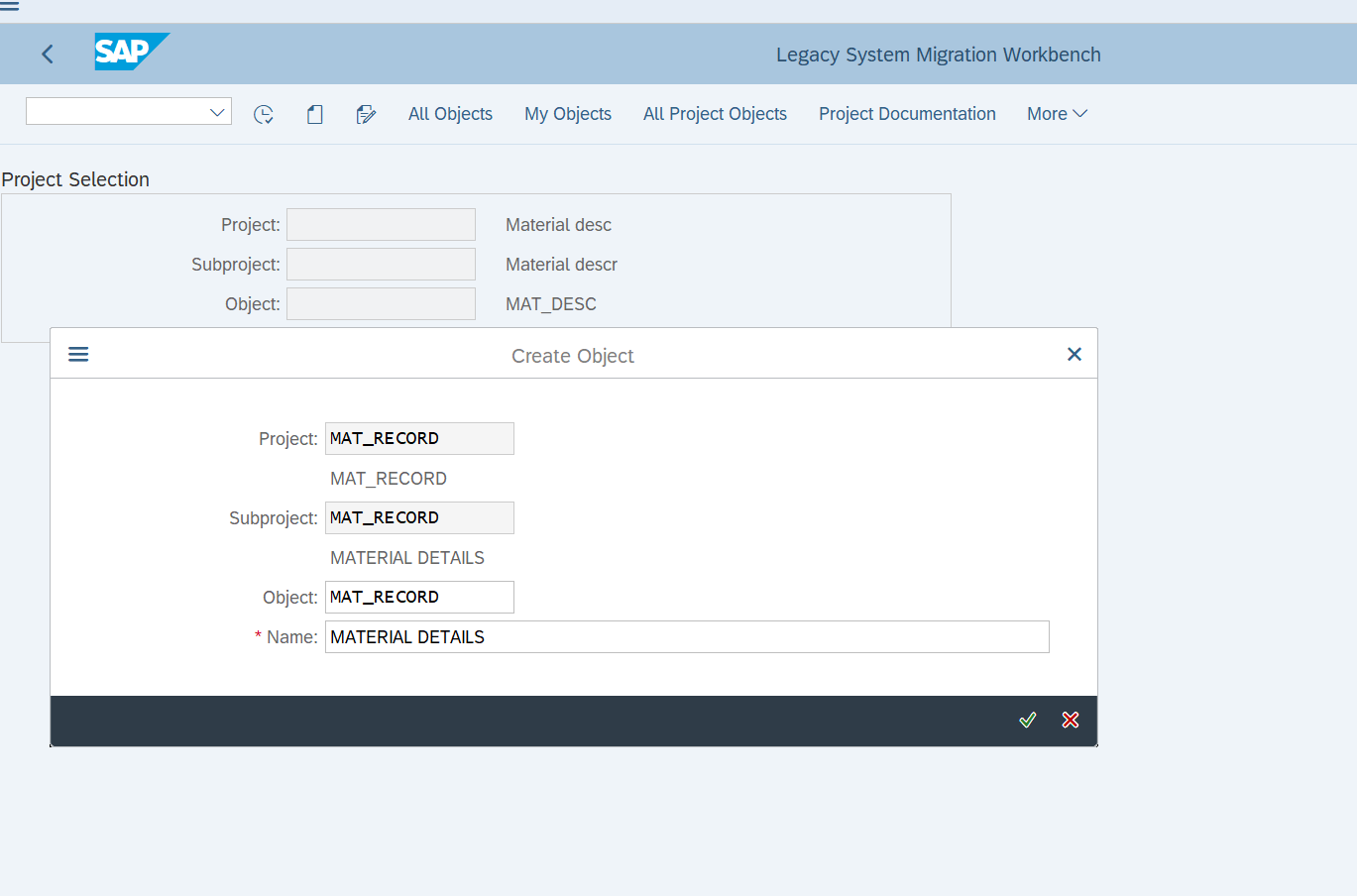


* Now program successfully transferred the data through two recorded screens of FI01 transaction code .

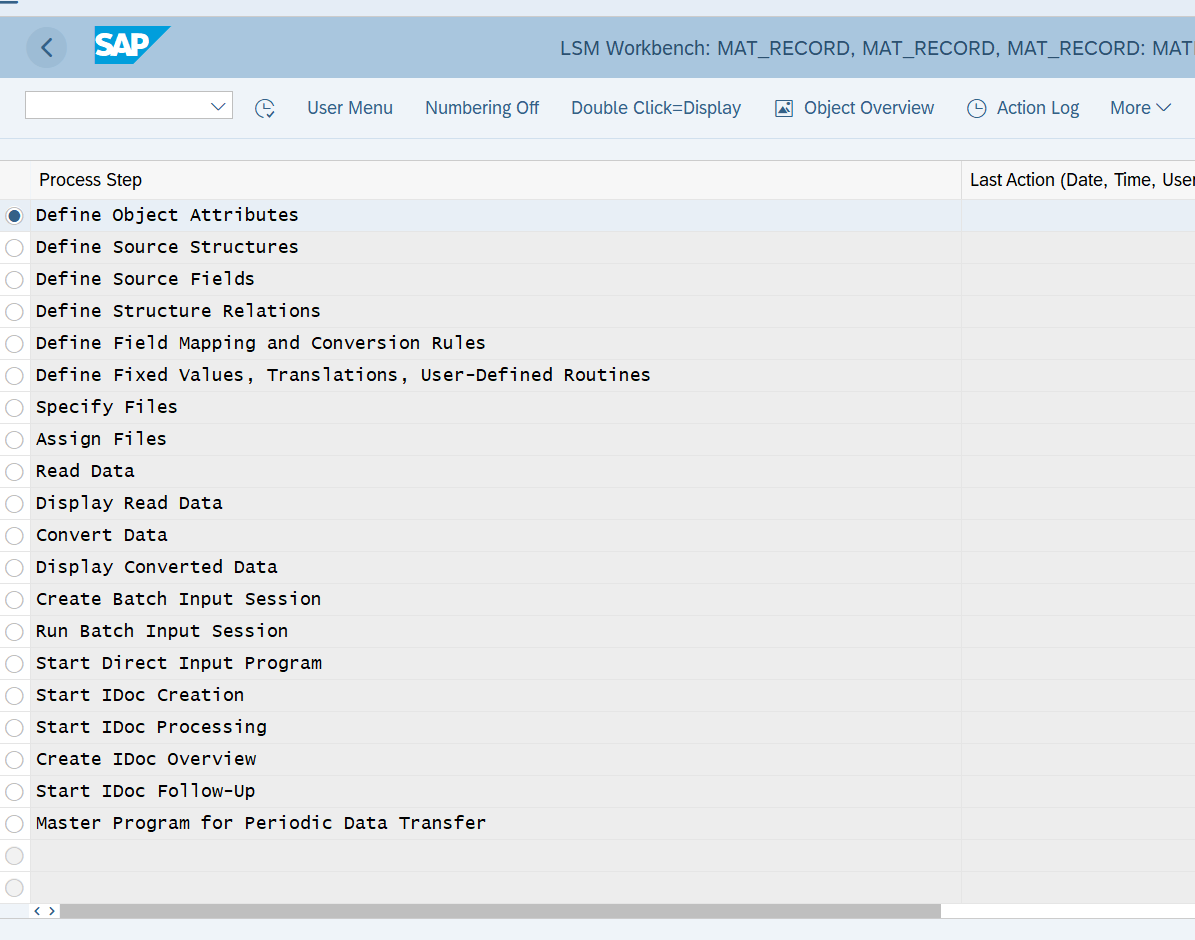
**Q3.** **Create a LSMW to update data~~.~~ Try LSMW with MM01, MM02, VA01 etc.**

**STEP 1:** GIVE LSMW TCODE IN COMMAND FIEELD.

* Click on the create button to create new project, subproject and the object.



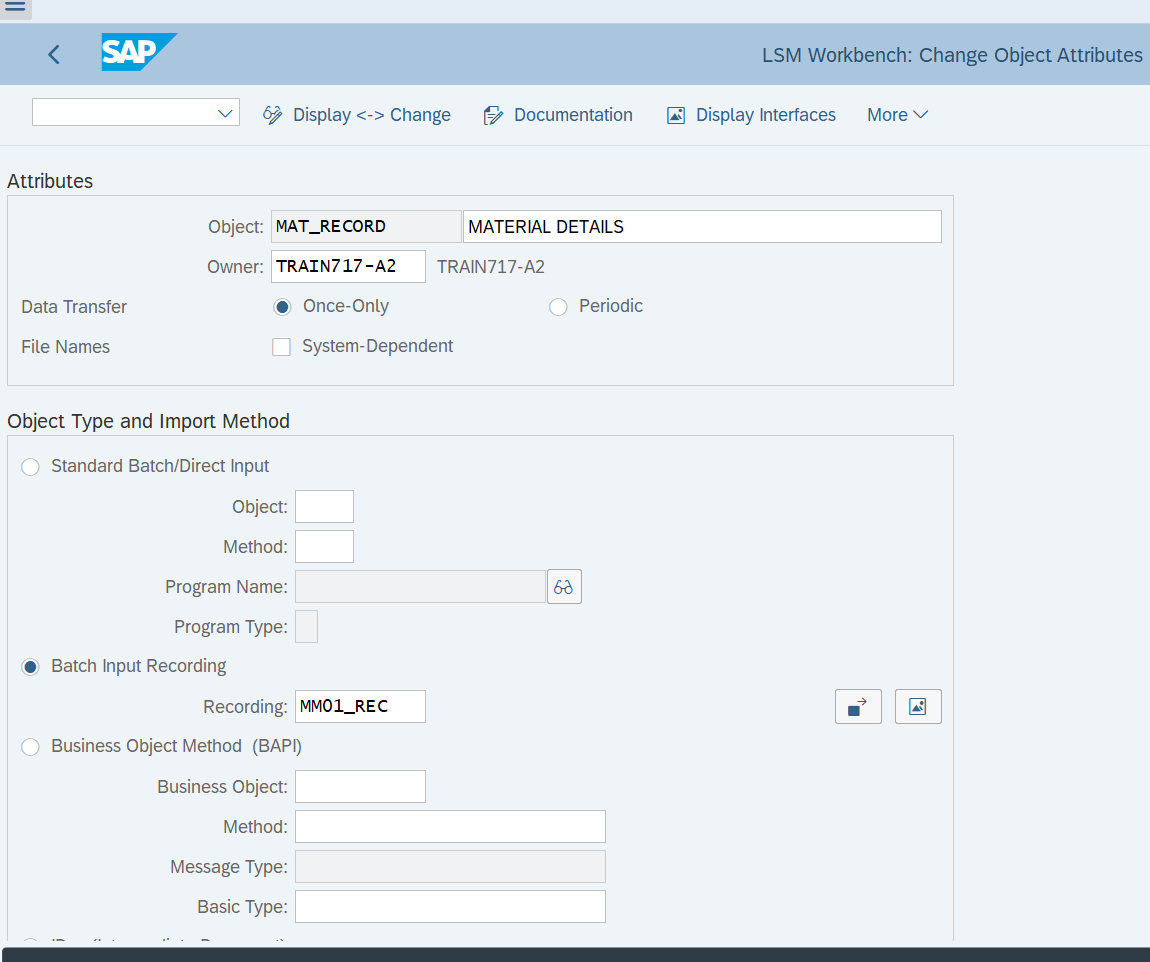
* Click on the execute button once the project, subproject and the object are created.



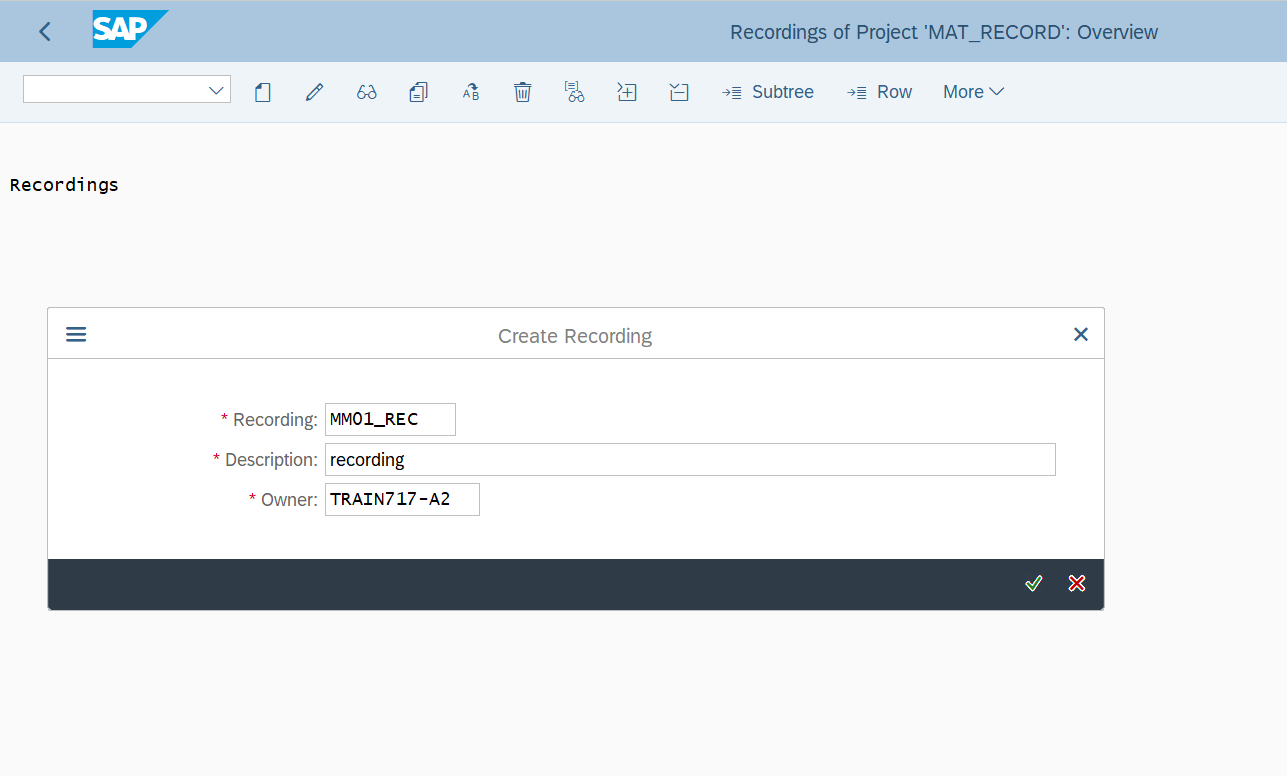
* Double click on Define Object Attributes.

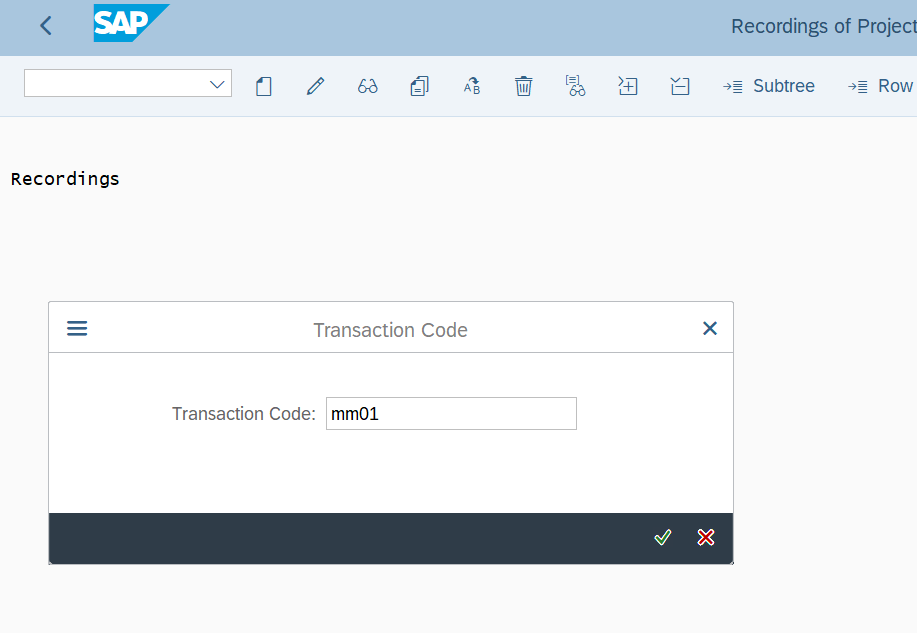
Screen will shown

* Click on Batch input Recording and give record name.

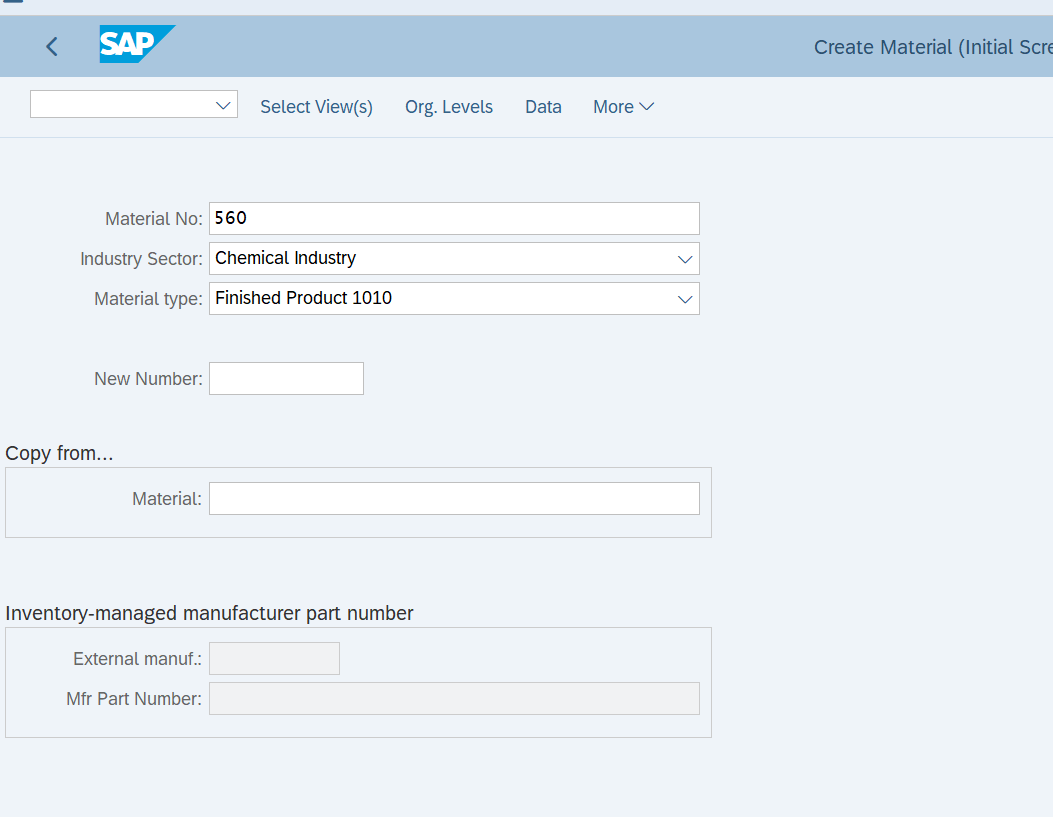


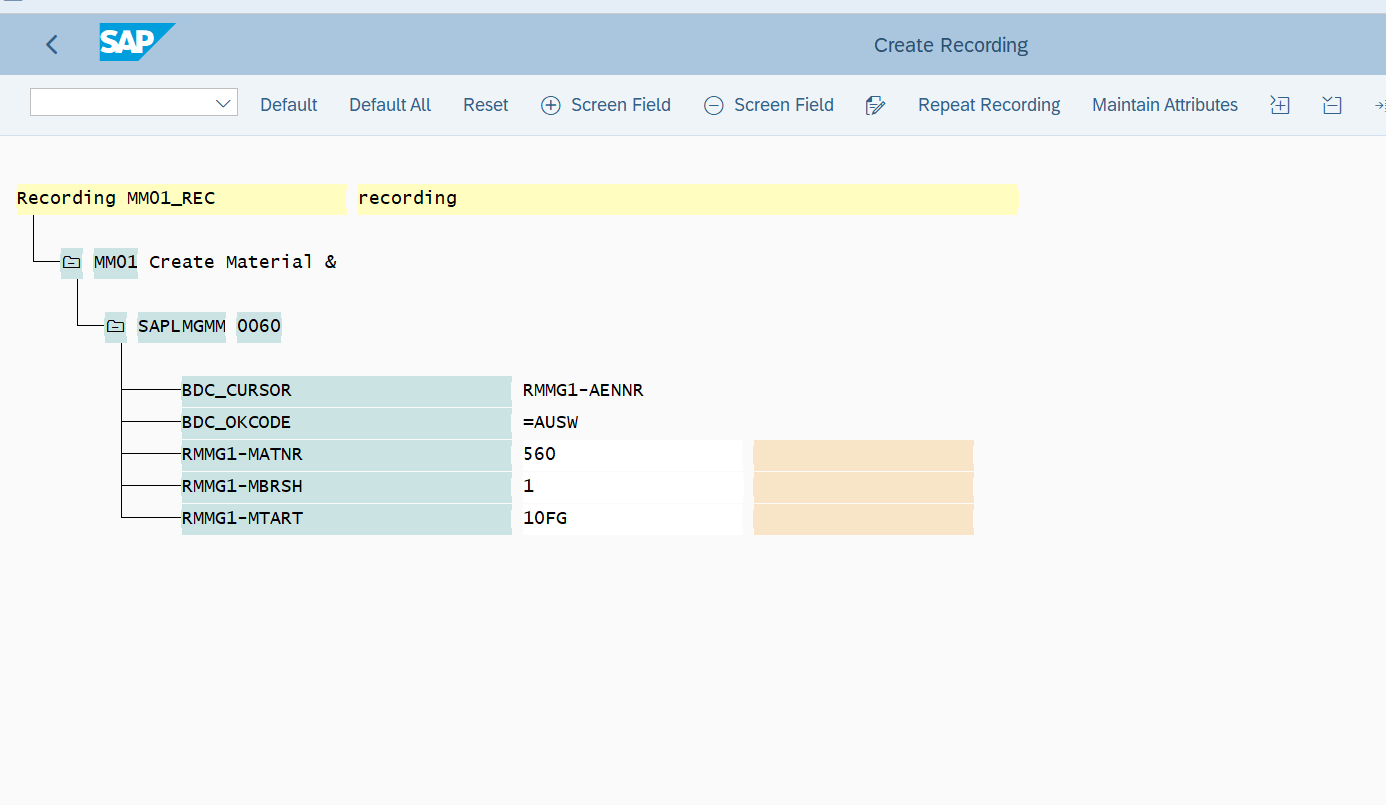
* Click on create to record the transaction and create a record.



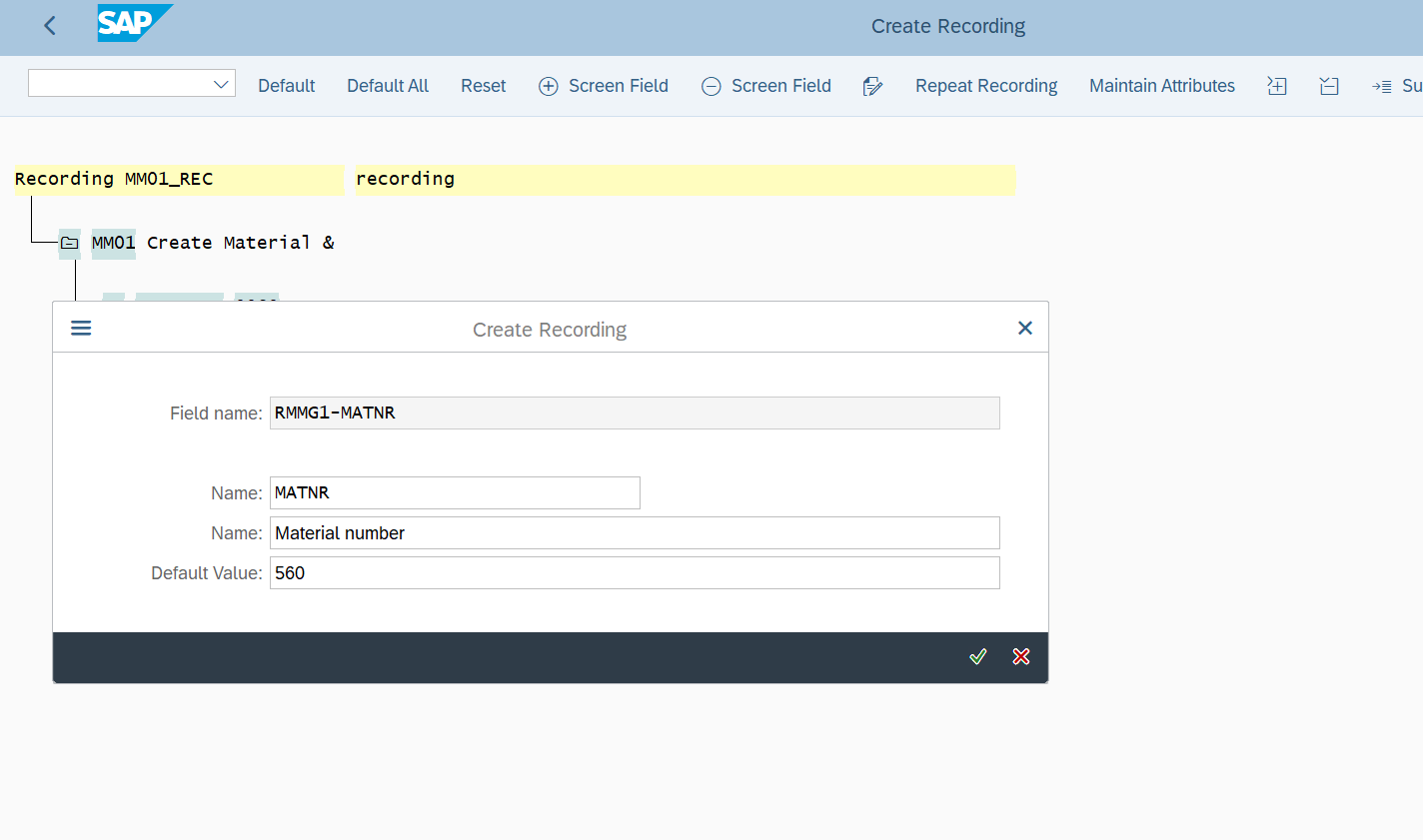


* Click on Done to actually start recording the transaction.
* Enter the fields required for recoding a transaction successfully.

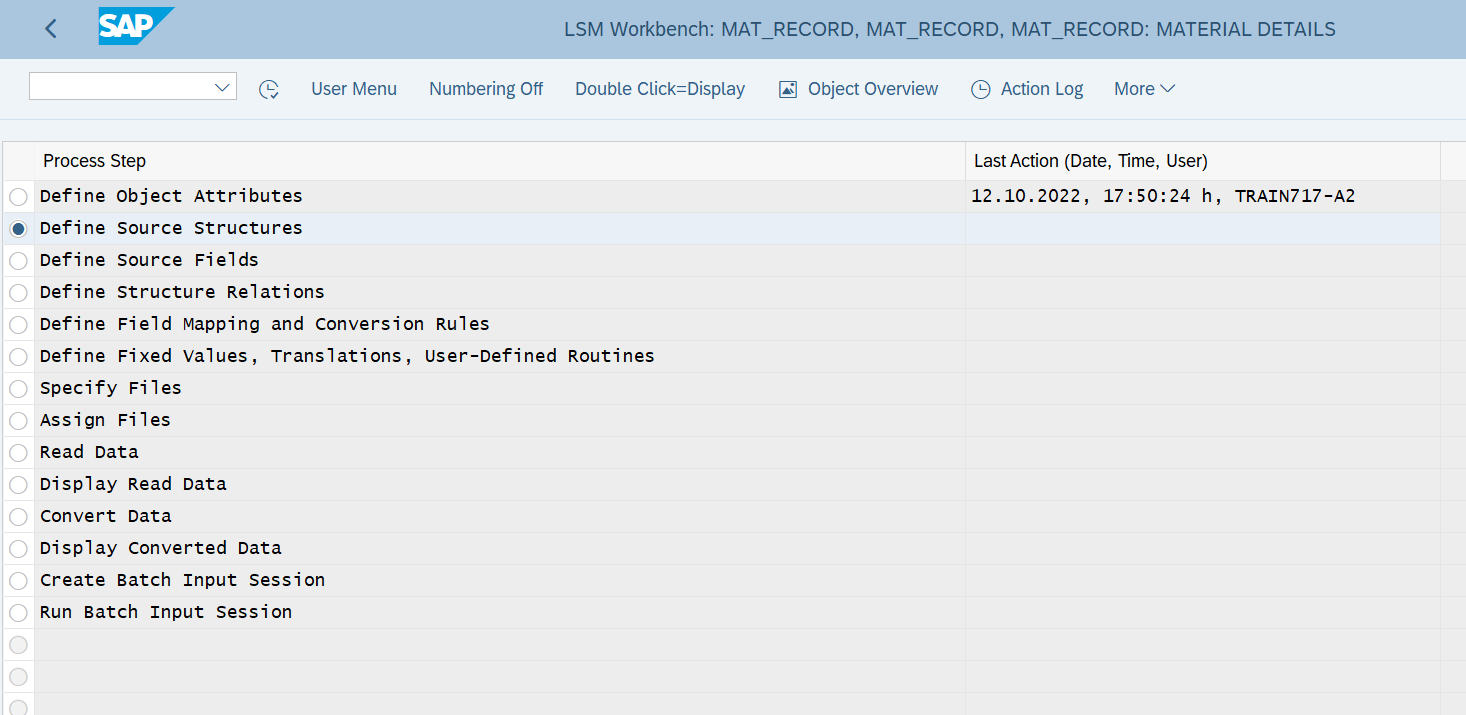




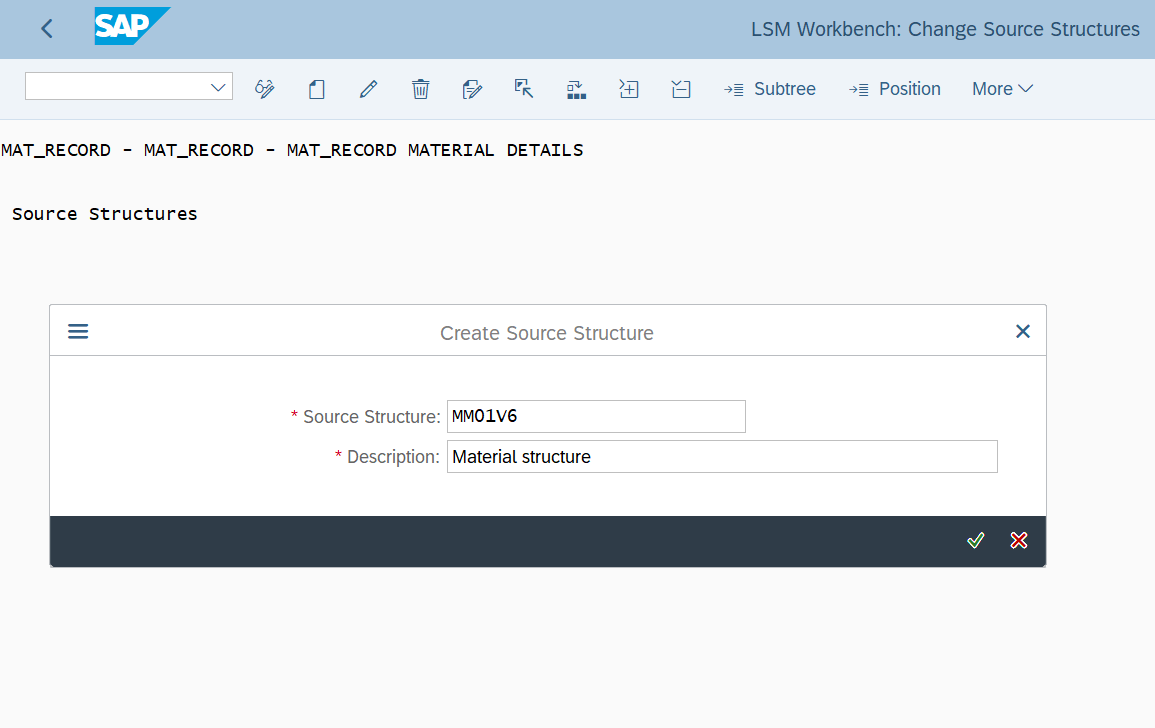
* Double-click on the field RMMG1-MATNRand enter the name as MATNR and the description as MATERIAL NUMBER and remove the default as shown below:



* Click on SAVE to save the recording. The click on BACK icon to the main screen. Save the while going back to the main screen.
* After completing the recoding the system will automatically take you to the second step as shown below:

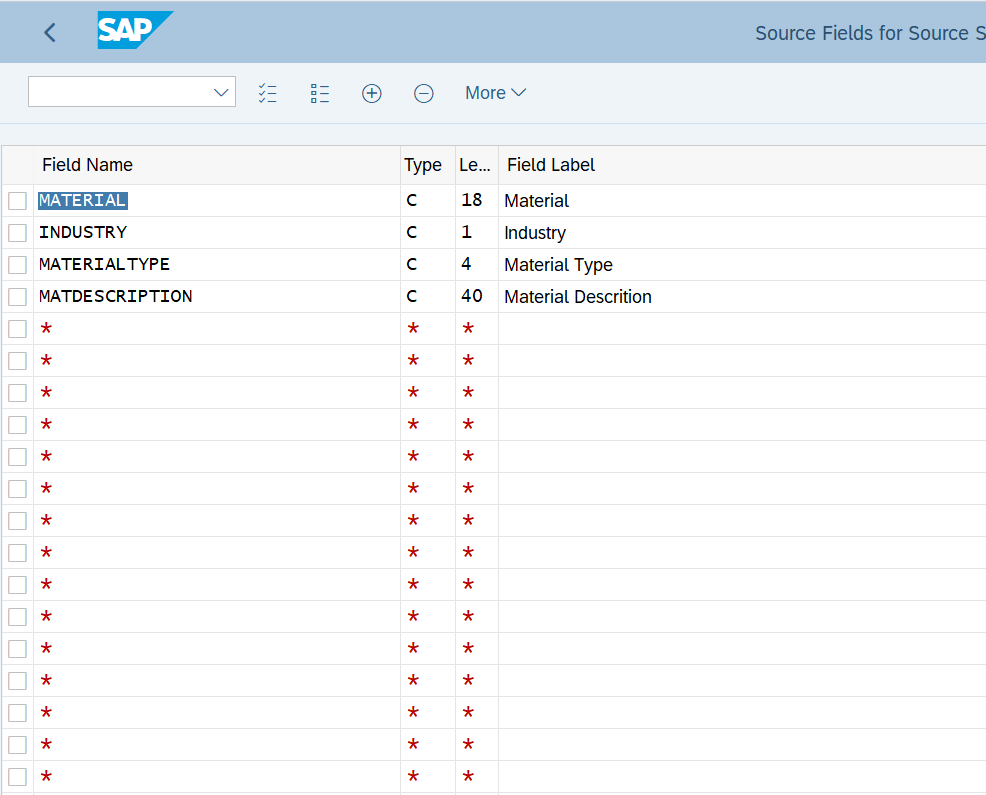


**Step 2.** Define Source Structures  
Click on CREATE to create a source structure. Give a name and a description to the source structure as shown below:



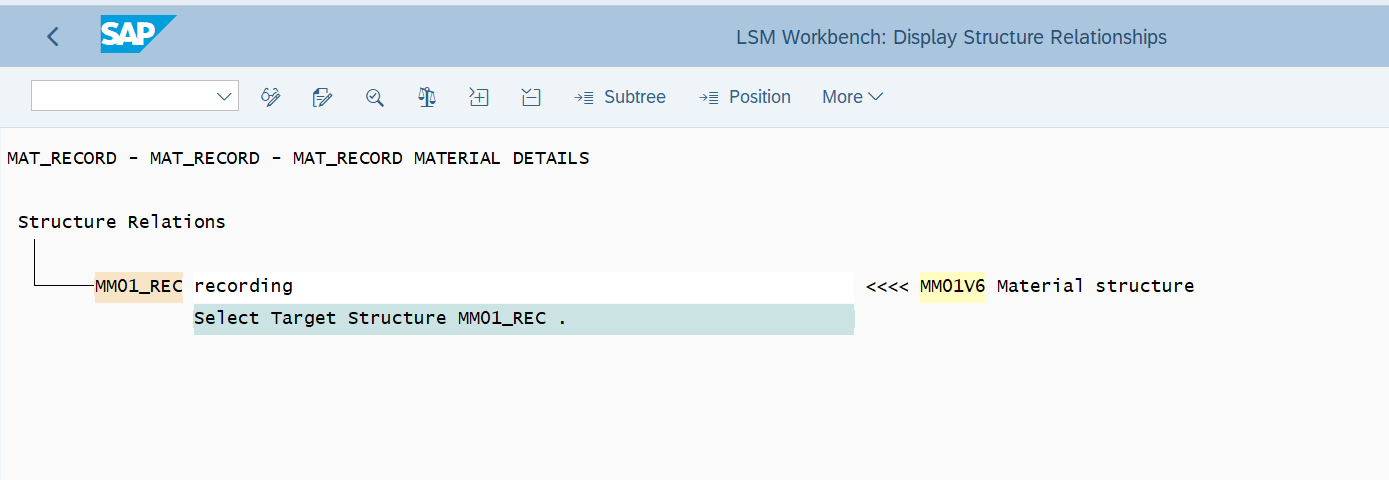
* Save the source structure and go to the main screen.

**Step 3.** Maintain Source Fields In this step, you need to list what fields are present in the source structure.



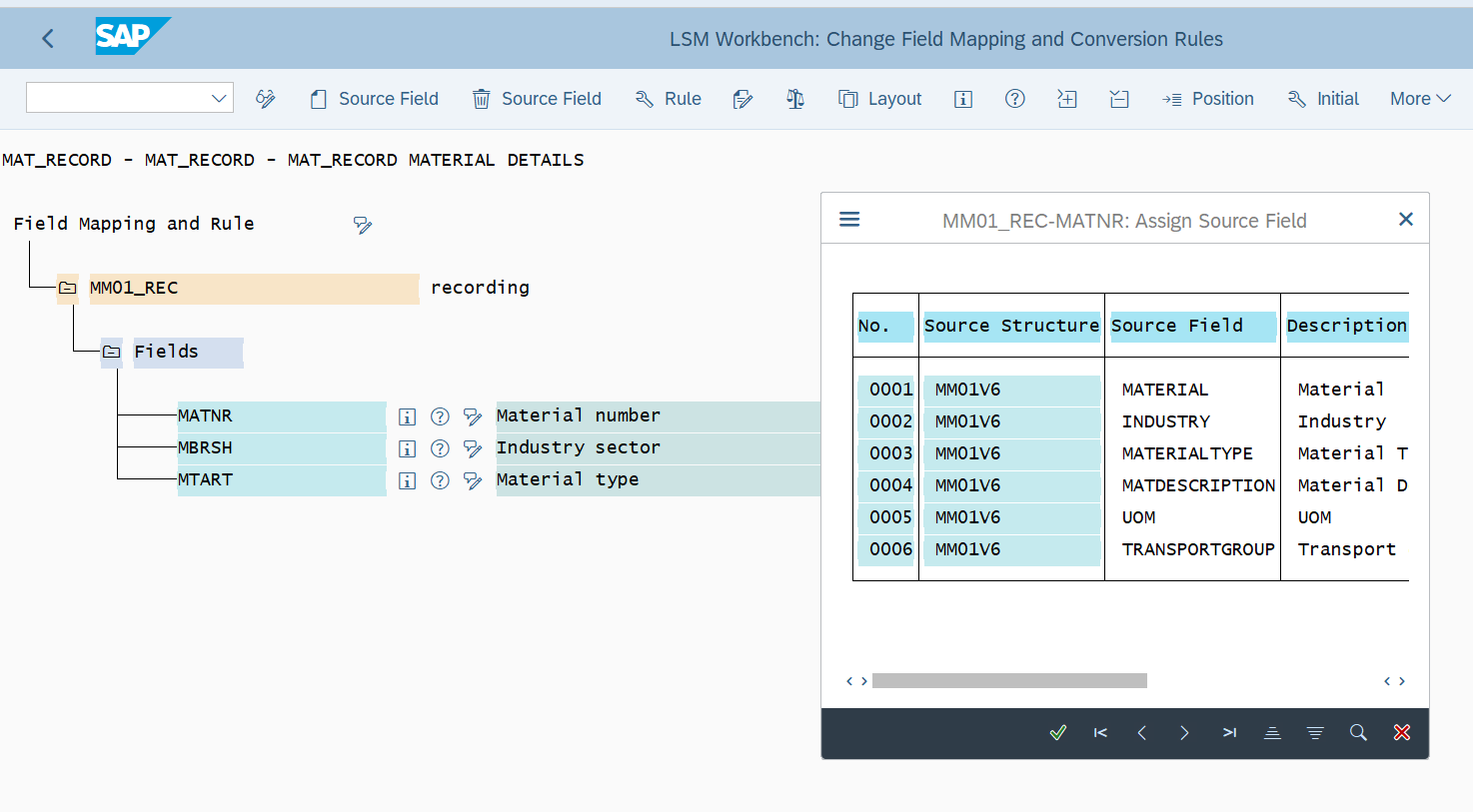
Save and come back to the screen.

**Step 4:** Maintain Structure Relations  
Execute a step to 'Maintain Structure Relations'. Since, there is only one Source and Target Structure, the relationship is defaulted automatically.

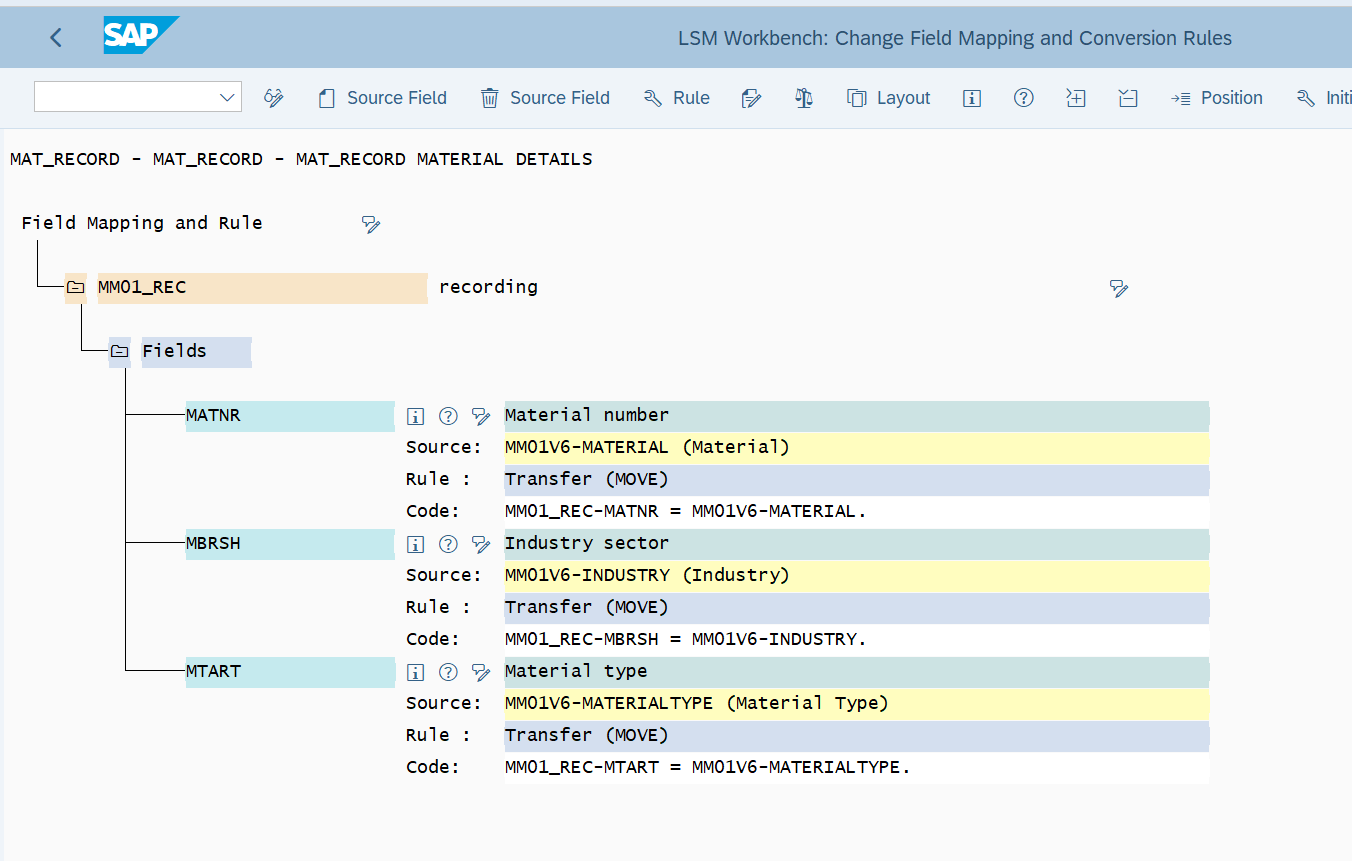


Save while coming back to the main screen.

**Step 5:** Maintain field mapping and conversion rules  
Keep cursor on field 'MATNR' and click on 'Assign Source field' icon to choose source field MATERIAL from structure MM01V6 as shown

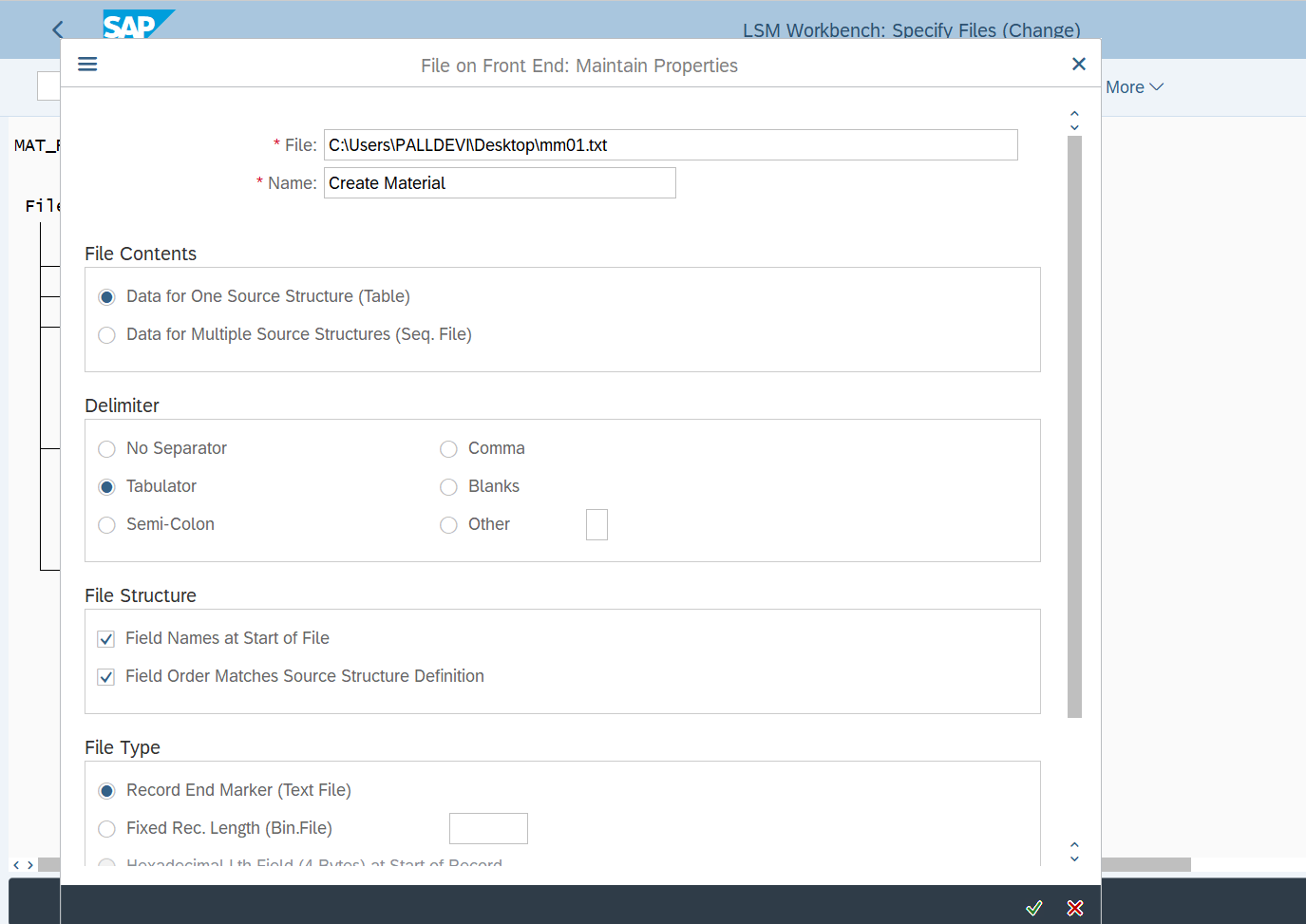


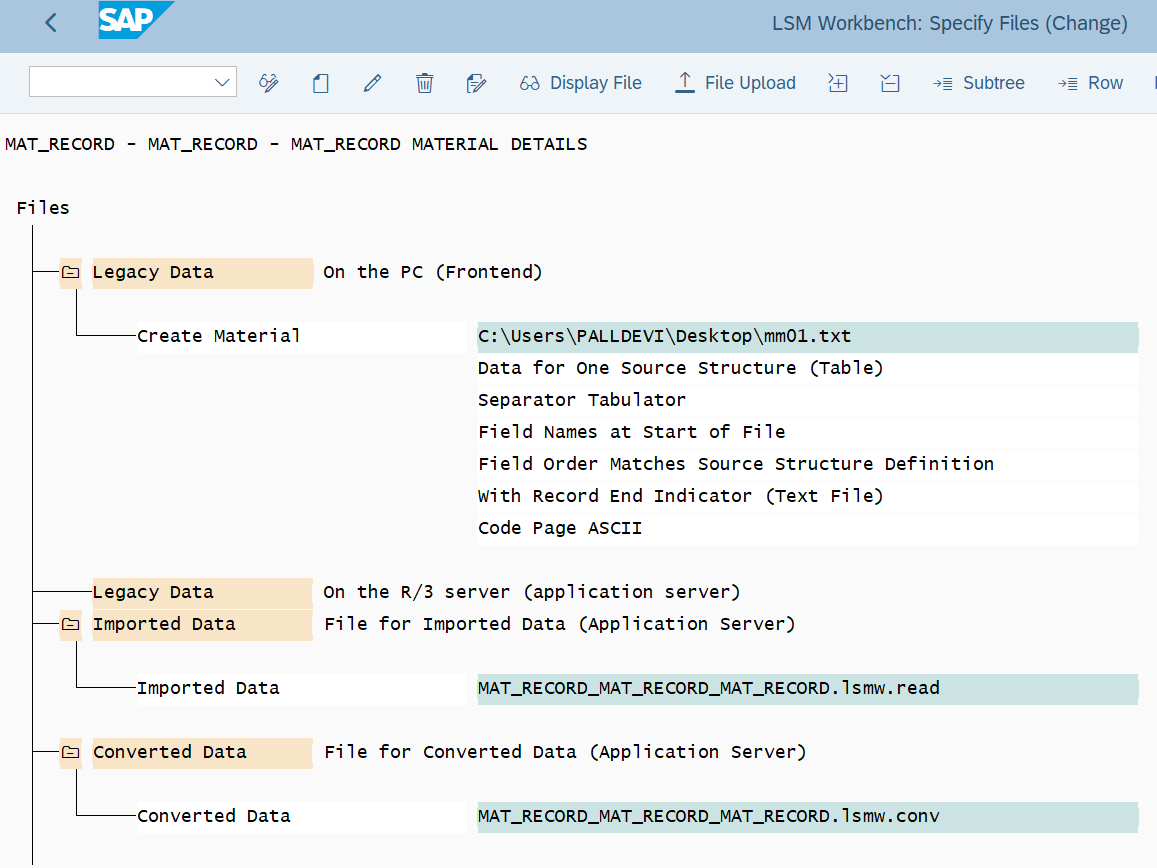
* Once all the fields are mapped, you should have an overview screen as shown.



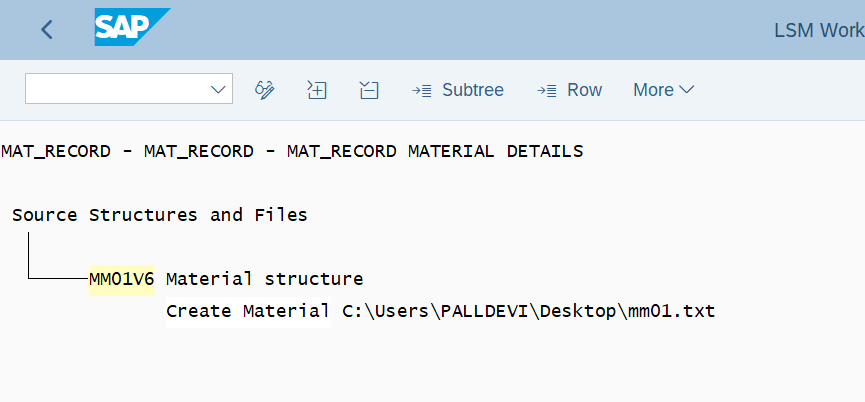
* Save while coming back to the main screen.

**Step 6: Specify files** In this step, we define how the layout of the input file is.



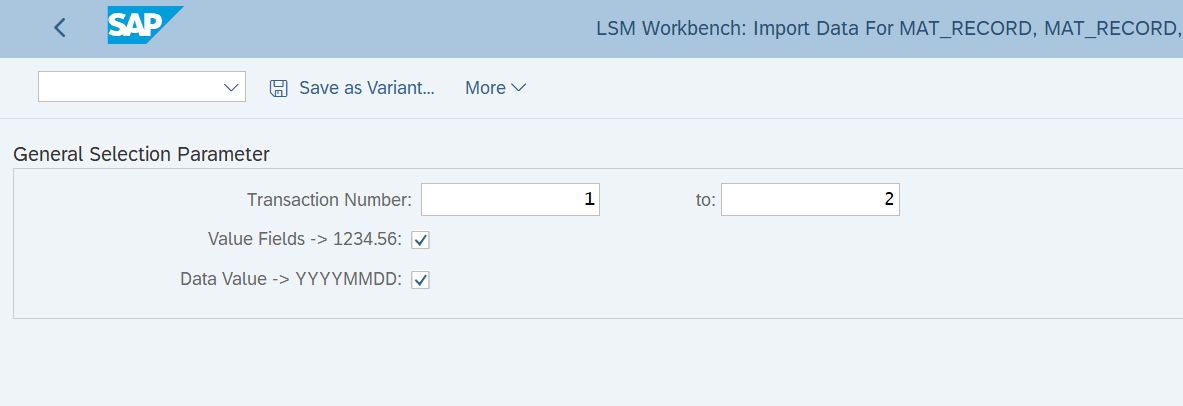


**Step 8:** Assign files Execute step 'Assign Files' and the system automatically defaults the filename to the source structure.

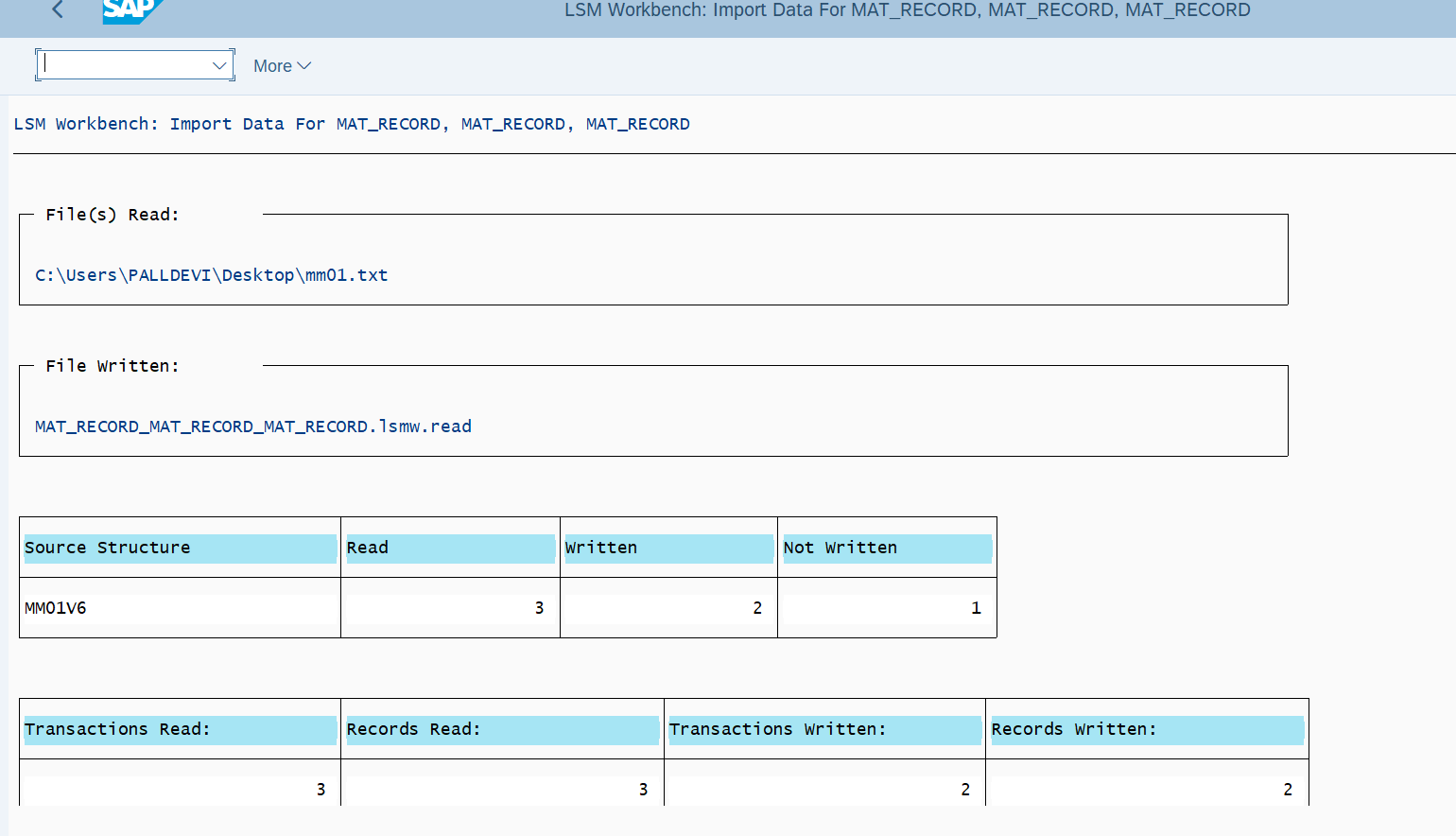


Save while going to main screen.

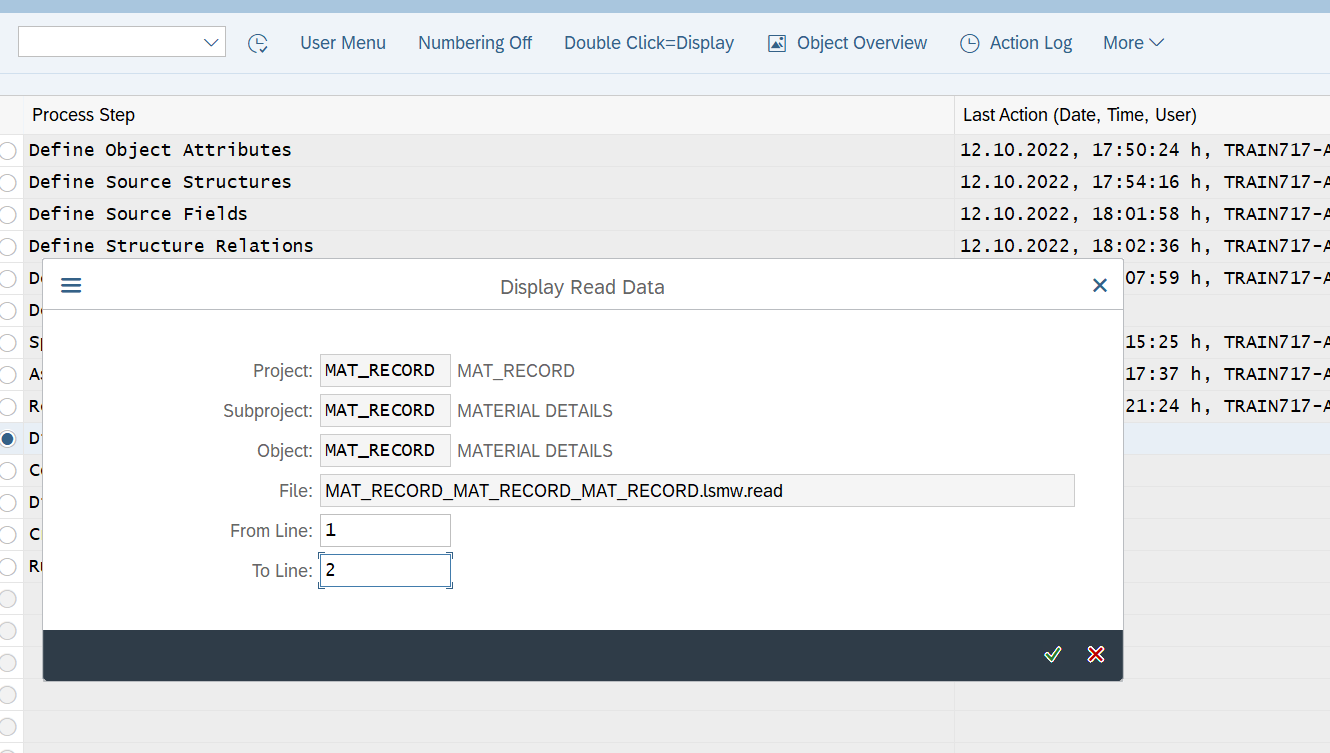
**Step 9: Read data**  
In this step, LSMW reads the data from the source file .

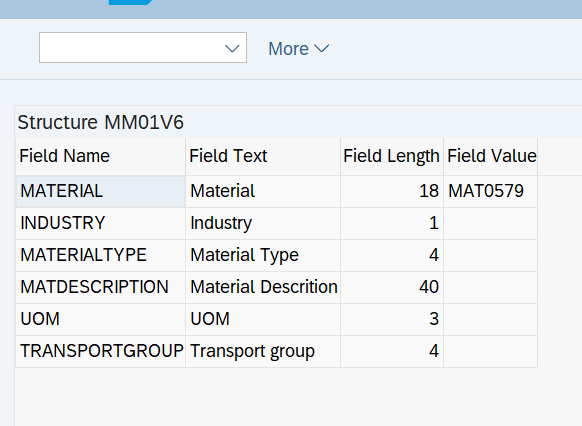


* After we execute the data read from the flat file is as shown below.



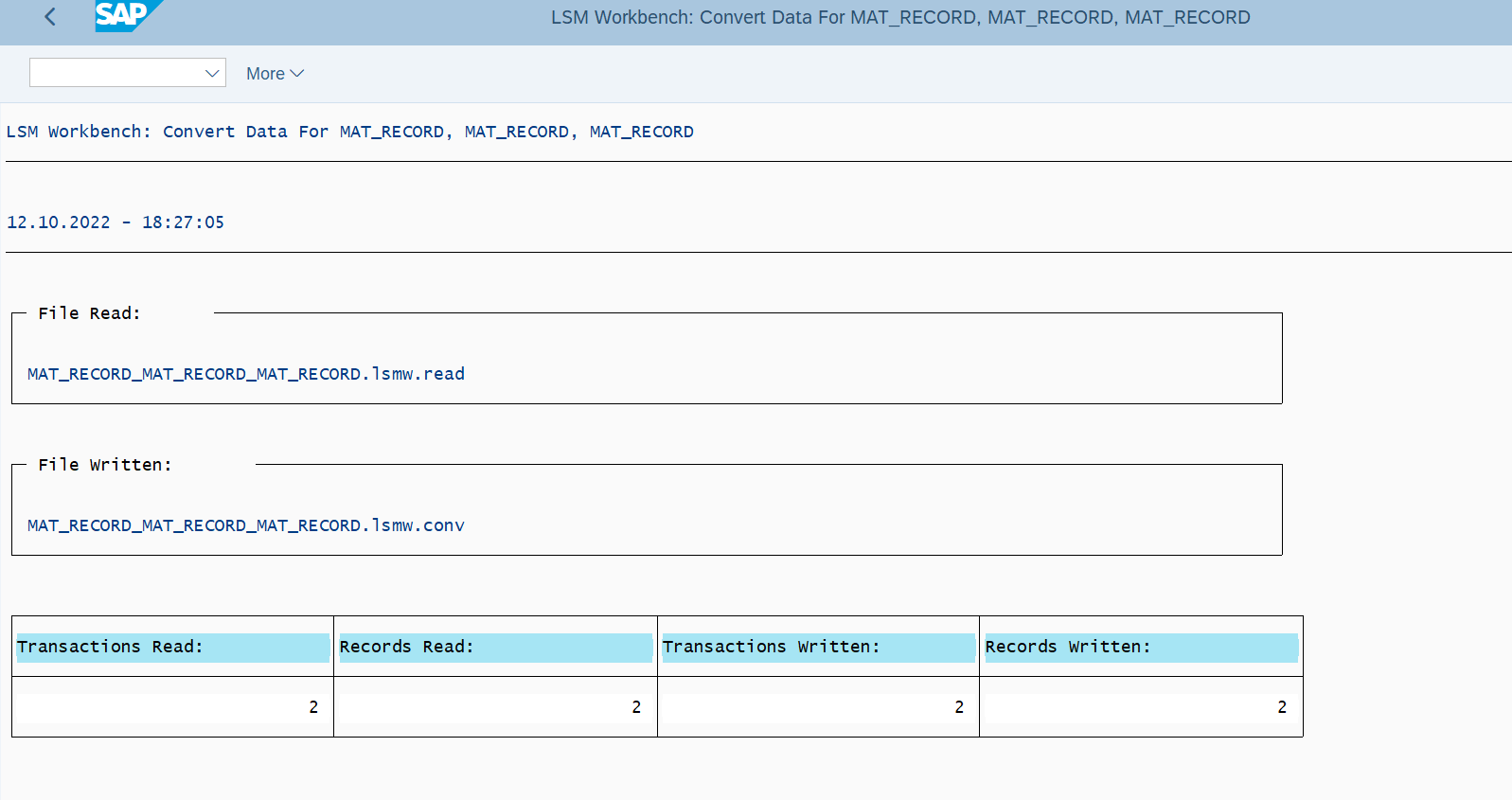
**Step 10: Display read data**,in this step you can review the field contents for the rows of data read



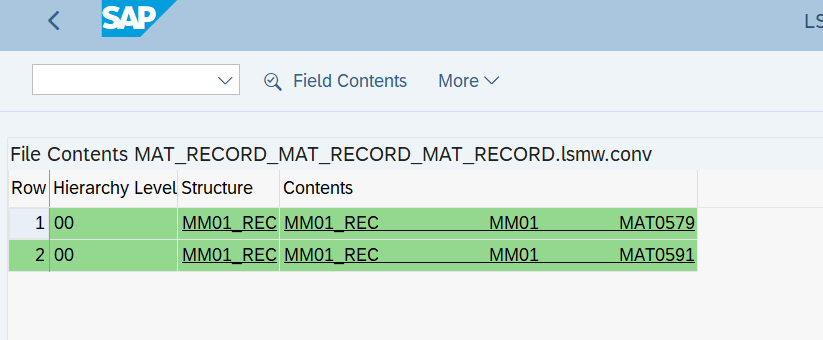


**Step 11: Convert data**

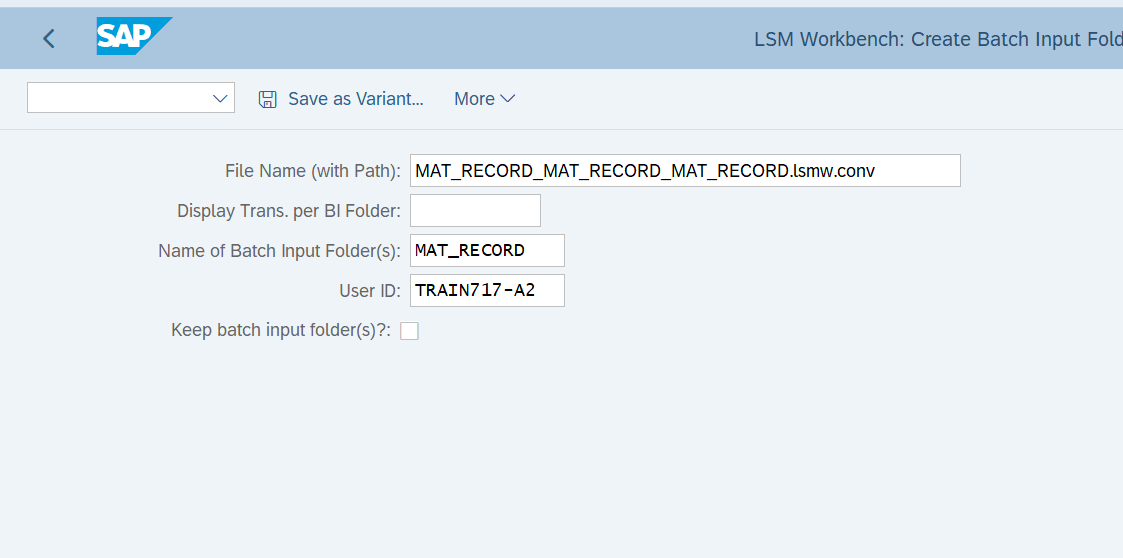
This is the step that actually converts the source data (in source format) to a target format.



**Step 12: Display Converted data**



**Step 13: Create batch input session**  
Once the source data is converted in an internal format, you can create a batch session to process updates.



**Step 14: Run Batch Input Session**

* You can execute the BDC session by **Run Batch input session.**
* Select the Processing Mode and then click on the PROCESS tab to executive the session.

