**LIT TRAINING**

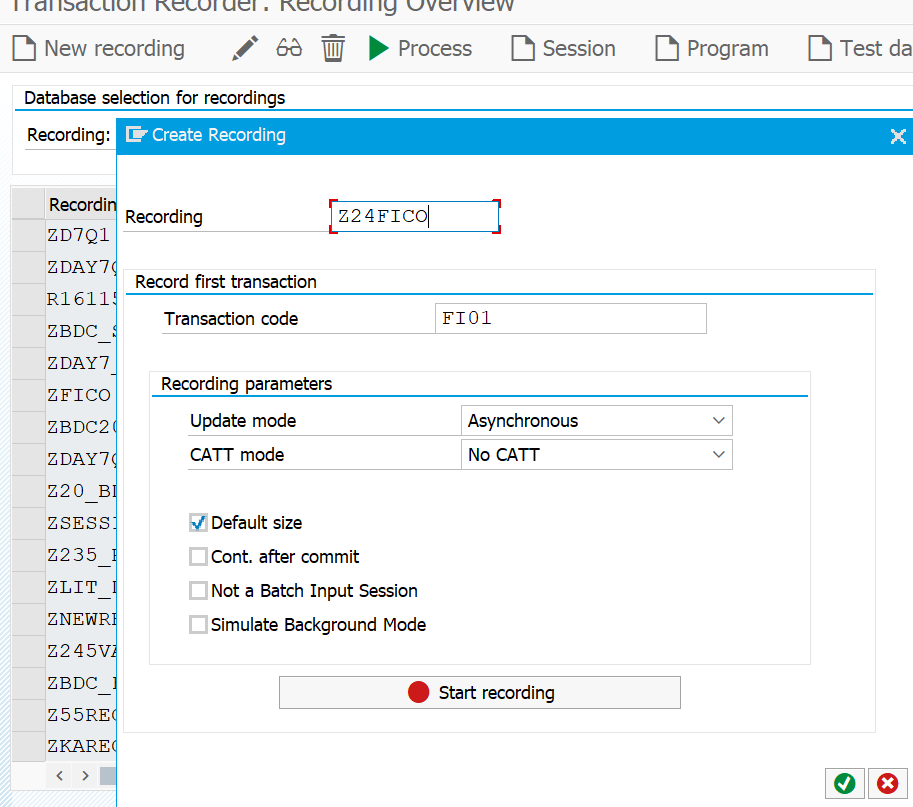
**Batch Name : SAP ABAP ON HANA** **DAY7 ASSIGNMENT**

**EMPLOYEE ID – 46255244** **DATE : 09th September, 2022**

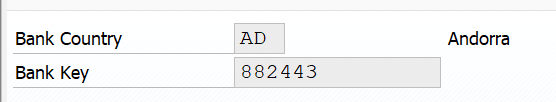
**1-Q Create an Upload program Using BDC Call Transacion to update data in VA01(Sales Order) .**

* **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**

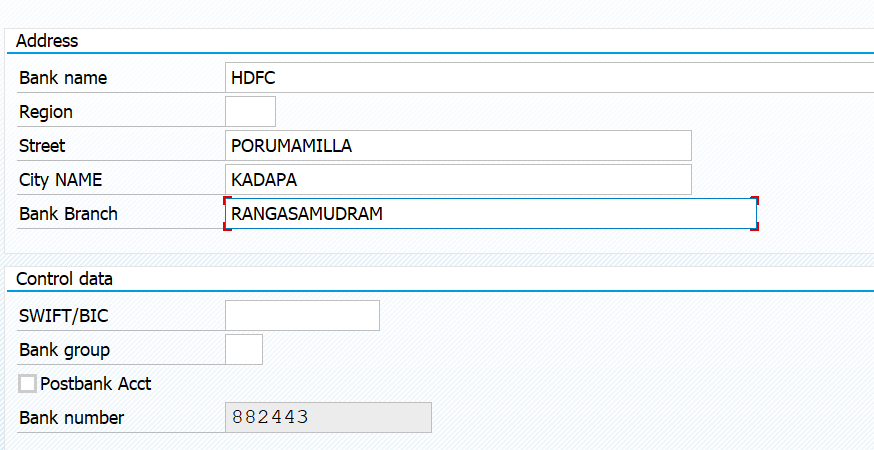
**STEP 1: CREATE A NEW RECORDING IN SHDB.**

****

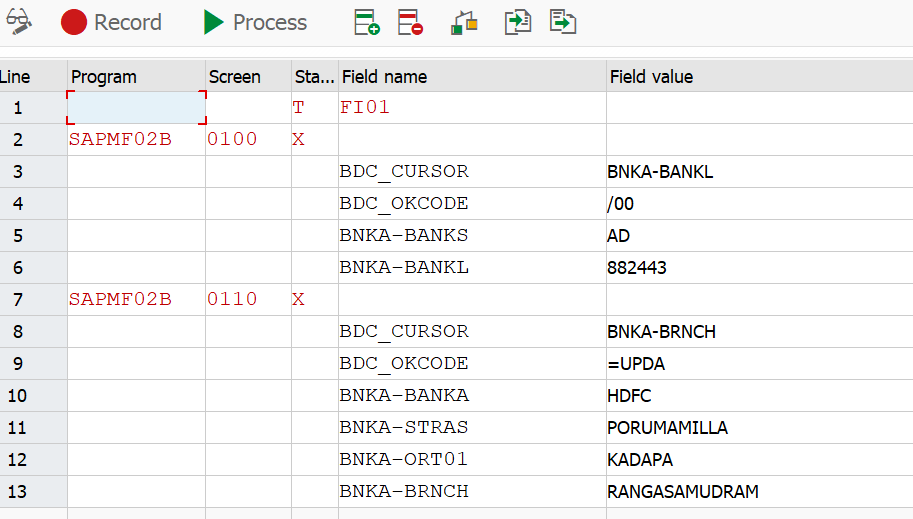
* Enter the Bank country and Key

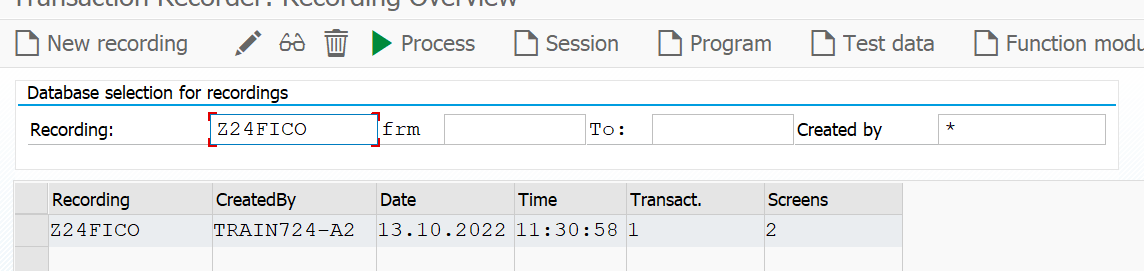
****

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

****

Now Save the Recording.

****

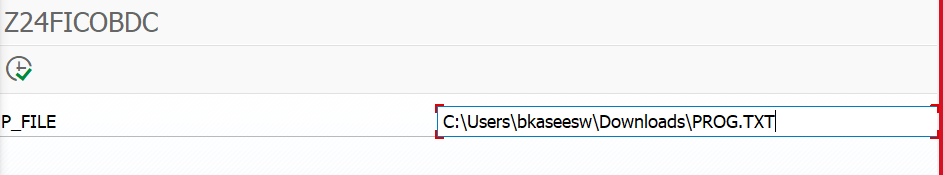
****

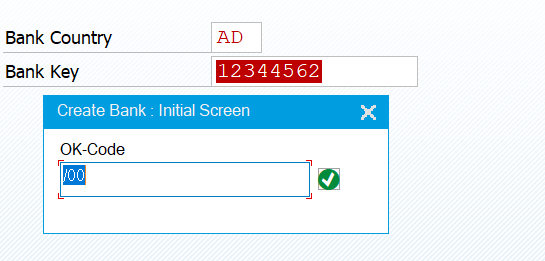
* Now select the created record and created a Program.
* Now BDC program will be created automatically and you can find the structures and dynpro and dynfield subroutines in Include BDCRECX1 main program.Copy them and paste in your source code.

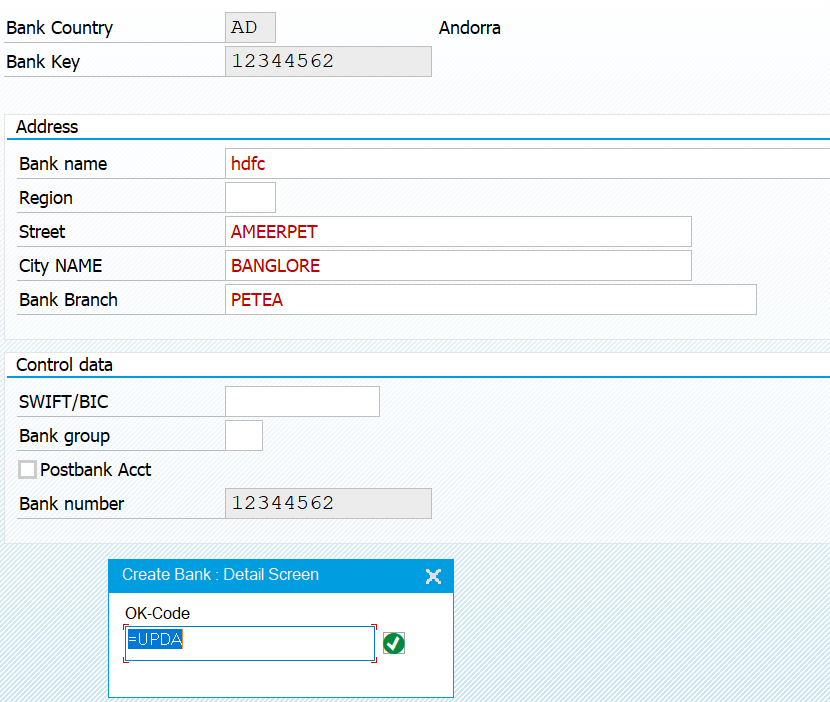
**REPORT:**

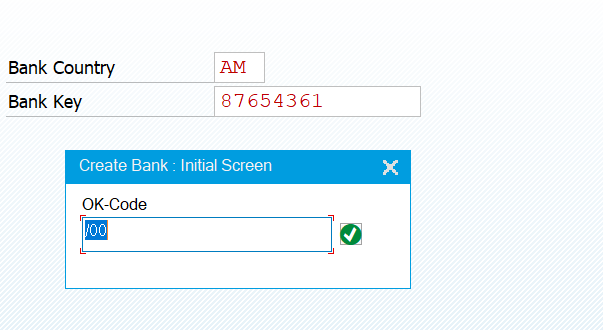
\*&---------------------------------------------------------------------\*  
\*& Report Z5236\_FICO\_BDC  
\*&---------------------------------------------------------------------\*  
\* Description:                  BDC Call Transacion to update data in FI01     
\*                                                                      \*  
\* Author:              BOLLA KASEESWARI                                
\*                                                                      \*  
\* Create date:          12/10/2022                                                                                                                  
REPORT Z24FICOBDC.  
  
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.  
\*                              record-BANKS\_001.  
perform bdc\_field       using 'BNKA-BANKL' WA\_BNKA-BANKL.  
\*                              record-BANKL\_002.  
perform bdc\_dynpro      using 'SAPMF02B' '0110'.  
perform bdc\_field       using 'BDC\_CURSOR'  
                              'BNKA-BRNCH'.  
perform bdc\_field       using 'BDC\_OKCODE'  
                              '=UPDA'.  
perform bdc\_field       using 'BNKA-BANKA' WA\_BNKA-BANKA.  
\*                              record-BANKA\_003.  
perform bdc\_field       using 'BNKA-STRAS' WA\_BNKA-STRAS.  
\*                              record-STRAS\_004.  
perform bdc\_field       using 'BNKA-ORT01' WA\_BNKA-ORT01.  
\*                              record-ORT01\_005.  
perform bdc\_field       using 'BNKA-BRNCH' wa\_BNKA-BRNCH.  
\*                              record-BRNCH\_006.  
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.

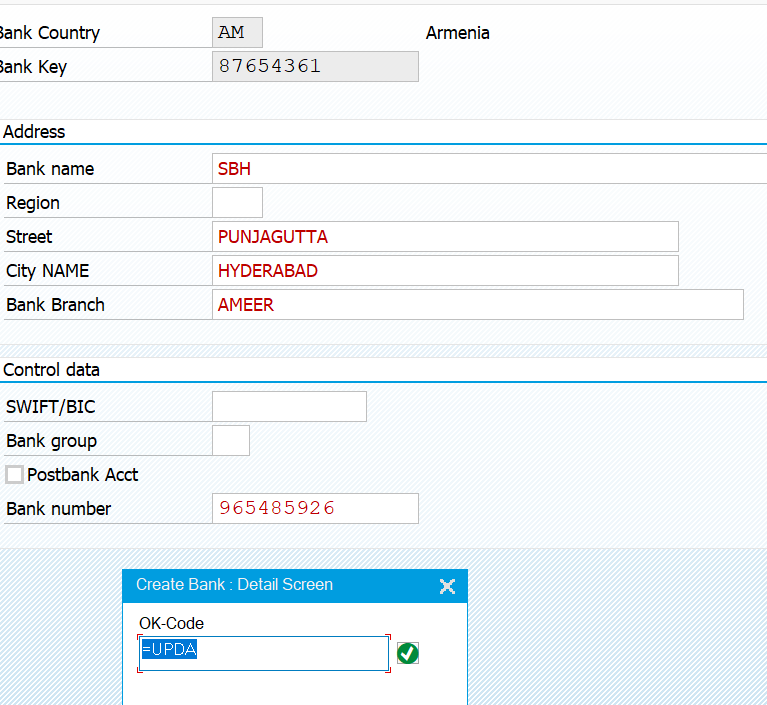
OUTPUT: Read data from the flat file

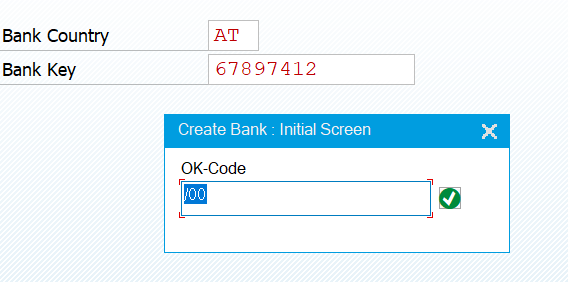


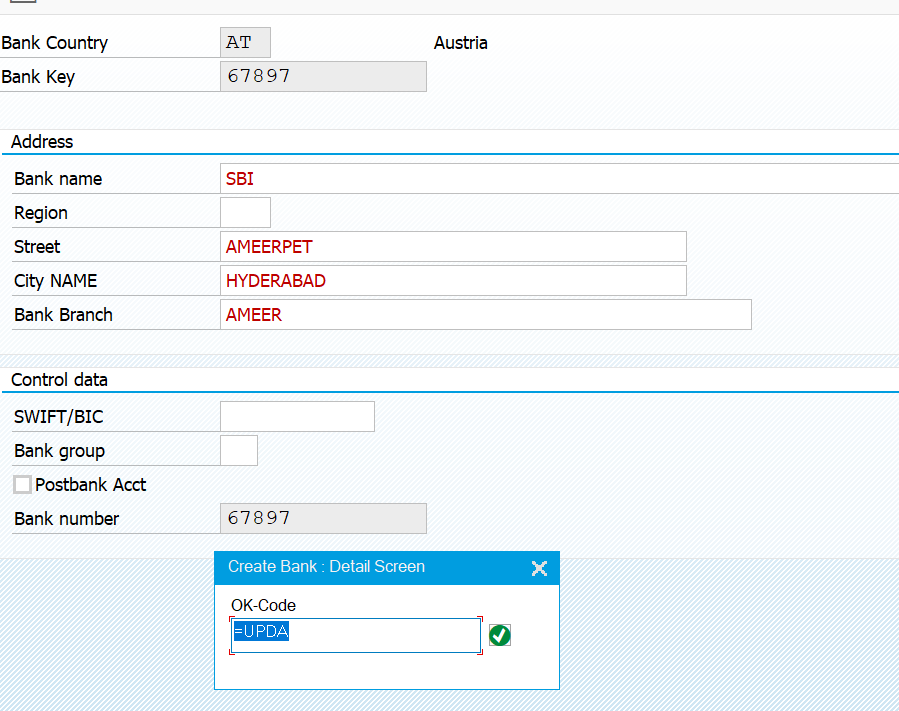


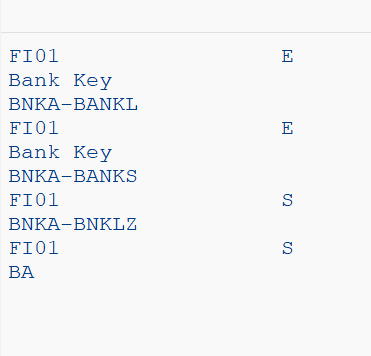






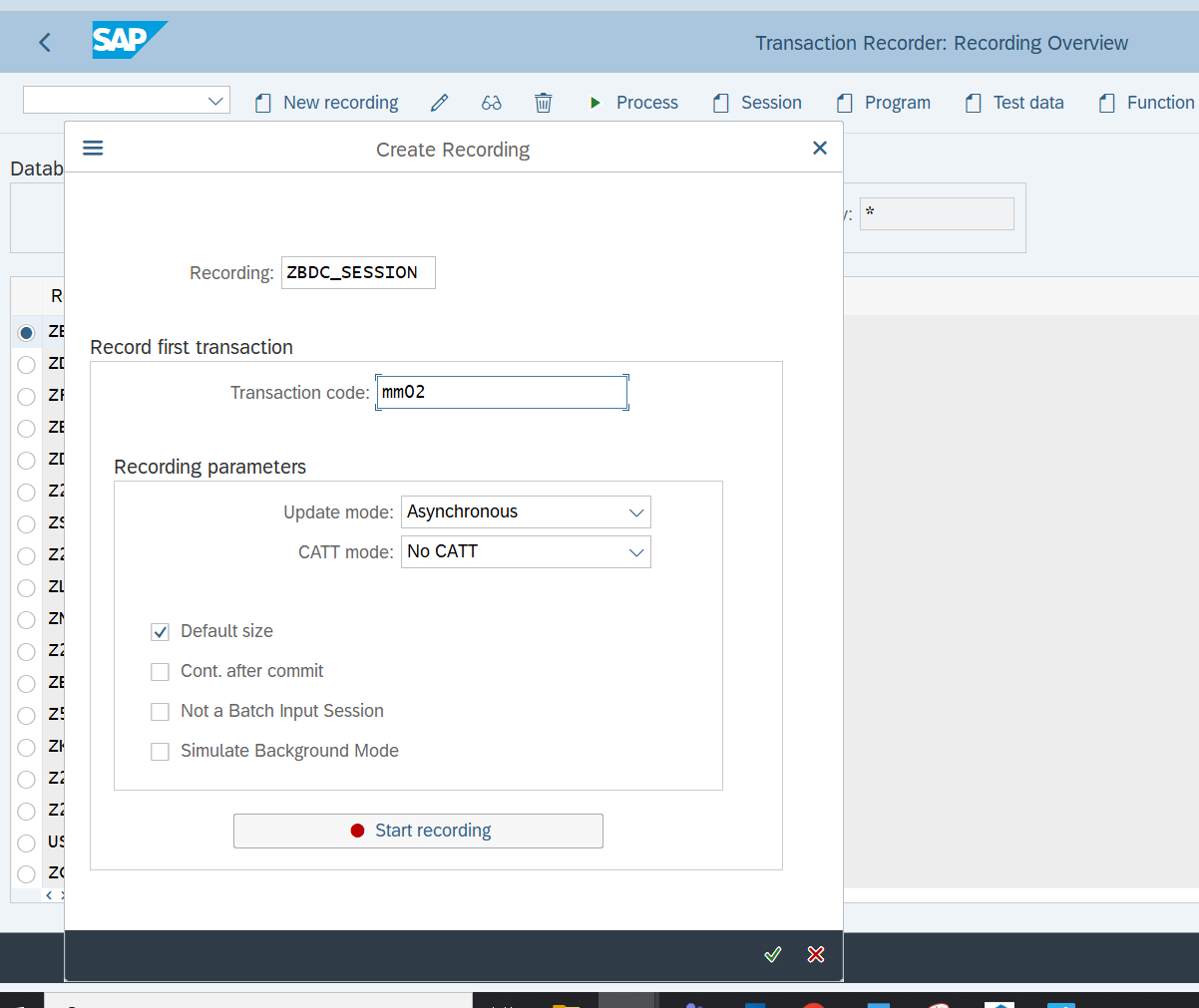




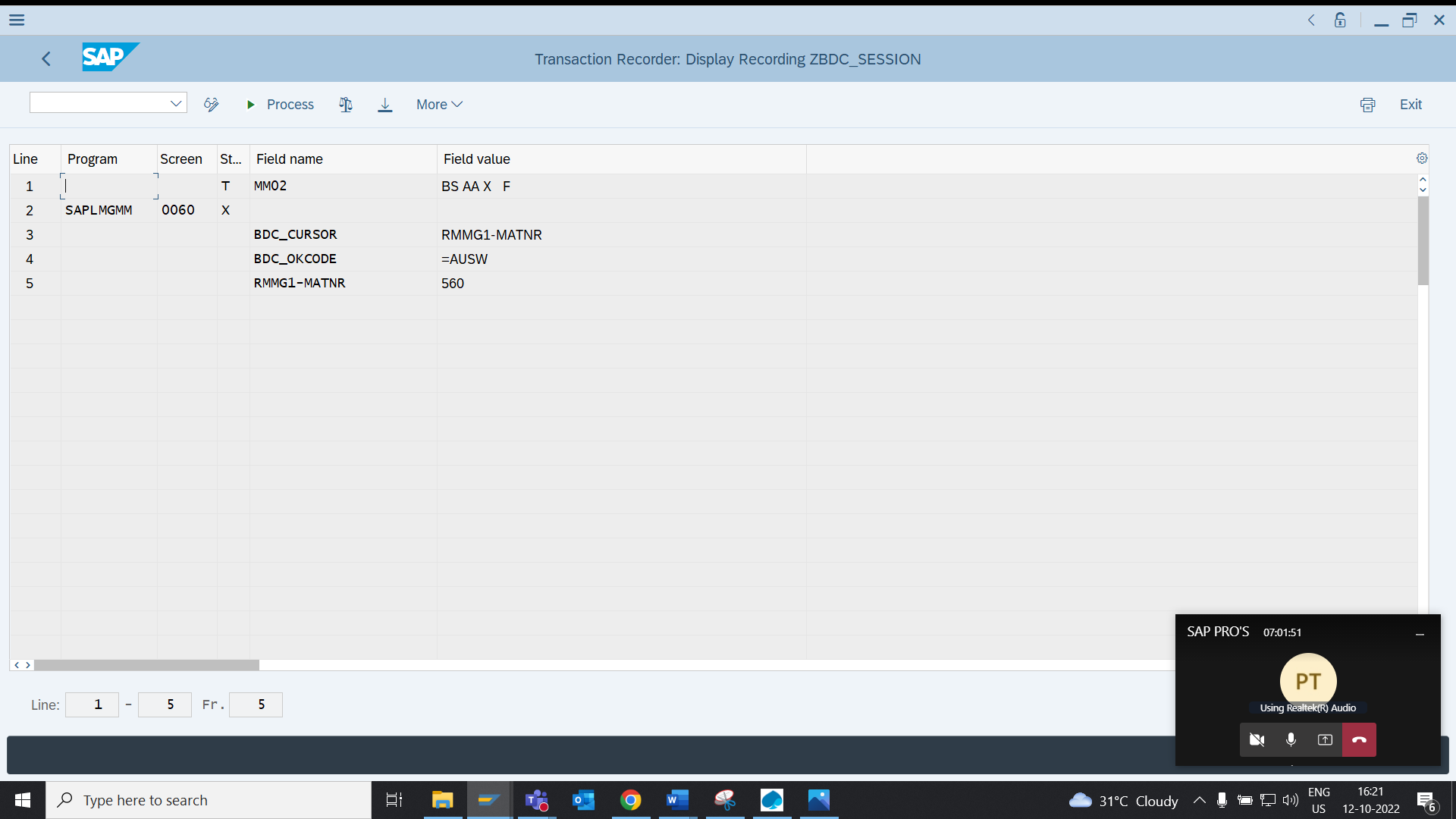


**Q2**. **Create an Upload program Using BDC Session Method to update data in MM02(Material Change).**

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



Now Save the Recording .



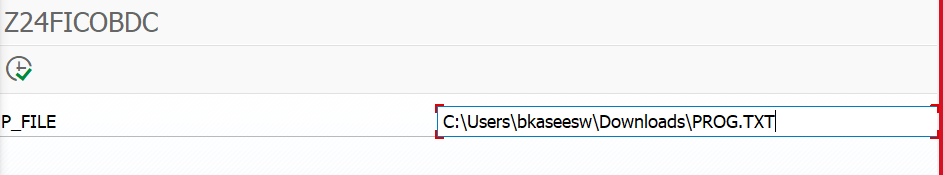
* Now select the created record and click on Program and create the program.
* Then BDC program will be created automatically and you can find the structures and dynpro and dynfield subroutines in Include BDCRECX1 main program.Copy them and paste in your source code.

**REPORT:**

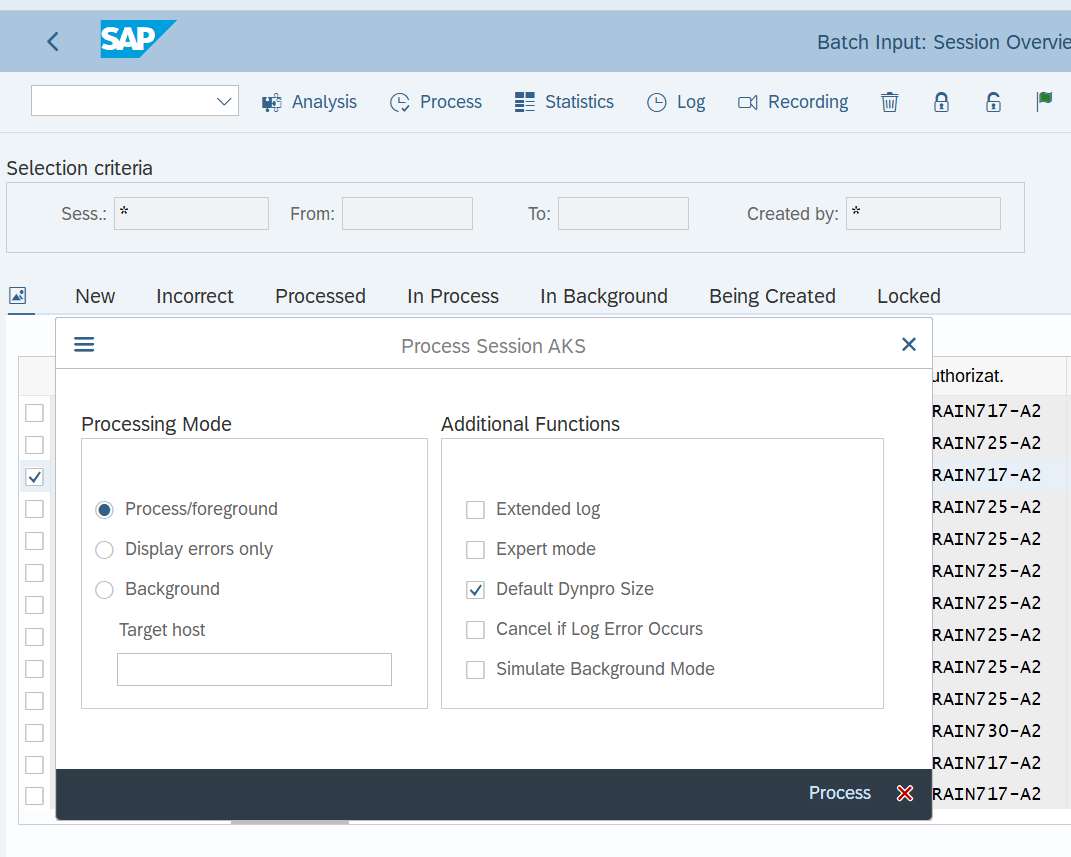
\*&---------------------------------------------------------------------\*  
\*& Report Z5236\_BDC\_SESSION  
\*&---------------------------------------------------------------------\*  
\* Description:           BDC SESSION METHOD USING MM02                                              \*  
\*                                                                      \*  
\* Author:     BOLLAKASEESWARI                                                  \*  
\*                                                                      \*  
\* 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.

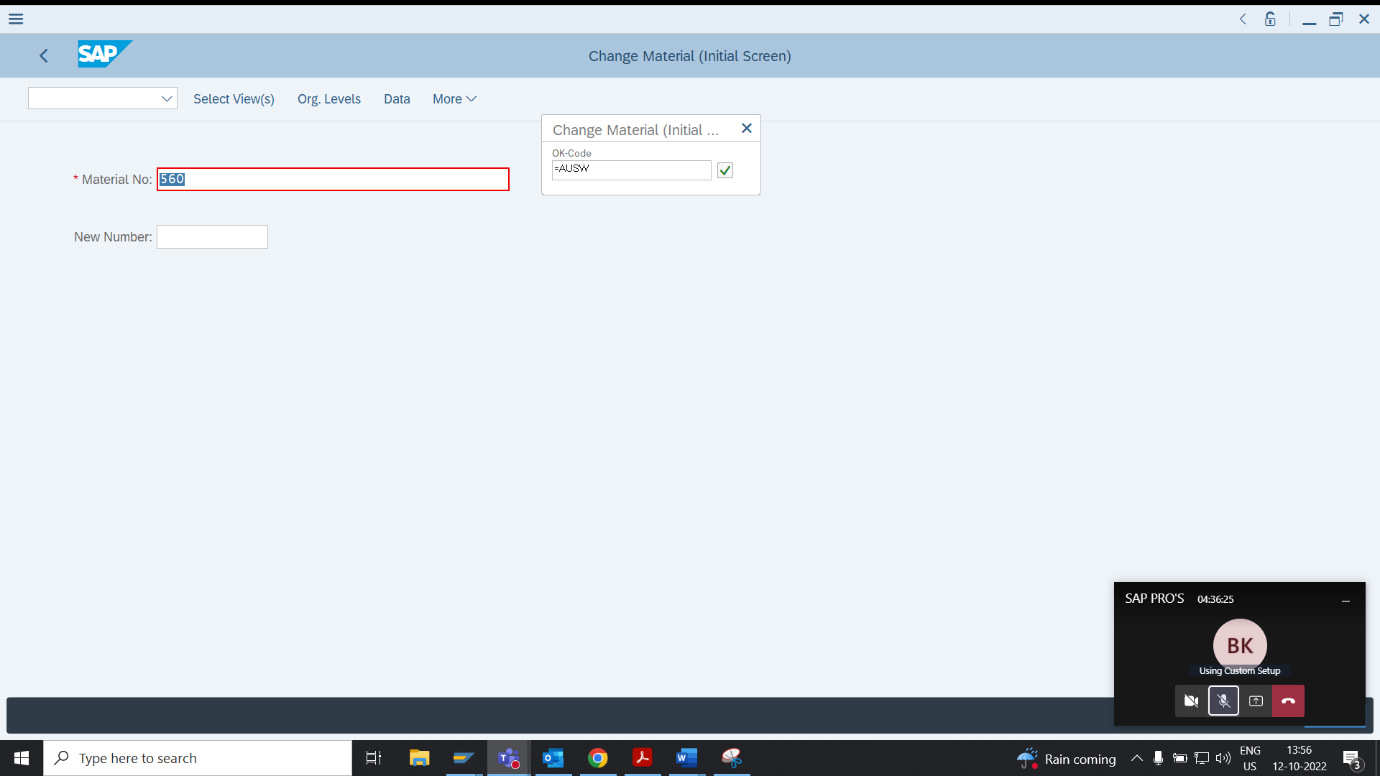
**OUTPUT:**

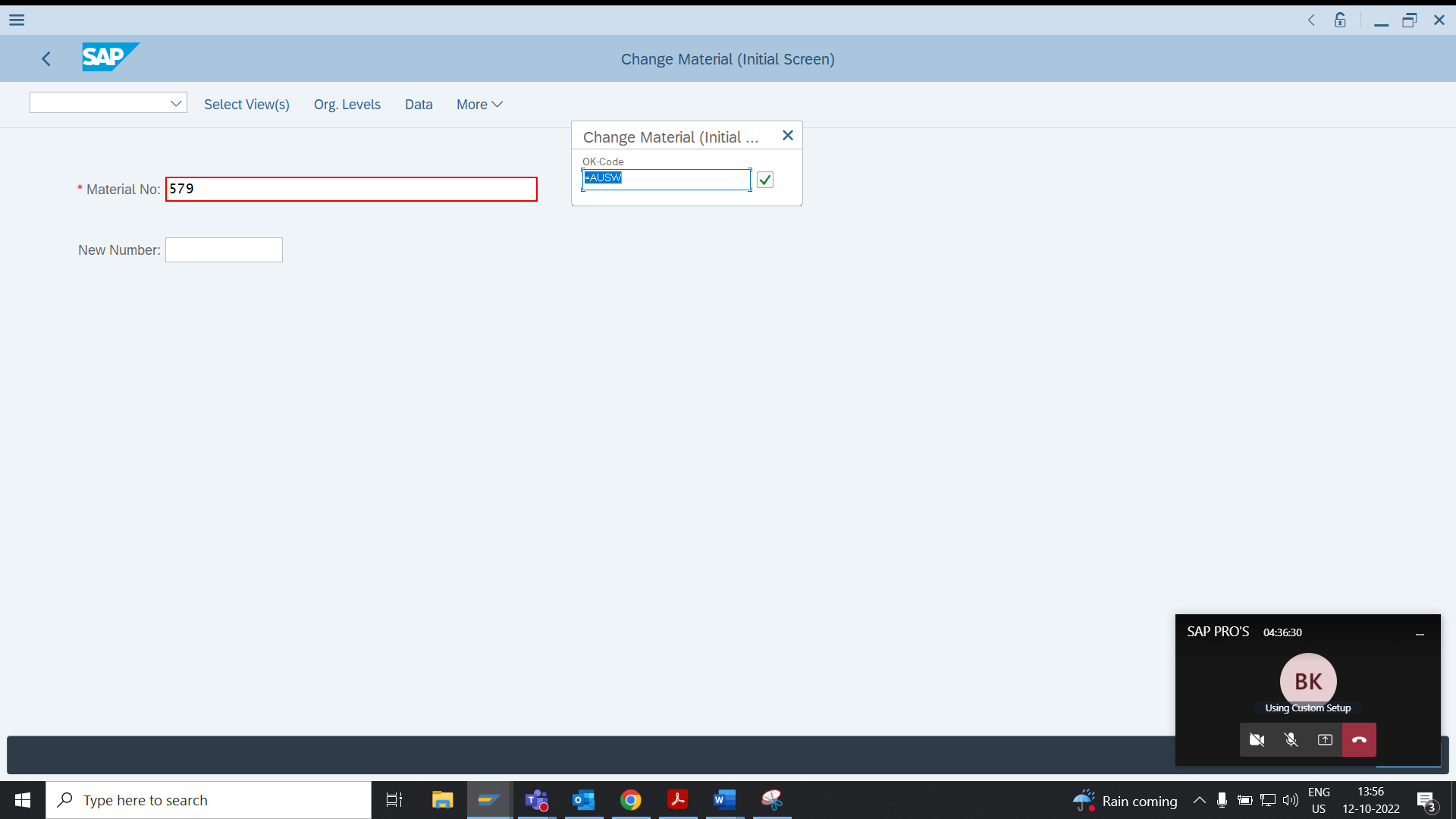
**selection screen:**

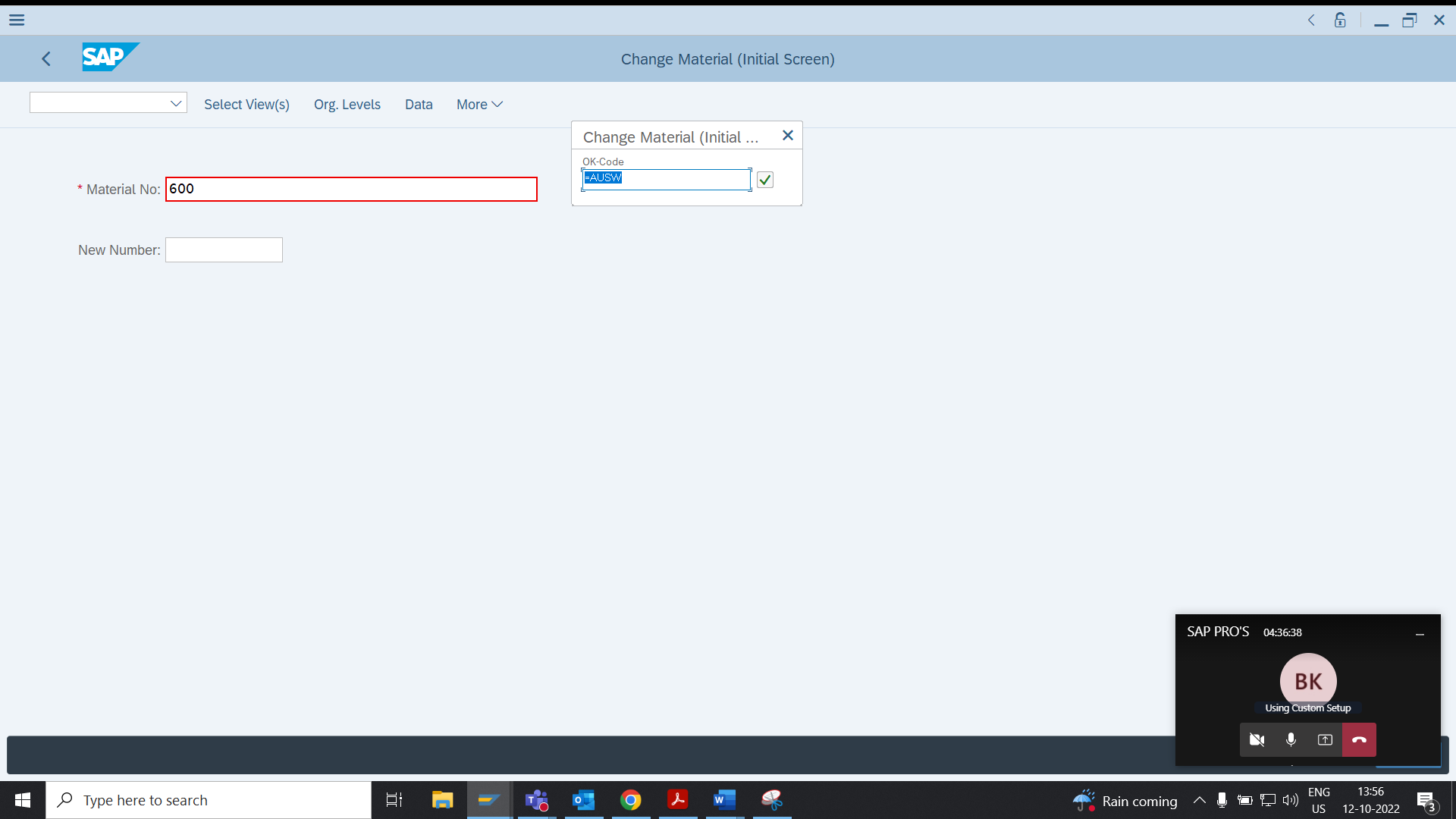


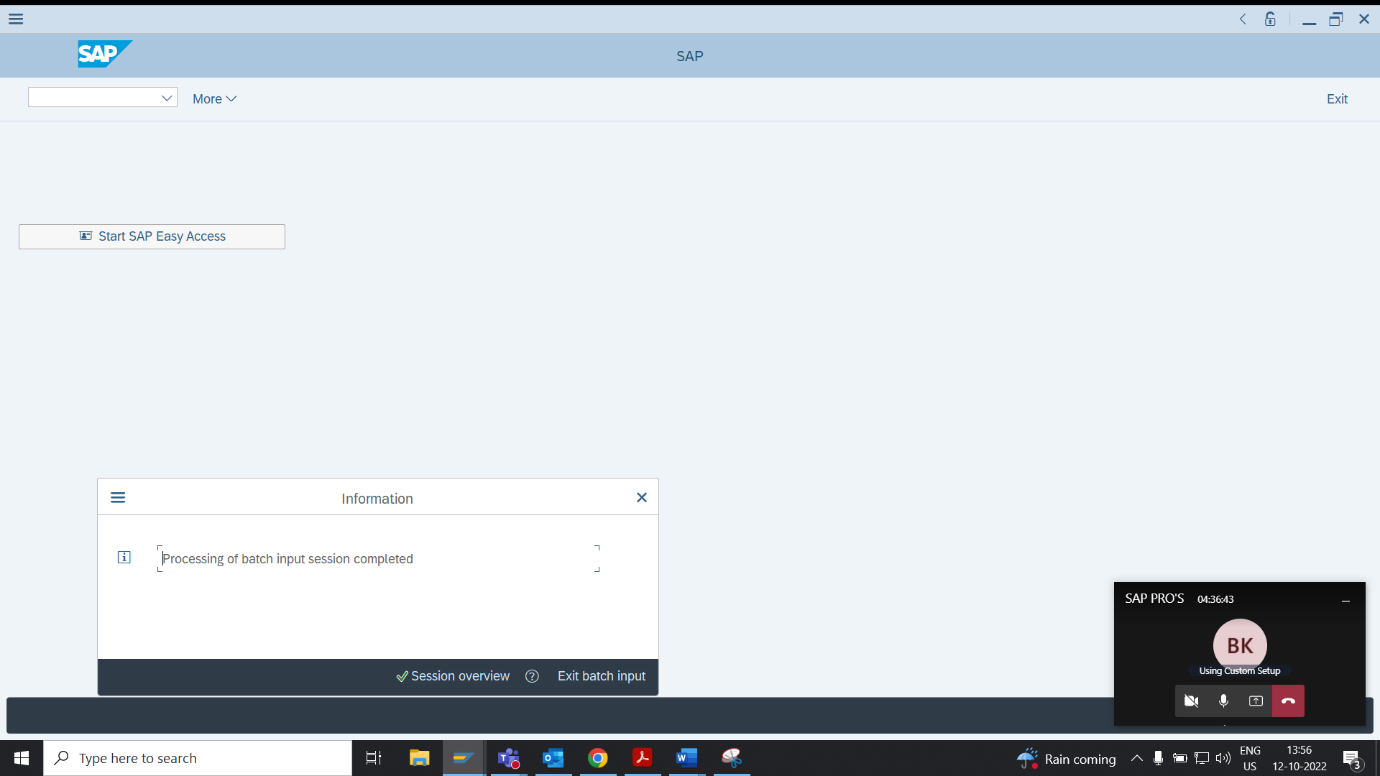
* Now program successfully transferred the data through two recorded screens of FI01 transaction code .





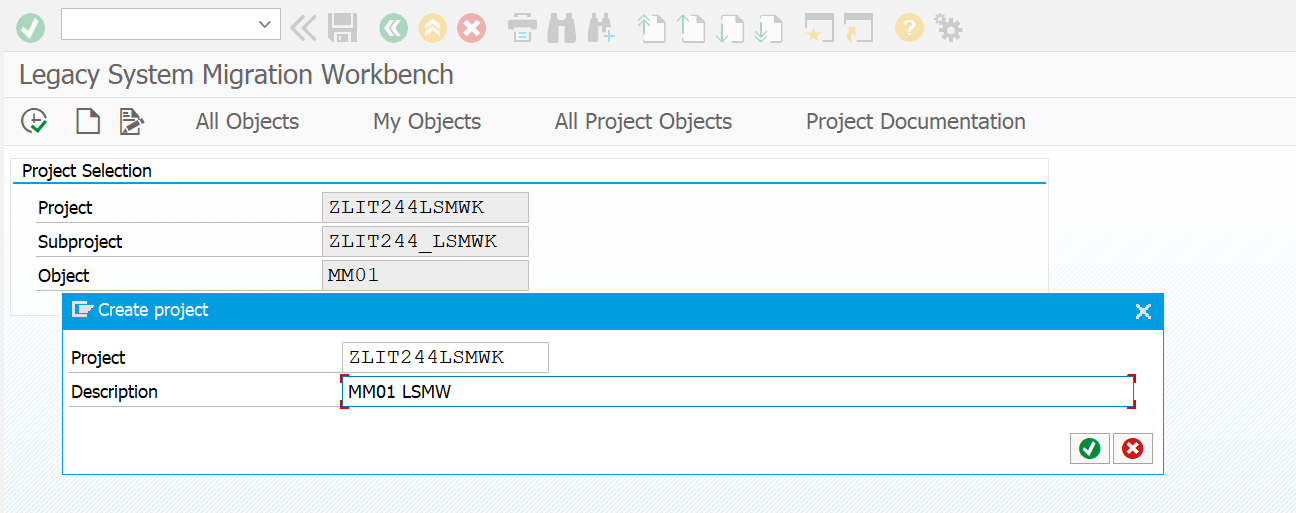


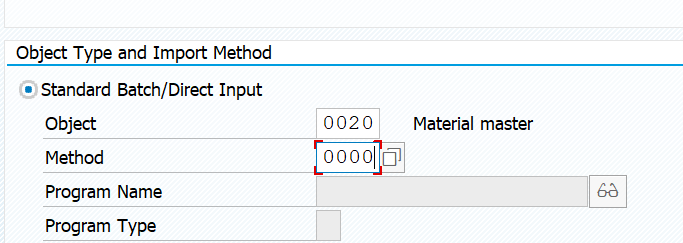


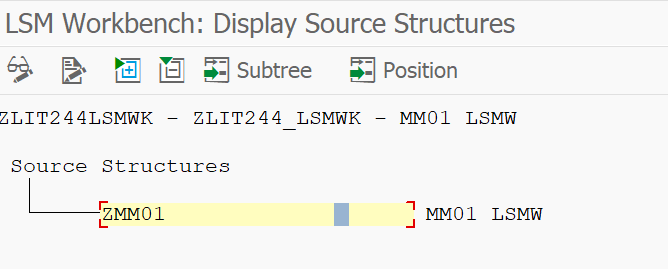


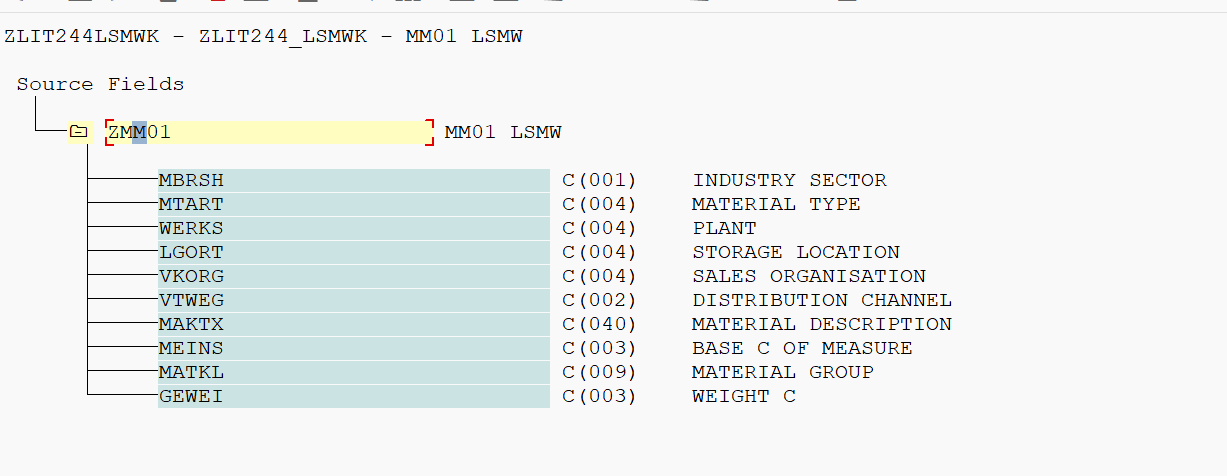
* Now program successfully transferred the data through two recorded screens of FI01 transaction code .

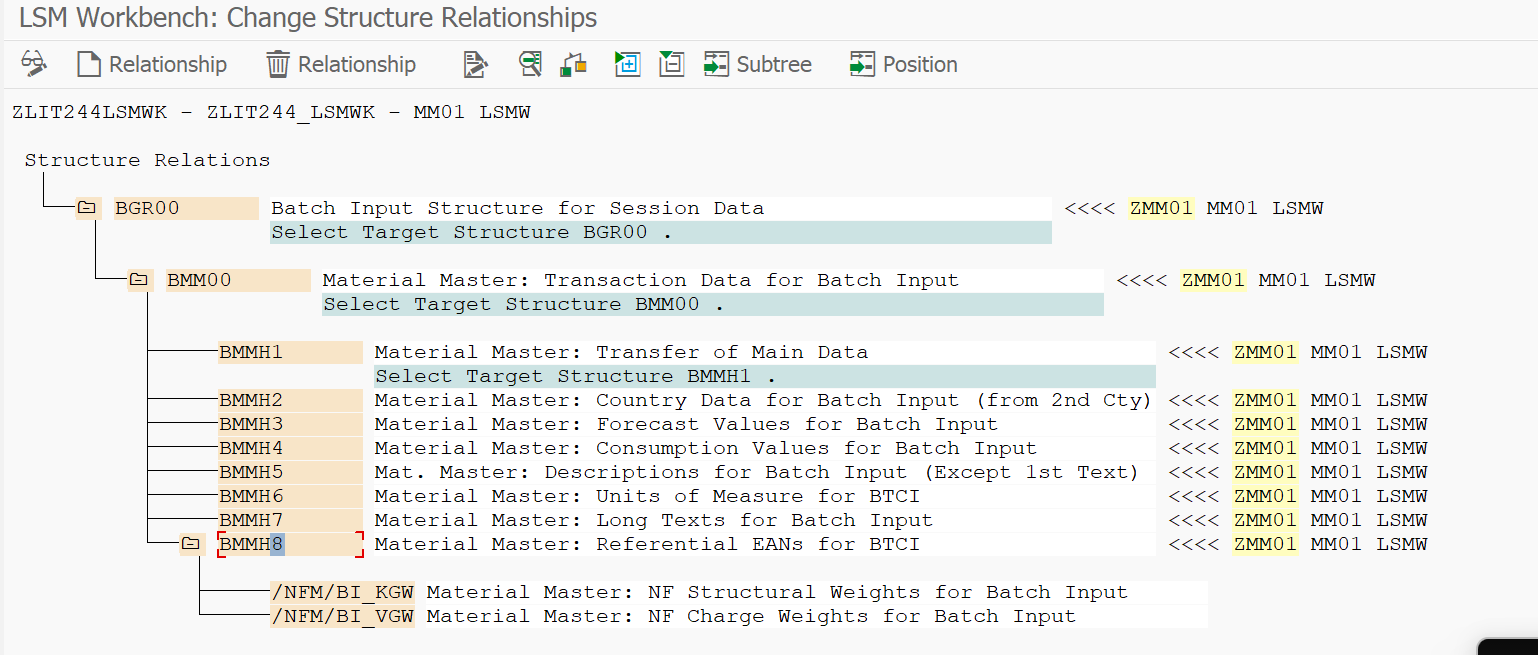
**3-Q Create a LSMW to update Data in ~~Transaction XD02.(Customer Key) -~~ Try LSMW with MM01, MM02, VA01 etc**

****

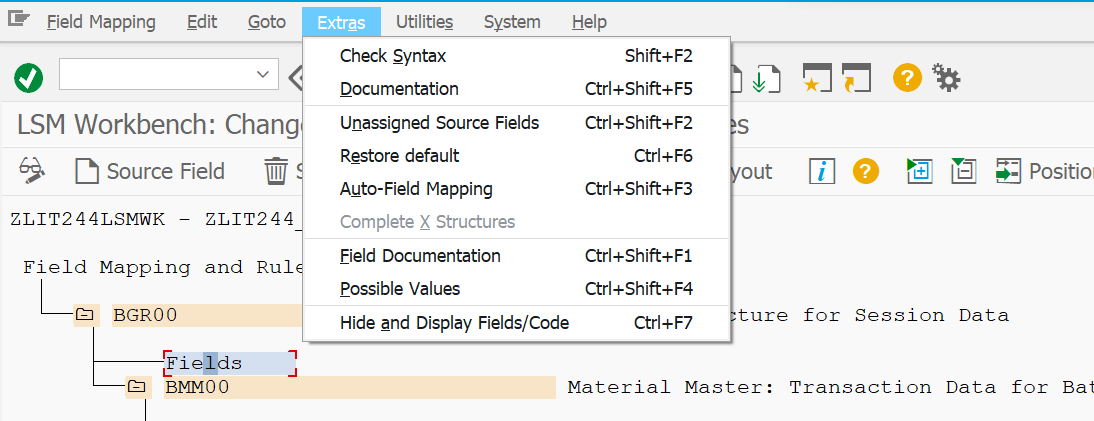
****

****

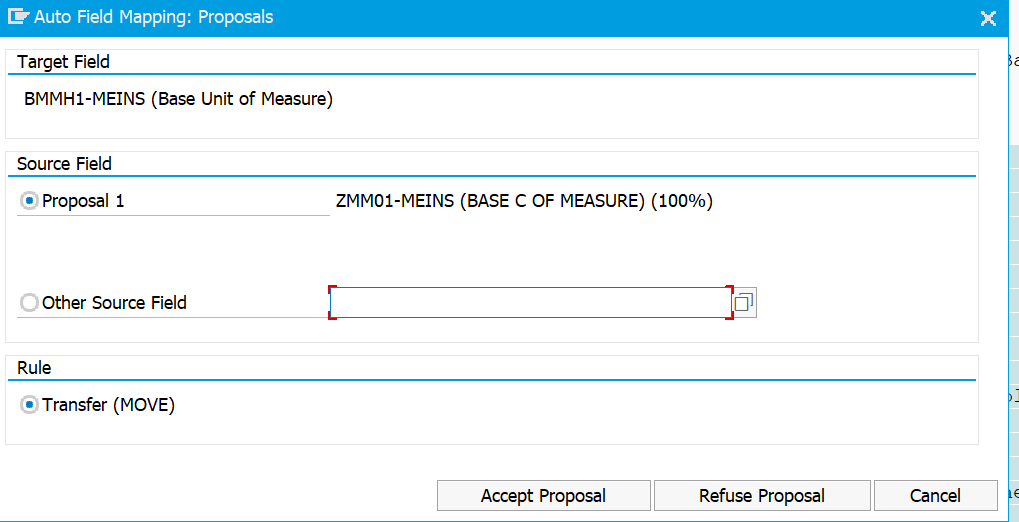
****

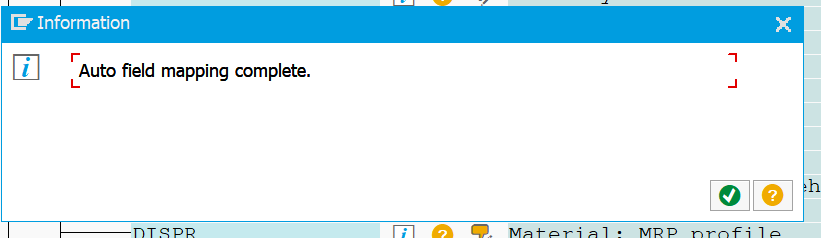
****

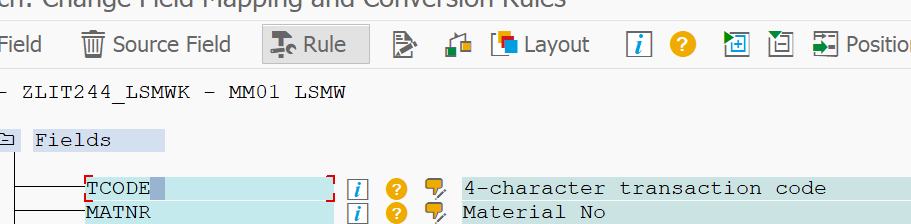
**AUTO FIELD MAPPING**

****

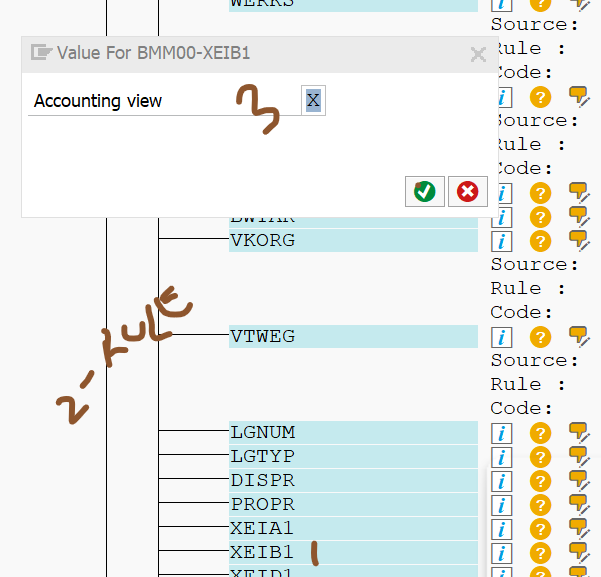
**ACCEPT PROPOSAL FOR ALL THE FIELDS**

****

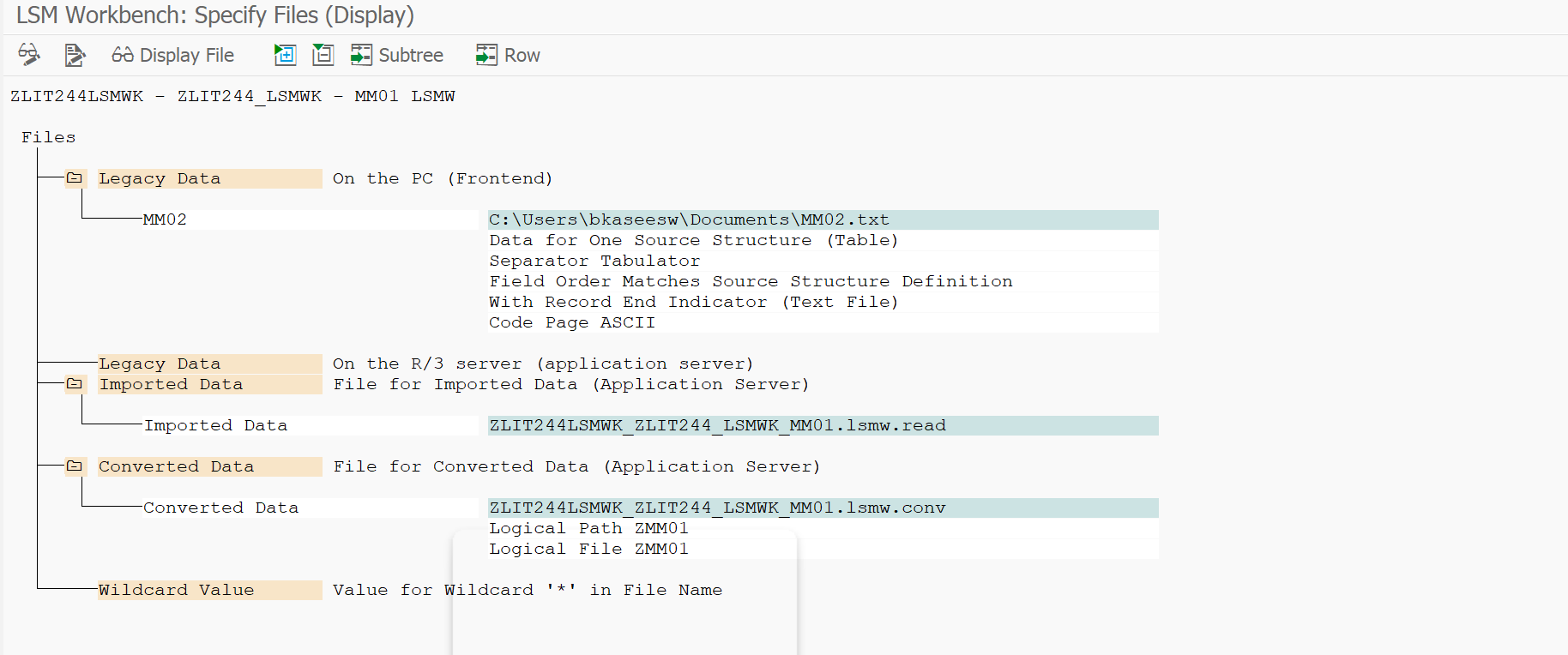
****

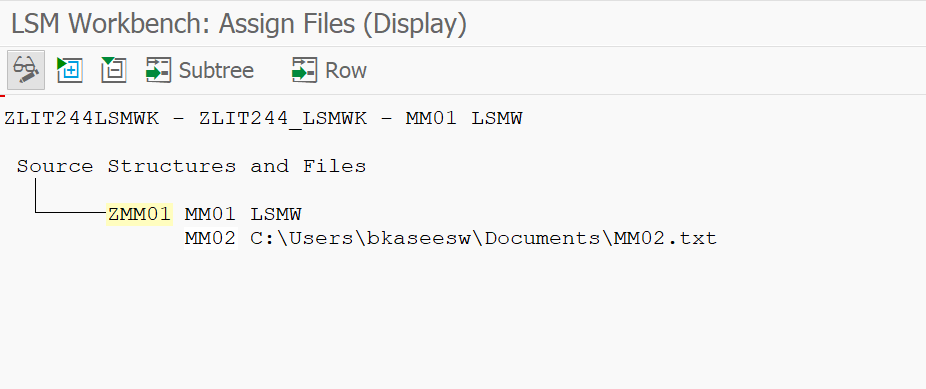
****

**MAINTAINING REQUIRED FIELDS MANUALLY**

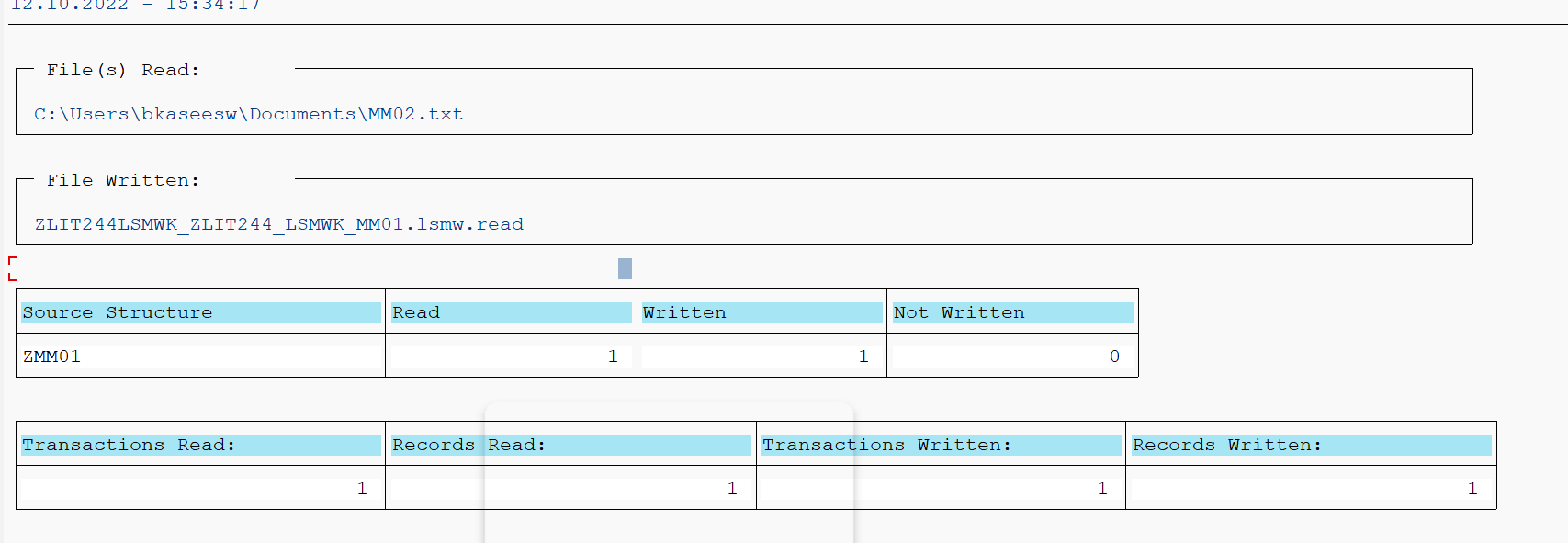
****

**UPLOADING NOTEPAD FILE IN SPECIFIED FIELDS**

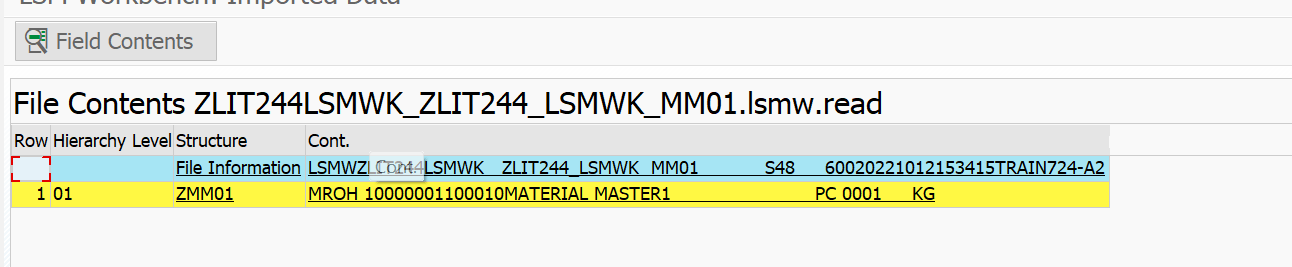
****

****

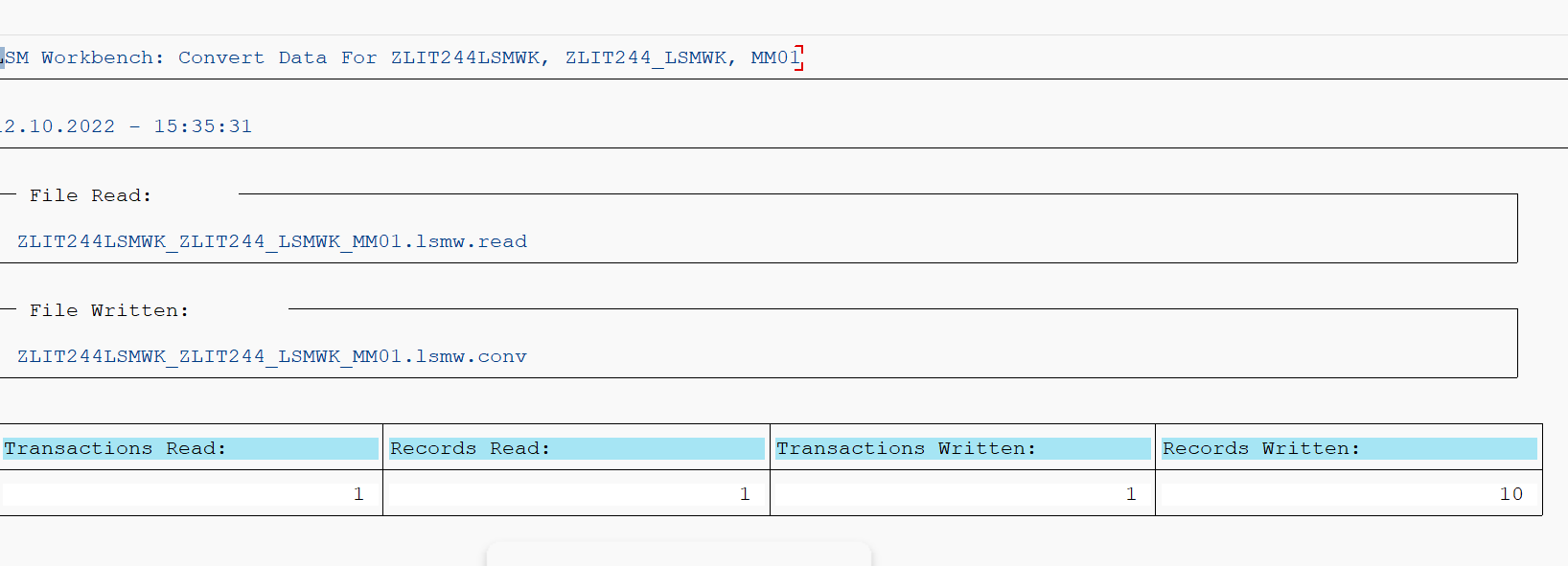
**READ DATA**

****

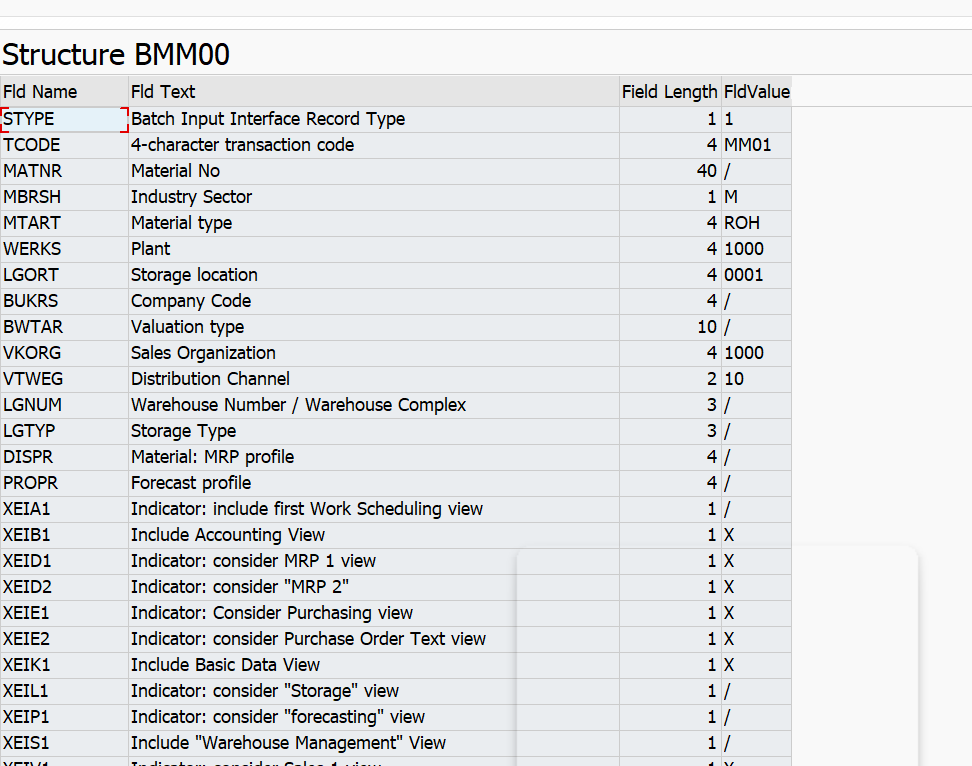
**DATA DISPLAY**

****

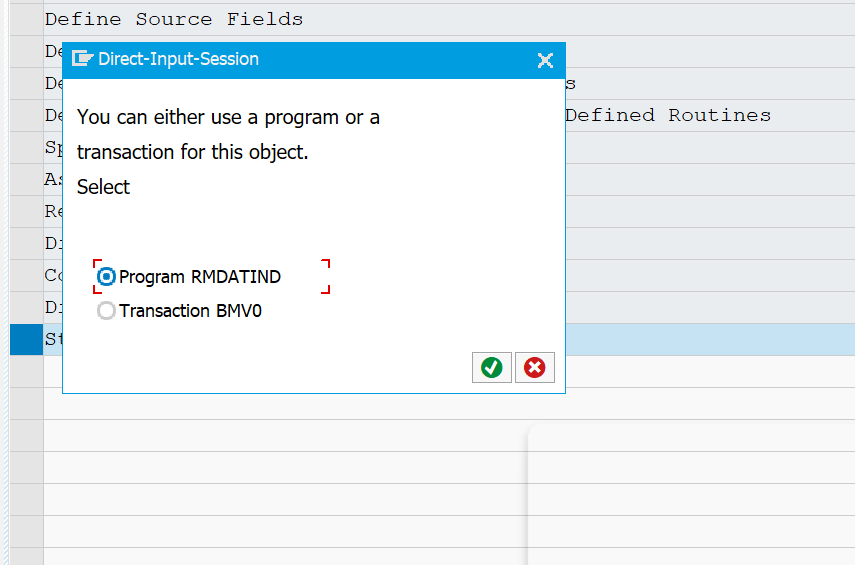
**CONVERTING FILE INTO CSV**

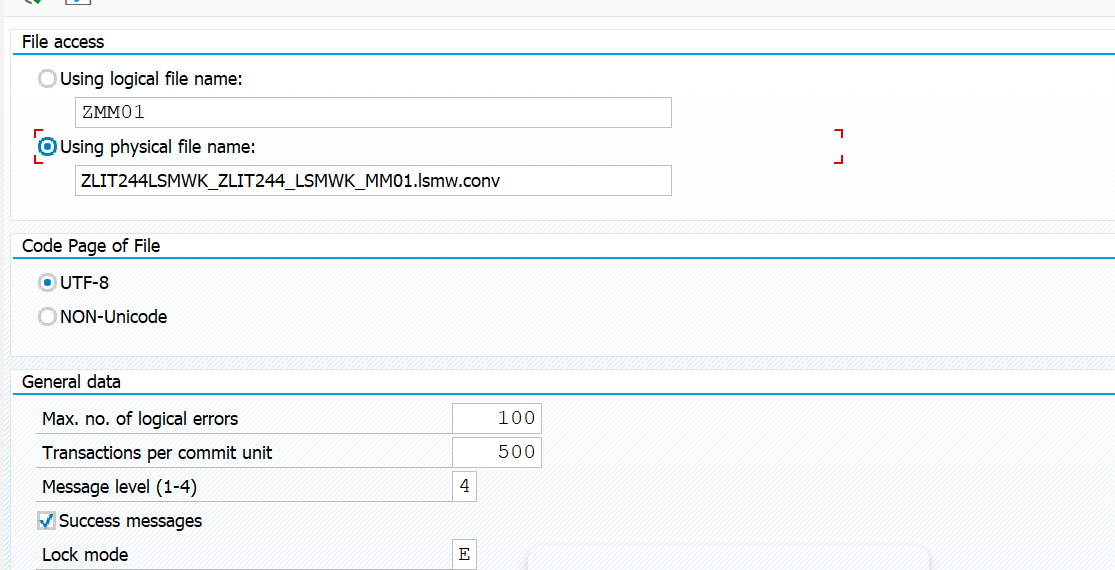
****

**DISPLAYING CONVERTED DATA AND MANUALLY CHANGED FIELDS**

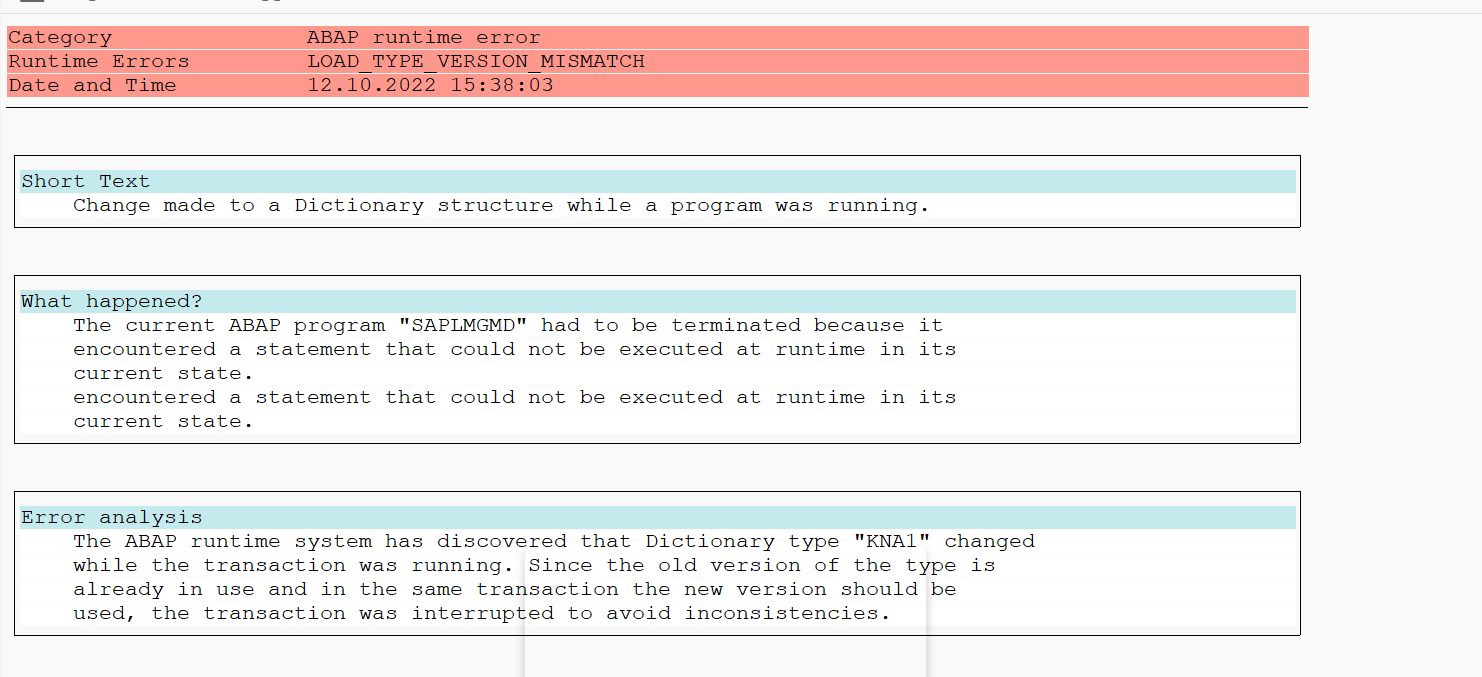
****

**DIRECT INPUT SESSION**

****

****

**RUN TIME ERROR DUE TO SERVER ISSUE - CHANGE IN KNA1 TABLE**

****