

SAP ABAP Handbook

by Kogent Learning Solutions, Inc.
Jones and Bartlett Publishers. (c) 2010. Copying Prohibited.

Reprinted for Julio De Abreu Molina, IBM

jdeabreu@ve.ibm.com

Reprinted with permission as a subscription benefit of **Books24x7**,
<http://www.books24x7.com/>

All rights reserved. Reproduction and/or distribution in whole or in part in electronic, paper or other forms without written permission is prohibited.



Chapter 11: The BDC and LSMW Tools

Overview

At times, you might need to transfer legacy data files from one SAP system to another or from a non-SAP system to an SAP system. Batch Data Communication (BDC) and Legacy System Migration Workbench (LSMW) are two migration tools used to transfer data in the SAP system. BDC uses some methods, such as the session and call transaction methods, to transfer data from one SAP system to another or from a non-SAP system to an SAP system. It is used to transfer data through SAP transactions. This means when you use the BDC tool for data transfer, the sequence of steps is the same as when you use standard SAP transaction screens to upload data. The only difference is that you can use different options for foreground and background processing.

The LSMW tool, on the other hand, is used to migrate legacy data from a non-SAP system to an SAP system. It provides various options, such as batch input, direct input, BAPIs, and IDocs to import legacy data into an SAP system. Some examples of legacy data are text files, Microsoft Excel sheets, and Comma Separated Values (CSV) files. The LSMW tool converts the legacy data into batch input files automatically. However, if you use the BDC tool, you have to write your batch input file manually, line by line. The BDC tool provides a collection of batch input functions so that you can write your own batch input file. On the other hand, the LSMW tool opens a wizard by which you can create batch input files. Note that a wizard can handle only some specific formats of files, such as Excel and CSV files, but not all file formats. You need to use the BDC method to transfer files not supported by LSMW.

In this chapter, you learn about BDC and LSMW. The chapter starts by explaining various techniques to transfer external data into an SAP system. Next, the chapter explores various data transfer methods. Then, you learn how to transfer external data into an SAP system by using BDCDATA structure. Finally, we discuss the BDC and LSMW tools in detail.

The Data Transfer Techniques

Data can be uploaded into an SAP system with some predefined standard data transfer techniques. Depending on the requirements of the user, the data can be transferred directly, through a batch input session, or by using the BAPI technique. The following are the three types of standard data transfer techniques employed:

- The batch input technique
- The direct input technique
- The BAPI technique

The Batch Input Technique

In the batch input technique, data is transferred through a batch input session. For this, a batch input session first is created by the data transfer program (also known as the batch input program). This batch input session holds all the transferable data related to a transaction. This data is then processed and transferred to the SAP system.

The following are some of the benefits of the batch input technique:

- Facilitates transfer of large sets of data from a non-SAP system to an SAP system or between two SAP systems.
- Allows periodic data transfer at regular intervals, such as hourly or monthly.
- Needs less monitoring as compared to other techniques. Only a periodic check made by the system administrator is sufficient to ensure that all the batch input sessions are running properly.

Figure 11.1 shows the sequence of steps required to process the batch input technique:

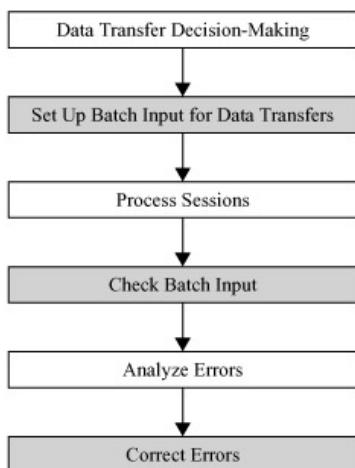


Figure 11.1: Process flow in the batch input technique

Now, let's discuss each of these steps in detail.

Data Transfer Decision-Making

The batch input process begins only after the decision to transfer data related to a transaction from an external source to an SAP system is made. The external source from where the data is transferred can be a legacy system or any other external system. For example, the data related to the customers in the older versions of SAP systems can be transferred to a new SAP system with the batch input process.

Set Up Batch Input for Data Transfers

The data related to a transaction can be entered in an SAP system either all at once or in small chunks at regular intervals. The manner in which data is transferred can be set with the help of customization settings in the SAP customization system.

Process Sessions

The data to be transferred in an SAP system is maintained in the batch input session. Running any batch input session means executing the transactions stored in the sessions and finally transferring the data into an SAP system. Usually, the SAP system automatically executes the batch input sessions; however, the data can also be processed with the SM35 transaction code. A batch input session is processed in the following three modes:

- **Process/foreground mode**— Specifies that you can interactively correct the incorrect transactions.
- **Background mode**— Specifies that you want to process the session in the background.
- **Display errors-only mode**— Specifies that transactions that are not yet processed and do not have any errors can be run non-interactively. If an error occurs in a transaction, the processing of the session stops and the screen on which the error has occurred is displayed.

Check Batch Input

The system administrator checks whether all the batch input sessions are running successfully. This check can be performed with the help of the SM35 transaction code provided by SAP.

Analyze Errors

In the Analyze Errors step, the system administrator analyzes the errors that have occurred while processing the transactions.

Correct Errors

After analyzing the errors, they need to be corrected, which is done by the combined effort of the system administrator and the department associated with the data transfer. The batch input system ensures that the processed transactions in a session are not executed again.

The Direct Input Technique

In the direct input technique of data transfer, the data to be transferred into an SAP system is stored in a data transfer file. This data transfer file is then checked to verify whether it meets certain criteria for data transfer, such as the format of data being transferred. If the criteria assigned by the SAP system are met, the data is transferred directly into the SAP system and the database of the SAP system is updated. The following are the two ways in which the direct input technique can be triggered:

- **Starting the program directly**— In this case, the system does not generate an error log. Moreover, it is not possible to restart the system if an error occurs.
- **Processing the data in the background**— In this case, if any logical error occurs during the processing of data, or the program performing the data transfer ends abruptly, the processing of data can be restarted in the background. In background processing, the input file is not stored on the application server.

In the direct input technique of data transfer, the load on the SAP system is reduced because no data can be transferred twice in the SAP system.

The BAPI Technique

A BAPI is a set of standard programming interfaces that provide access to the different business processes and data used in a business application system, such as SAP. Apart from accessing different business processes running within a business application system, BAPIs define their own interfaces. These interfaces are implemented outside the SAP system in such a way that they can be called from different external systems or applications developed by different customers.

Data Transfer Methods

In an SAP system, various methods and techniques are used to transfer data. In this section, we explore certain predefined methods that facilitate easy transfer of data in an SAP system:

- The direct input method
- The call transaction method
- The batch input with sessions method

The Direct Input Method

The direct input method generally is used when you need to transfer a large amount of data directly into an SAP system. In this method, a number of function modules are called to transfer data directly to the database of an SAP system. These function modules also make relevant checks to avoid any kind of errors during data transfer. If an error occurs, the error is fixed with the help of a restart mechanism. However, to activate this mechanism, direct input programs must be executed in the background. [Table 11.1](#) lists some predefined direct input programs:

Table 11.1: Direct input programs

| Direct Input Programs | Application |
|-----------------------|---|
| RFBIBL00 | Designed for finance (FI) applications |
| RMDATIND | Designed for material management (MM) applications |
| RVAFSS00 | Designed for sales and distribution (SD) applications |
| RAALTD11 | Designed for asset management (AM) applications |
| RKEVEXT0 | Designed for controlling (CO) applications |

The Call Transaction Method

In the call transaction method, an ABAP program is created to transfer data. This ABAP program uses an ABAP statement, `CALL TRANSACTION USING`, to run an SAP transaction. In this method, the data to be transferred does not need to be stored in a session for later processing. Instead, the entire processing takes place directly with the ABAP program. The call transaction method transfers data quicker than the batch input with session method, which we discuss next.

The Batch Input with Session Method

In the batch input with session method, an ABAP program first reads the external data that is to be transferred to an SAP system. This data is then stored in a batch input session (also known as a session) created by the ABAP program. This session uses SAP transactions to record all the actions required to transfer data into an SAP system. The transactions in the generated sessions are executed whenever the session is processed.

A session can be monitored easily with the batch input management function. You can start the batch input management function by selecting System > Services > Batch Input from the SAP Easy Access screen.

Table 11.2 lists the differences between the call transaction method and the batch input with session method:

Table 11.2: Comparing the call transaction and batch input with session method

| Parameter | Call Transaction Method | Batch Input with Session Method |
|---------------------------------|--|--|
| Data processing | The data is processed synchronously. | The data is processed asynchronously. |
| Data transfer | The data from an individual transaction is transferred each time the CALL TRANSACTION USING statement is called. | The data can be transferred from many transactions at a time. |
| Data update in the SAP database | The data can be updated in the database of the SAP system both synchronously and asynchronously. | The data can be updated in the database of the SAP system synchronously. |
| Processing log | No processing log is created for the batch input to be transferred. | A batch input processing log is generated for a session. |

Data Transfer by Using the *BDCDATA* Structure

The data to be transferred in an SAP system is first processed by a call transaction or a batch input session method. For this, you must first create an internal table. This internal table must be structured according to the predefined ABAP Dictionary structure named *BDCDATA*. The internal table stores the data that has to be transferred to an SAP system and all the actions needed to process the required transactions. The following syntax shows how to declare an internal table in an ABAP program:

```
DATA: <bdc_internal_tab> LIKE BDCDATA OCCURS <occurs-parameter>
WITH HEADER LINE.
```

In this syntax, the *<bdc_internal_tab>* expression shows the name of the internal table. *BDCDATA* represents the predefined ABAP Dictionary structure of the internal table. This *BDCDATA* structure consists of five fields: *PROGRAM*, *DYNPRO*, *DYNBEGIN*, *FNAM*, and *FVAL*.

Note To learn more about internal tables, refer to Chapter 7.

Table 11.3 shows the *BDCDATA* structure with technical characteristics:

Table 11.3: Structure of BDCDATA

| Field Name | Type | Length | Short Text |
|------------|------|--------|-------------------|
| PROGRAM | CHAR | 40 | BDC module pool |
| DYNPRO | NUMC | 4 | BDC screen number |
| DYNBEGIN | CHAR | 1 | BDC screen start |
| FNAM | CHAR | 132 | BDC field name |
| FVAL | CHAR | 132 | BDC field content |

Now, let's explore using the BDC tool to transfer data to an SAP system.

The BDC Tool

The BDC tool is the oldest batch input tool used to upload data into an SAP system. Note that the transfer of data is not bidirectional; that is, the data can only be transferred into an SAP system, not from the SAP system. The BDC tool works through an ABAP program and is based on the principle of simulating user input for transactional screens. This means that a model of all the data for different types of screens is created. The input is provided in the form of a text file, also known as a flat file. The created BDC program reads the flat file and formats the input data screen by screen into an internal table

whose structure is similar to the structure of `BDCDATA`. The transaction is then started and processed in the background by using an internal table. [Figure 11.2](#) shows how the data is transferred by using the BDC tool:

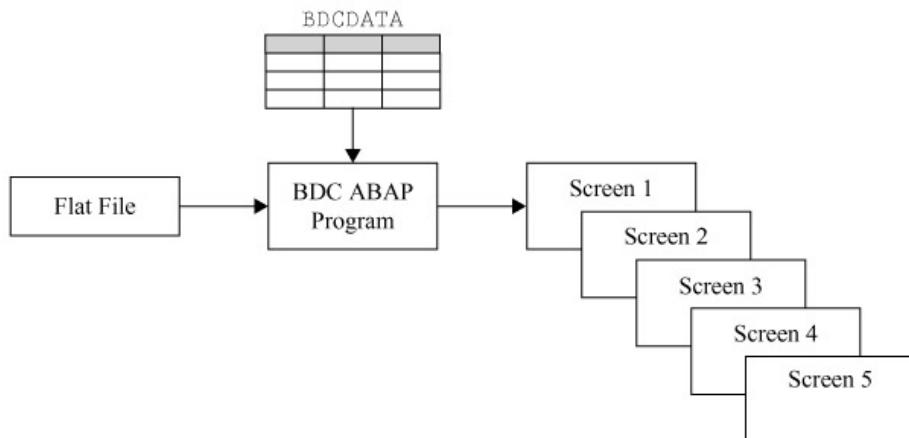


Figure 11.2: BDC method

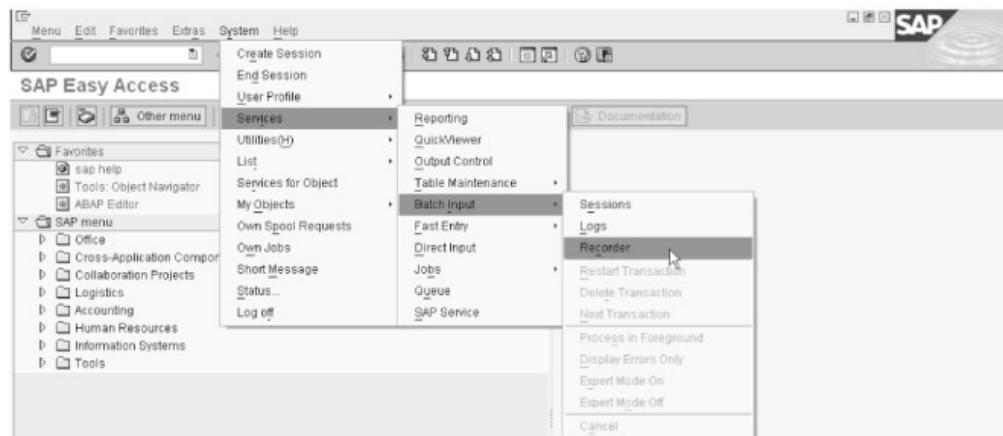
The BDC tool can be run either by using the call transaction method or by using the batch input session method. Now, let's learn how to create a BDC program by using the call transaction method.

Creating a BDC Program by Using the Call Transaction Method

In the call transaction method, we transfer data related to a material, such as the name of the material, material number, and description of a material, into an SAP system. For this, we first create a flat file, which holds the data to be transferred to an SAP system. The flat file is then uploaded into the internal table, which is then mapped into the `BDCDATA` structure. Now, by using the `CALL TRANSACTION USING` statement in the ABAP program, the data is transferred from a non-SAP system to an SAP system in a particular transaction. In this case, the transaction related to material is `MM01`.

Perform the following steps to enable data transfer:

1. Open the initial screen of the batch input recorder from the SAP Easy Access screen by selecting System > Services > Batch Input > Recorder, as shown in [Figure 11.3](#):



© SAP AG. All rights reserved.

Figure 11.3: Starting the transaction recorder screen

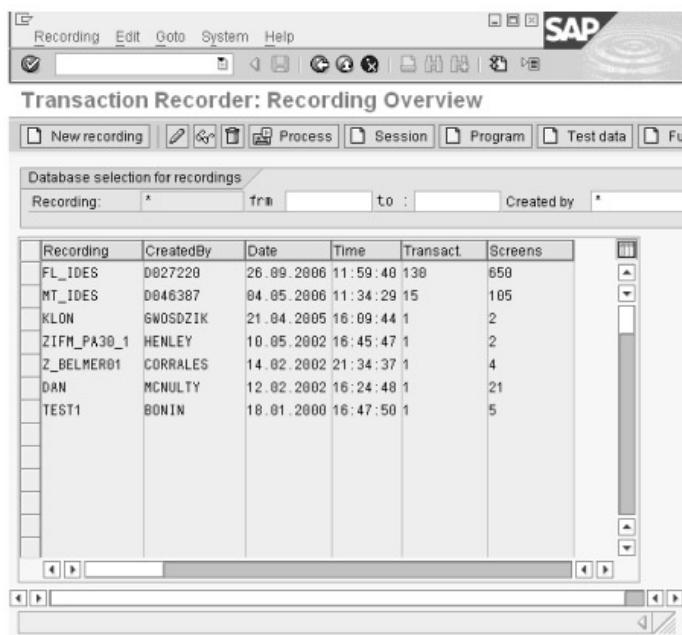
Note An alternate way to start the initial screen of the batch input recorder is by entering the `SHDB` transaction code in the Command field on the standard toolbar and pressing the ENTER key or clicking the Enter (✓) icon, as shown in [Figure 11.4](#).



© SAP AG. All rights reserved.

Figure 11.4: The transaction code in the command field

The Transaction Recorder screen appears, as shown in [Figure 11.5](#):



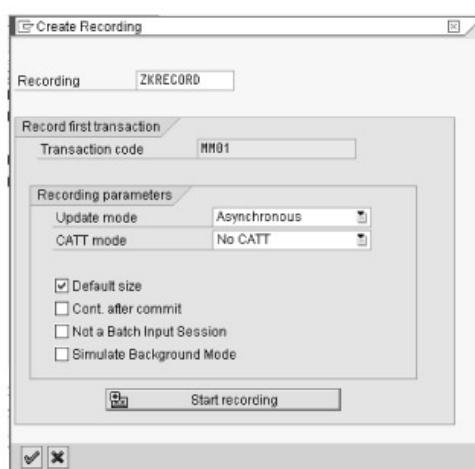
© SAP AG. All rights reserved.

Figure 11.5: The initial screen of the transaction recorder

[Figure 11.5](#) shows the recording screen, where you record the transactions related to the material data. The screen displays an overview of the previously created recordings by different users.

2. Click the New recording button.

The Create Recording dialog box appears, as shown in [Figure 11.6](#):



© SAP AG. All rights reserved.

Figure 11.6: The create recording dialog box

3. Enter the name of the recording, say, ZKRECORD, in the Recording field and the transaction, MM01, in the Transaction code field. Click the Start recording button.

The Create Material (Initial Screen) screen appears, as shown in [Figure 11.7](#):



© SAP AG. All rights reserved.

Figure 11.7: The create material initial screen

4. Enter the name of the material to be created in the Material field, say, MIRIINDA1.

The information related to the Material field, such as name, type, and length, is stored in the MARA table. In this case, the name of this field is MATNR, the type is character, and the length is 18.

5. Select the appropriate sector from the drop-down list in the Industry sector field. In this case, we have selected Food and Related Products.

The information related to Industry sector field, such as name, type, and length, is stored in the MARA table. In this case, the name of this field is MBRSH, the type is character, and the length is 1.

6. Select the appropriate sector from the drop-down list in the Material Type field. In this case, we have selected Drinks as the material type.

The information related to the Material Type field, such as name, type, and length, is stored in the MARA table. In this case, the name for the Material Type field is MTART, the type is character, and the length is 4.

Note To view the information related to a particular field, place the cursor in that field and press the F1 key. A documentation for that field appears. Now, either press the F9 key or click the Technical Information (F3) icon to view the technical information related to that field.

7. Click the Select view(s) button (see [Figure 11.7](#)).

The Select View(s) dialog box appears, as shown in [Figure 11.8](#):



© SAP AG. All rights reserved.

Figure 11.8: The select views screen

8. Select the Basic Data1 option and click the Continue (✓) icon (see Figure 11.8).

The Create Material MIRIINDA (Drinks) screen appears, as shown in Figure 11.9:

© SAP AG. All rights reserved.

Figure 11.9: The create material MIRIINDA (Drinks) screen

9. Enter the description related to the MIRIINDA material in the text box beside the Material field. In this case, we have entered cold drink, as shown in Figure 11.9.

The information related to the description of the material, such as name, type, and length, is stored in the MAKTX table. In this case, the name of this field is MAKT, the type is character, and the length is 1.

10. Select the unit as L in the Base Unit of Measure field by clicking the Search help (🔍) icon (see Figure 11.9).

The information related to the description of the Base Unit of Measure, such as name, type, and length, is stored in the MARA table. In this case, the name of this field is MEINS, the type is character, and the length is 3.

Note The number of fields that can be handled at a time is actually decided by the number of views that we select in the Select View(s) screen, as shown in [Figure 11.8](#). In this case, we have selected Basic Data 1, so a maximum of five fields can be handled at a time.

11. Click the Save (□) icon (see [Figure 11.9](#)).

In the status bar, a message Recording running appears followed by the message Recording complete; transaction has been transferred.

The Transaction Recorder: Change Recording ZKRECORD screen appears.

12. Click the Save (□) icon on the Standard toolbar to save the information displayed in the screen, as shown in [Figure 11.10](#):

| Transaction Recorder: Change Recording ZKRECORD | | | | | |
|---|-----------|--------|----|-----------------------|----------------------|
| | Program | Screen | St | Field name | Field value |
| 1 | | | T | RMMG1 | |
| 2 | SAPLMMGMM | 0060 | X | BDC_CURSOR | RMMG1-MATNR |
| 3 | | | | BDC_OKCODE | =AUSW |
| 4 | | | | RMMG1-MATNR | MIRIINDA |
| 5 | | | | RMMG1-MBRSH | F |
| 6 | | | | RMMG1-MTART | FGTR |
| 7 | SAPLMMGMM | 0070 | X | BDC_CURSOR | MSICHTAUSW-DYTXT(01) |
| 8 | | | | BDC_OKCODE | =ENTR |
| 9 | | | | MSICHTAUSW-KZSEL (01) | X |
| 10 | SAPLMMGMM | 0060 | X | BDC_CURSOR | RMMG1-MATNR |
| 11 | | | | BDC_OKCODE | =AUSW |
| 12 | | | | RMMG1-MATNR | MIRIINDA1 |
| 13 | | | | RMMG1-MBRSH | F |
| 14 | | | | RMMG1-MTART | FGTR |
| 15 | SAPLMMGMM | 0070 | X | BDC_CURSOR | MSICHTAUSW-DYTXT(01) |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |

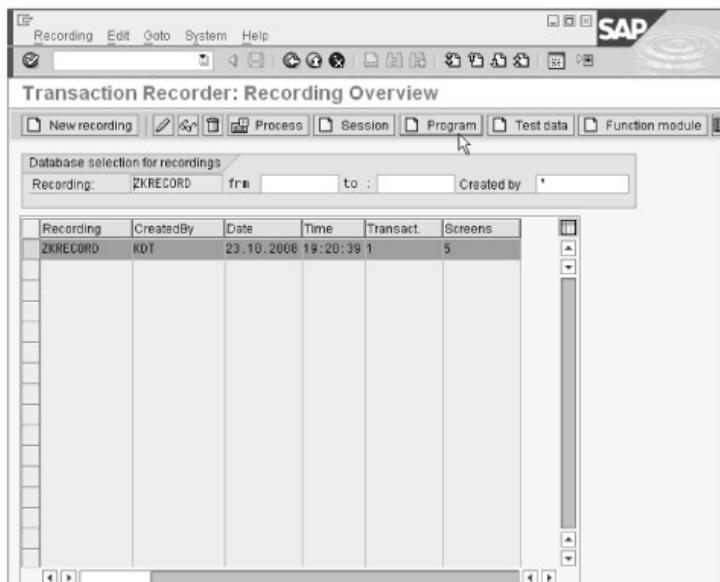
© SAP AG. All rights reserved.

Figure 11.10: The transaction recorder—change recording ZKRECORD screen

A message Recording complete; transaction has been transferred appears in the status bar. In addition to this, you see five different columns, Program, Screen, Status, Field name, and Field value.

13. Click the Back (⌚) icon (see [Figure 11.10](#)).

The Transaction Recorder: Recording Overview screen appears, as shown in [Figure 11.11](#):

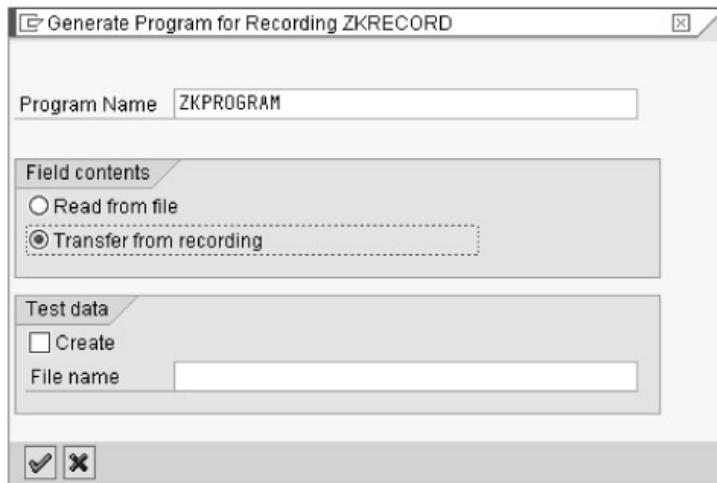


© SAP AG. All rights reserved.

Figure 11.11: The transaction recorder—recording overview screen

14. Select the name of the recording (that is, ZKRECORD) and click the Program button, as shown in Figure 11.11.

The Generate Program for Recording ZKRECORD dialog box appears, as shown in Figure 11.12:

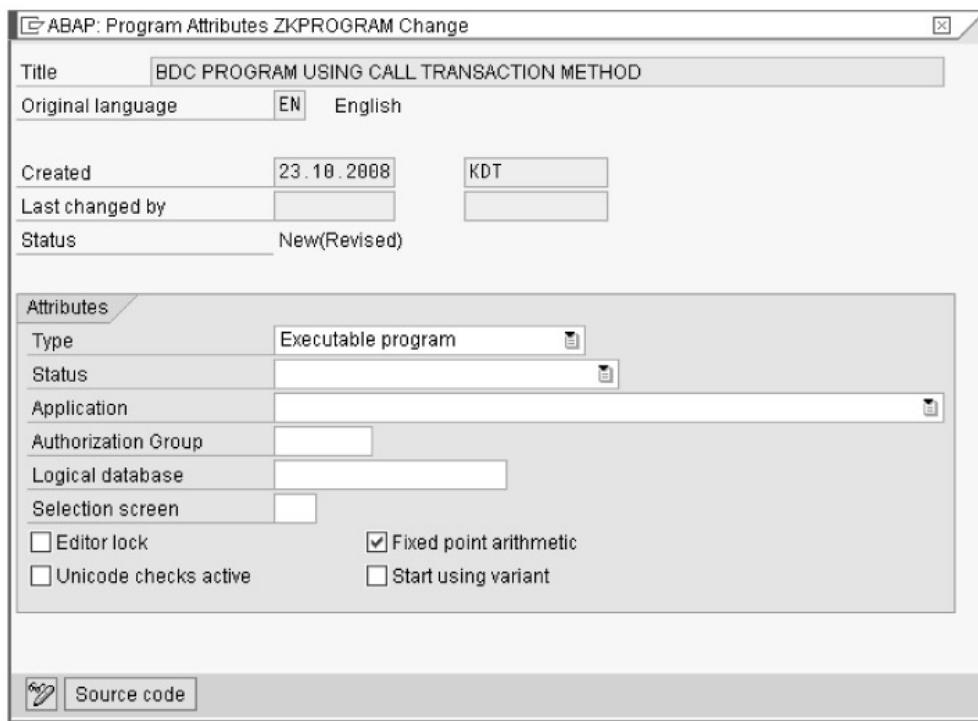


© SAP AG. All rights reserved.

Figure 11.12: Creating a program

15. Enter the name of the program, say, ZKPROGRAM, in the Program Name field, select the Transfer from recording radio button, and click the Continue (✓) icon (see Figure 11.12).

The ABAP: Program Attributes ZKPROGRAM dialog box appears, as shown in Figure 11.13:



© SAP AG. All rights reserved.

Figure 11.13: The attributes dialog box related to the program

16. Enter the title as BDC PROGRAM USING CALL TRANSACTION METHOD in the Title field and select the type as Executable program in the Type field (see [Figure 11.13](#)).
17. Click the Source code button and save the program in the package named ZKOG_PCKG.

The ABAP Editor: Change Report ZKPROGRAM screen appears, as shown in [Figure 11.14](#):

```

REPORT ZKPROGRAM
  NO STANDARD PAGE HEADING LINE-SIZE 255.

INCLUDE BDCRECX1.

START-OF-SELECTION.

PERFORM OPEN_GROUP.

PERFORM BDC_DYNPRO      USING 'SAPLMMGMN' '0060'.
PERFORM BDC_FIELD        USING 'BDC_CURSOR'          'RMMG1-MATNR'.
PERFORM BDC_FIELD        USING 'BDC_OKCODE'         '=AUSW'.
PERFORM BDC_FIELD        USING 'RMMG1-MATNR'        IT-MATNR.
PERFORM BDC_FIELD        USING 'RMMG1-MBRSH'        IT-MBRSH.
PERFORM BDC_FIELD        USING 'RMMG1-MTART'        IT-MTART.
PERFORM BDC_DYNPRO      USING 'SAPLMMGMN' '0070'.
PERFORM BDC_FIELD        USING 'BDC_CURSOR'          'MSICHTAUSW-DYTXT(01)'.
PERFORM BDC_FIELD        USING 'BDC_OKCODE'         '=ENTR'.
PERFORM BDC_FIELD        USING 'MSICHTAUSW-KZSEL(01)' 'X'.
PERFORM BDC_DYNPRO      USING 'SAPLMMGMN' '0060'.
  
```

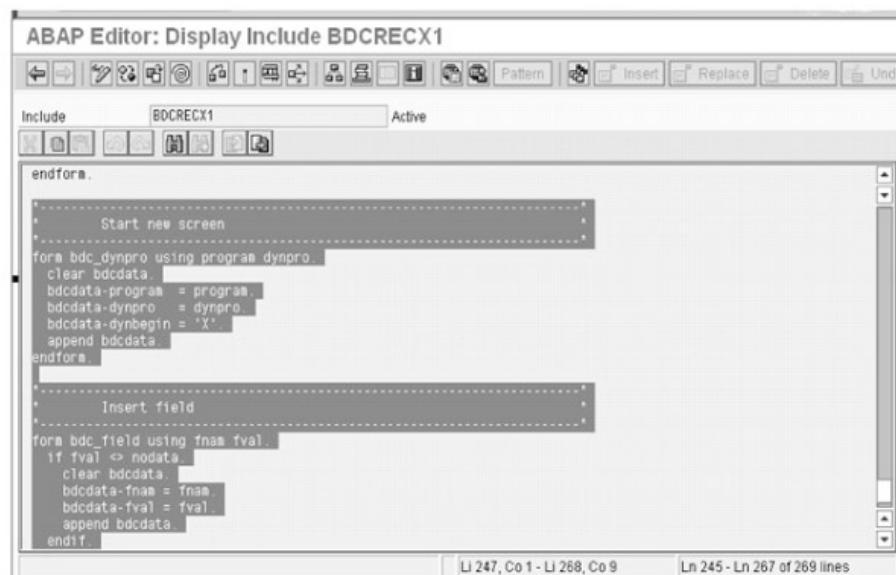
© SAP AG. All rights reserved.

Figure 11.14: Diplaying the ABAP editor screen

In [Figure 11.14](#), some predefined statements appear for the data transfer program.

18. Double-click the included structure, named bdcrecx1, as shown in [Figure 11.14](#).

The ABAP Editor: Display Include BDCCRECX1 screen appears, displaying different statements of the included structure, as shown in [Figure 11.15](#):



The screenshot shows the SAP ABAP Editor interface with the title bar "ABAP Editor: Display Include BDCCRECX1". The toolbar includes icons for file operations like Open, Save, Print, and a Pattern search. The menu bar has "File", "Edit", "View", "Insert", "Replace", "Delete", and "Undo". The status bar at the bottom shows "Ln 247, Co 1 - Ln 268, Co 9" and "Ln 245 - Ln 267 of 269 lines". The main editor area contains two sections of ABAP code:

```

endform.

* Start new screen
form bdc_dynpro using program dynpro.
  clear bddata.
  bddata-program = program.
  bddata-dynpro = dynpro.
  bddata-dynbegin = 'X'.
  append bddata.
endform.

* Insert field
form bdc_field using fnam fval.
  if fval <> nodata.
    clear bddata.
    bddata-fnam = fnam.
    bddata-fval = fval.
    append bddata.
  endif.

```

© SAP AG. All rights reserved.

Figure 11.15: Listing of the BDCCRECX1 structure

19. Scroll down to the end of the page and copy two subroutines, named `form bdc_dynpro using program dynpro` and `form bdc_field using fnam fval`, as shown in [Figure 11.15](#).
20. Click the Back (⌚) icon (see [Figure 11.15](#)) to return to [Figure 11.14](#).
21. Paste the copied subroutines at the end of the ZKPROGRAM program. Note that when the ZKPROGRAM program is executed by performing this action, every screen appears with the automatically populated values that have to be transferred to the SAP system, as shown in [Figure 11.16](#):
22. Replace `nodata` in `form bdc_field using fnam fval` with the space addition, as shown in [Figure 11.16](#).

ABAP Editor: Change Report ZKPROGRAM

```

Report      ZKPROGRAM      Inactive
Form      ZKPROGRAM      Inactive

perform bdc_field      using 'MAKT-MAKTX'
perform bdc_field      using 'BDC_CURSOR'
perform bdc_field      using 'MARA-MEINS'.
perform bdc_field      using 'MARA-MEINS'.
perform bdc_field      using 'MARA-MTPOS_MARA'
'NORM'.
*perform bdc_transaction using 'MM01'.
*perform close_group.
ENDFORM.

*-----*
*----- Start new screen -----*
*-----*

form bdc_dynpro using program dynpro.
clear bdcdata.
bdcdata-program = program.
bdcdata-dynpro = dynpro.
bdcdata-dynbegin = 'X'.
append bdcdata.
endiform.

*-----*
*----- Insert field -----*
*-----*

form bdc_field using fnam fval.
if fval <> nodata.
  clear bdcdata.
  bdcdata-fnam = fnam.
  bdcdata-fval = fval.
  append bdcdata.
endif.
endiform.

```

Replace nodata with
space addition

© SAP AG. All rights reserved.

Figure 11.16: Subroutines in the program

The next screen appears, as shown in [Figure 11.17](#):

```

perform bdc_field      using 'MARA-MEINS'
'L'.
perform bdc_field      using 'MARA-MTPOS_MARA'
'NORM'.
perform bdc_transaction using 'MM01'.

perform close_group.

*-----*
*----- Start new screen -----*
*-----*

form bdc_dynpro using program dynpro.
clear bdcdata.
bdcdata-program = program.
bdcdata-dynpro = dynpro.
bdcdata-dynbegin = 'X'.
append bdcdata.
endiform.

*-----*
*----- Insert field -----*
*-----*

form bdc_field using fnam fval.
if fval <> space.
  clear bdcdata.
  bdcdata-fnam = fnam.
  bdcdata-fval = fval.
  append bdcdata.
endif.
endiform.

```

© SAP AG. All rights reserved.

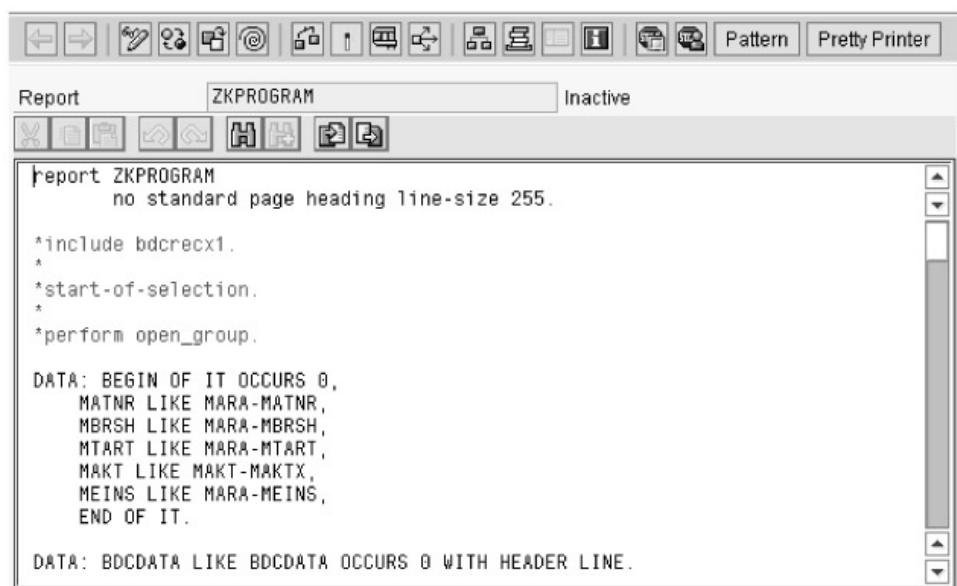
Figure 11.17: The screen with changes

23. Put an asterisk (*) before the statements that you do not want to execute. A statement preceded by an asterisk is treated as a comment and, therefore, is not executed (see [Figure 11.18](#)). In this case, we put an asterisk before the following statements:

- include bdcrecx1
- start-of-selection

- perform open_group
- perform bdc_transaction using MM01
- perform close_group

24. Declare an internal table named IT, containing five fields and a BDCDATA structure, as shown in [Figure 11.18](#):
25. Click the Pattern button (see [Figure 11.18](#)).



```

Report ZKPROGRAM Inactive
report ZKPROGRAM
no standard page heading line-size 255.

*include bdcrecx1.
*
*start-of-selection.
*
*perform open_group.

DATA: BEGIN OF IT OCCURS 0,
      MATNR LIKE MARA-MATNR,
      MBRSH LIKE MARA-MBRSH,
      MTART LIKE MARA-MTART,
      MAKT LIKE MAKT-MAKTX,
      MEINS LIKE MARA-MEINS,
      END OF IT.

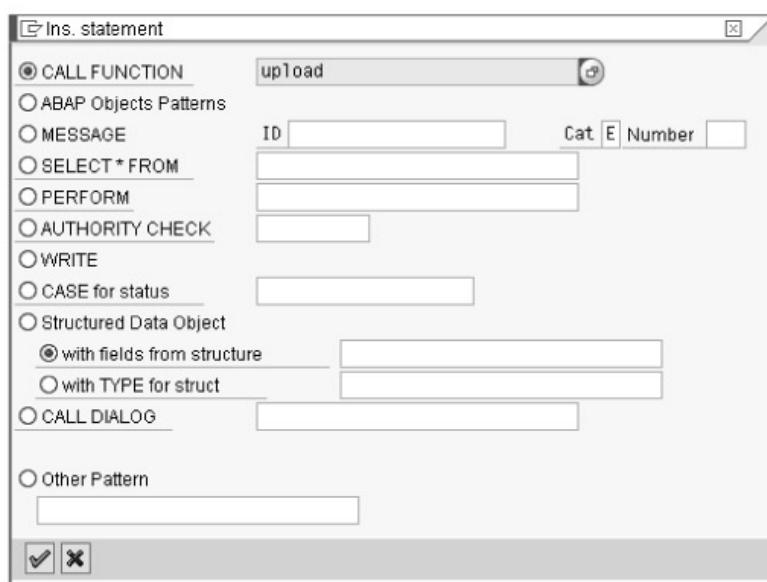
DATA: BDCDATA LIKE BDCDATA OCCURS 0 WITH HEADER LINE.

```

© SAP AG. All rights reserved.

Figure 11.18: Defining an internal table and declaring the BDCDATA structure

The Ins. statement dialog box appears, as shown in [Figure 11.19](#):



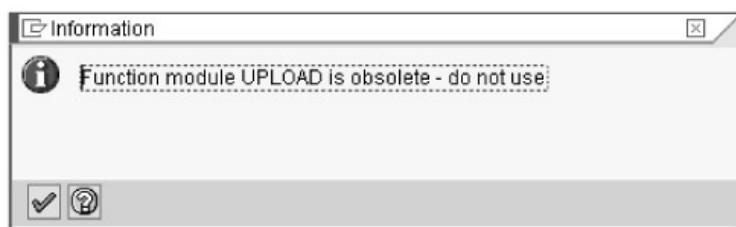
© SAP AG. All rights reserved.

Figure 11.19: Calling the function module

26. Enter the name of the function module (that is, upload) in the CALL FUNCTION field and click the Continue icon (see [Figure 11.19](#)).



The Information dialog box appears, as shown in [Figure 11.20](#):



© SAP AG. All rights reserved.

Figure 11.20: The information dialog box

27. Click the Continue () icon ([Figure 11.20](#)).

The definition of the UPLOAD function module containing various options is displayed on the ABAP Editor screen, as shown in [Figure 11.21](#):

The screenshot shows the ABAP Editor interface with the title bar 'Report ZKPROGRAM Inactive'. The code area contains the following ABAP code:

```

REPORT ZKPROGRAM.
DATA: BDCDATA LIKE BDCDATA OCCURS 0 WITH HEADER LINE.
CALL FUNCTION 'UPLOAD'
  EXPORTING
    * CODEPAGE           = ''
    * FILENAME          = 'DAT'
    * FILETYPE          = ''
    * ITEM               = ''
    * FILEMASK_MASK     = ''
    * FILEMASK_TEXT      = ''
    * FILETYPE_NO_CHANGE = ''
    * FILEMASK_ALL       = ''
    * FILETYPE_NO_SHOW   = ''
    * LINE_EXIT          = ''
    * USER_FORM          = ''
    * USER_PROG          = ''
    * SILENT             = 'S'
  IMPORTING
    * FILESIZE           =
    * CANCEL              =
    * ACT_FILENAME        =
    * ACT_FILETYPE        =
  TABLES
    DATA_TAB            = IT
  EXCEPTIONS
    CONVERSION_ERROR    = 1
    INVALID_TABLE_WIDTH = 2
    INVALID_TYPE         = 3
    NO_BATCH             = 4
    UNKNOWN_ERROR        = 5
    GUI_REFUSE_FILETRANSFER = 6
    OTHERS               = 7
  
```

© SAP AG. All rights reserved.

Figure 11.21: Changes in the UPLOAD function

28. Now, delete the asterisk (*) symbol from the EXPORTING, FILENAME, and FILETYPE options.

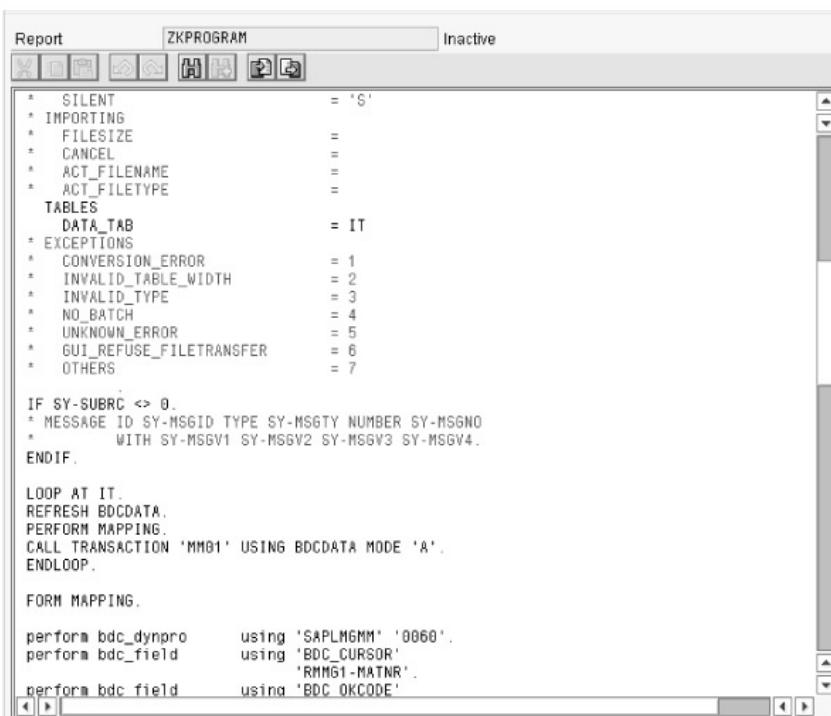
The EXPORTING option is uncommented to export the data from the external file to the internal table. The type of file that will be exported in this case is DAT, as shown in [Figure 11.21](#).

29. Now, enter the following code snippet after the definition of the UPLOAD function module:

```

LOOP AT IT.
REFRESH BDCDATA.
PERFORM MAPPING.
CALL TRANSACTION 'MM01' USING BDCDATA MODE 'A'.
ENDLOOP.
  
```

[Figure 11.22](#) shows the insertion of preceding code snippet after the definition of the UPLOAD function module:



The screenshot shows the SAP ABAP Editor with the title bar "Report ZKPROGRAM Inactive". The code area contains the following ABAP code:

```

* SILENT           = 'S'
* IMPORTING       =
* FILESIZE        =
* CANCEL          =
* ACT_FILENAME    =
* ACT_FILETYPE    =
TABLES
  DATA_TAB          = IT
* EXCEPTIONS      =
* CONVERSION_ERROR = 1
* INVALID_TABLE_WIDTH = 2
* INVALID_TYPE     = 3
* NO_BATCH         = 4
* UNKNOWN_ERROR    = 5
* GUI_REFUSE_FILETRANSFER = 6
* OTHERS           = 7

IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

LOOP AT IT.
REFRESH BDCDATA.
PERFORM MAPPING.
CALL TRANSACTION 'MM01' USING BDCDATA MODE 'A'.
ENDLOOP.

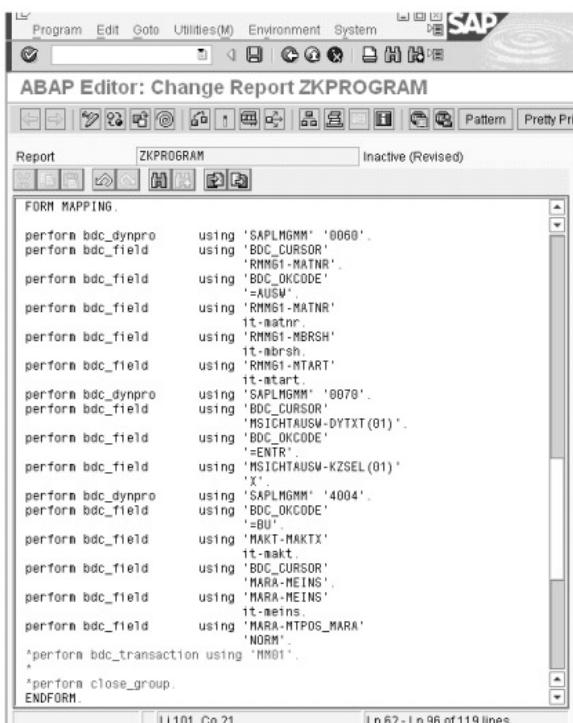
FORM MAPPING.
  perform bdc_dynpro   using 'SAPLMM01' '0060'.
  perform bdc_field    using 'BDC_CURSOR'.
  perform bdc_field    using 'RMMG1-MATNR'.
  perform bdc_field    using 'BDC_OKCODE'

```

© SAP AG. All rights reserved.

Figure 11.22: Displaying code insertion after the UPLOAD function module

30. Now, in the definition of the subroutine named MAPPING, replace all the content, such as MIRIINDA, F, FGTR, cold drink, and L, with the corresponding fields of the internal tables; that is, it-matnr, it-mbrsh, it-mtart, it-makt, and it-meins, respectively, as shown in [Figure 11.23](#):



The screenshot shows the SAP ABAP Editor with the title bar "ABAP Editor: Change Report ZKPROGRAM". The code area contains the following ABAP code:

```

FORM MAPPING.
  perform bdc_dynpro   using 'SAPLMM01' '0060'.
  perform bdc_field    using 'BDC_CURSOR'.
  perform bdc_field    using 'RMMG1-MATNR'.
  perform bdc_field    using 'BDC_OKCODE'.
  perform bdc_field    using 'RMMG1-MATNR'.
  perform bdc_field    using 'RMMG1-MBRSH'.
  perform bdc_field    using 'RMMG1-MTART'.
  perform bdc_field    using 'RMMG1-MTART'.
  perform bdc_dynpro   using 'SAPLMM01' '0070'.
  perform bdc_field    using 'BDC_CURSOR'.
  perform bdc_field    using 'MSICHTAUSW-DYTXT(01)'.
  perform bdc_field    using 'BDC_OKCODE'.
  perform bdc_field    using 'ENTR'.
  perform bdc_field    using 'MSICHTAUSW-KZSEL(01)'.
  perform bdc_field    using 'X'.
  perform bdc_dynpro   using 'SAPLMM01' '4004'.
  perform bdc_field    using 'BDC_CURSOR'.
  perform bdc_field    using 'MARA-MEINS'.
  perform bdc_field    using 'MARA-MEINS'.
  perform bdc_field    using 'MARA-MTPOS_MARA'.
  perform bdc_field    using 'NORM'.
*perform bdc_transaction using 'MM01'.
*perform close_group.
ENDFORM.

```

© SAP AG. All rights reserved.

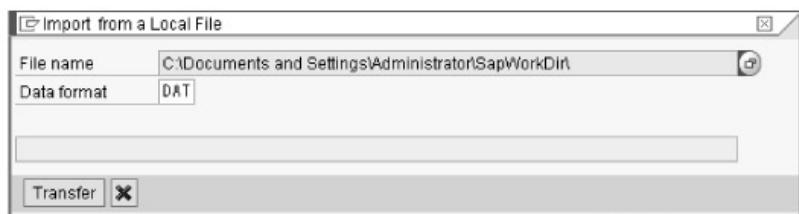
Figure 11.23: Listing of a BDC program by using the call transaction method

The content is replaced so that the data can be mapped from the flat file to the BDCDATA structure.

31. Click the Save (□) icon, the Check (■) icon, and the Activate (■) icon (see [Figure 11.23](#)).

32. Click the Direct Processing (█) icon.

The system now asks for the file name that is to be transferred into the SAP system, as shown in **Figure 11.24**:

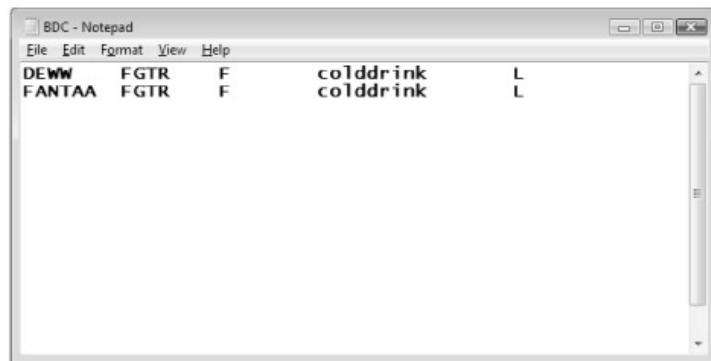


© SAP AG. All rights reserved.

Figure 11.24: The import from a local file screen

In **Figure 11.24**, we need to enter the name of the file containing the data that has to be transferred to the SAP system. For this, we first create a file in a notepad.

33. Create a notepad file and save it at a particular location so that it can be uploaded easily. **Figure 11.25** displays the notepad file:



© SAP AG. All rights reserved.

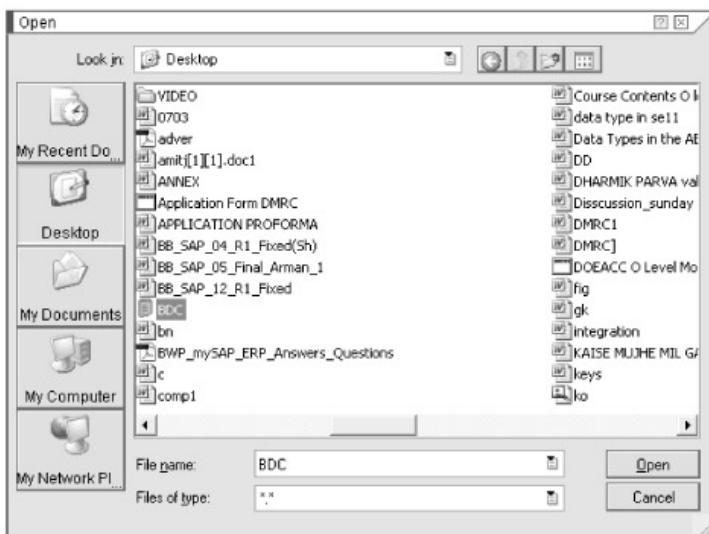
Figure 11.25: The notepad file

In the opened notepad file, we define the properties of two materials, DEWW and FANTAA, specifying the Industry sector as FGTR, Drinks as F, Material Description as colddrink, and the Basic unit of Measure as L.

34. Save this file. In this case, the file is saved as BDC on the desktop.

35. Now, click the Search Help (█) icon, as shown in **Figure 11.24**.

The Open dialog box appears, as shown in **Figure 11.26**:



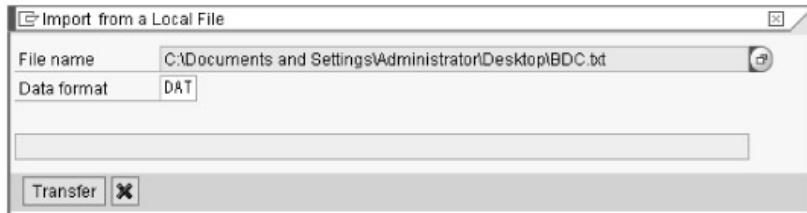
© SAP AG. All rights reserved.

Figure 11.26: Importing a file

36. Select the name of the file from the stored location, as shown in [Figure 11.26](#).

37. Click the Open button.

The Import from a Local File dialog box appears, with the address of the file in the File name field, as shown in [Figure 11.27](#):



© SAP AG. All rights reserved.

Figure 11.27: Importing the BDC named text file

38. Click the Transfer button (see [Figure 11.27](#)).

The Create Material (Initial Screen) screen appears, as shown in [Figure 11.28](#):

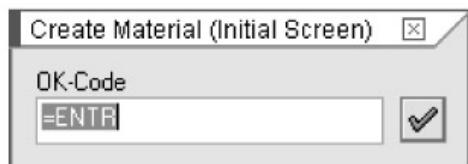


© SAP AG. All rights reserved.

Figure 11.28: The initial material screen

In the Create Material (Initial Screen) screen, the first data entry of the notepad file (that is, DEWW, FGTR,

F, colddrink, and L), gets transferred automatically to the respective fields, as shown in [Figure 11.29](#). In addition, the Create Material (Initial Screen) dialog box showing the OK-Code for the screen is also displayed.



© SAP AG. All rights reserved.

Figure 11.29: OK-Code field

Note Whenever a user selects a function code (a sequence of 20 characters that can be assigned to the specific control elements of a user interface), the SAP system copies the function code into a field called the OK-Code.

39. Click the Continue (checkbox) icon in the the Create Material (Initial Screen) dialog box (see [Figure 11.28](#)).

The OK-Code value for the Select View dialog box (see [Figure 11.8](#)) is displayed, as shown in [Figure 11.29](#):

40. Click the Continue (checkbox) icon.

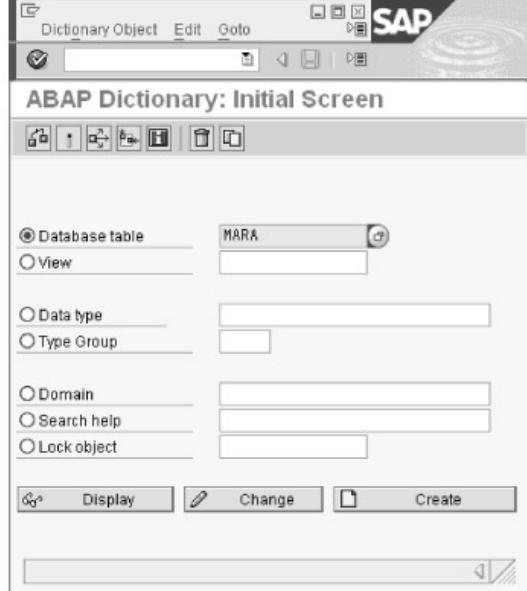
The Create Material MIRIINDA1 (Drinks) screen appears, as shown in [Figure 11.9](#).

41. Enter the description related to the material DEWW in the Description field and L in the Basic unit of measure field. Click the Save (button) icon and then click the Back (button) icon to perform the same sequence of steps for the second entry made in the notepad file; that is, FANTAA.

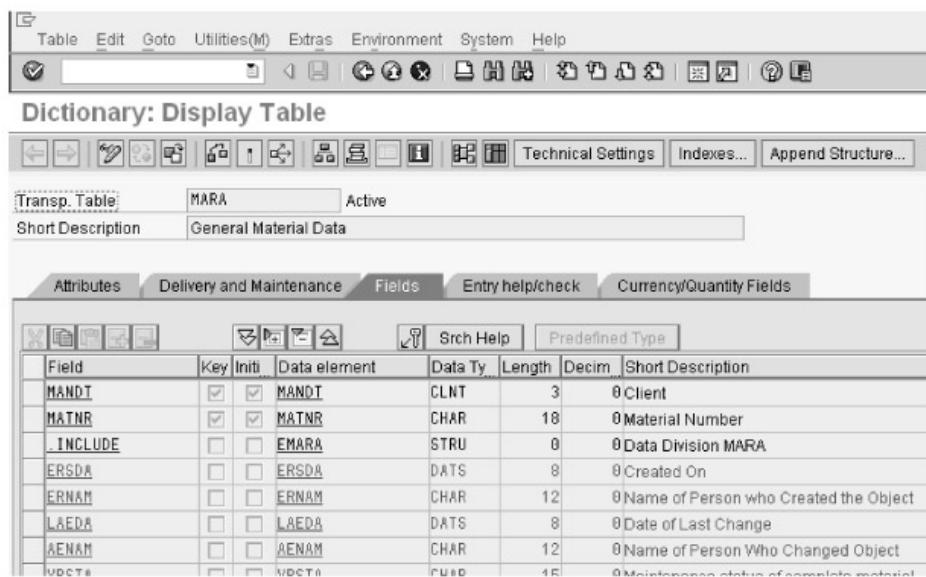
Note When you click the Back (button) icon after saving the DEWW data, a message, such as Material DEWW is created, appears in the status bar.

42. To check whether or not the material has been created, open a new session by entering the SE11 transaction code in the Command field. [Figure 11.30](#) displays the initial screen of the ABAP Dictionary:
43. Enter MARA as the name of the table corresponding to the the data related to material, and click the Display button, as shown in [Figure 11.30](#). The Dictionary: Display table screen appears, as shown in [Figure 11.31](#):

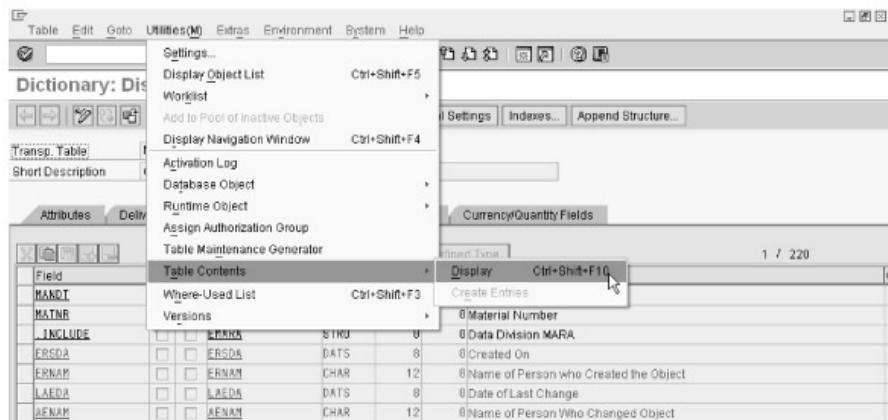
44. Select Utilities > Table Contents > Display, as shown in [Figure 11.32](#):



© SAP AG. All rights reserved.

Figure 11.30: The ABAP dictionary screen

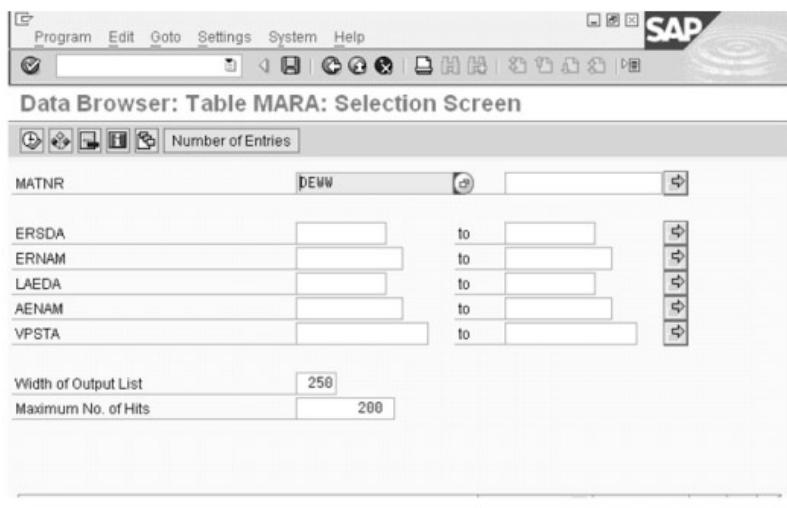
© SAP AG. All rights reserved.

Figure 11.31: Displaying the data fields of the MARA table

© SAP AG. All rights reserved.

Figure 11.32: Displaying the contents of MARA

The Data Browser: Table MARA: Selection Screen screen appears, displaying all the contents related to the selected table, as shown in [Figure 11.33](#):



© SAP AG. All rights reserved.

Figure 11.33: The screen for displaying the content of MARA

45. Enter the data to be searched in the MATNR field (that is, DEWW and FANTAA), and click the Execute (Execute icon) (see [Figure 11.33](#)).

[Figure 11.34](#) shows that the DEWW data has been transferred from a non-SAP system to an SAP system:

| MANDT | MATNR | ERSDA | ERNAM | LAEDA | AENAM | VPSTA | PSTAT | LVORM | HTART | MBRSR | MATKL |
|-------|-------|------------|-------|------------|-------|-------|-------|-------|-------|-------|-------|
| 800 | DEWW | 24.10.2008 | KDT | 00.00.0000 | | K | K | | FGTR | F | |

© SAP AG. All rights reserved.

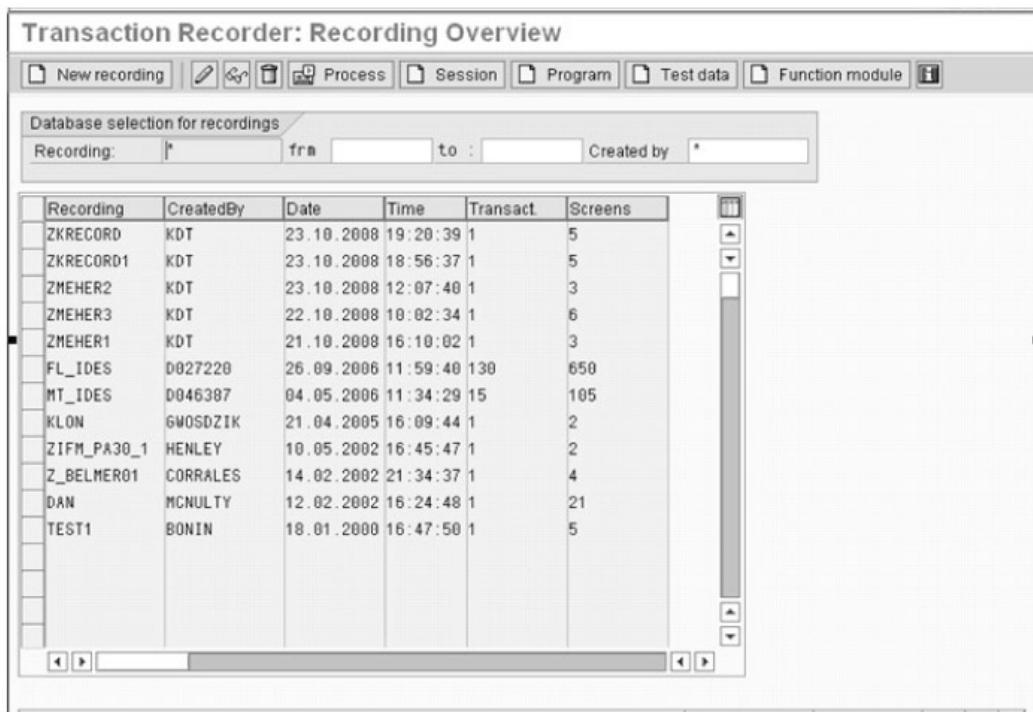
Figure 11.34: The screen showing the transferred DEWW material in the SAP system

Note You can follow the same procedure to transfer the FANTAA data as well.

Transferring Data by Using the Batch Input with Session Method of BDC

The batch input with session method of BDC is used to transfer data to a transaction with the help of a session. In this case, we transfer data to ABAP Dictionary (transaction code SE11). The data is transferred to the ZSUBJECT_DATA database table. Perform the following steps to learn the procedure of data transfer:

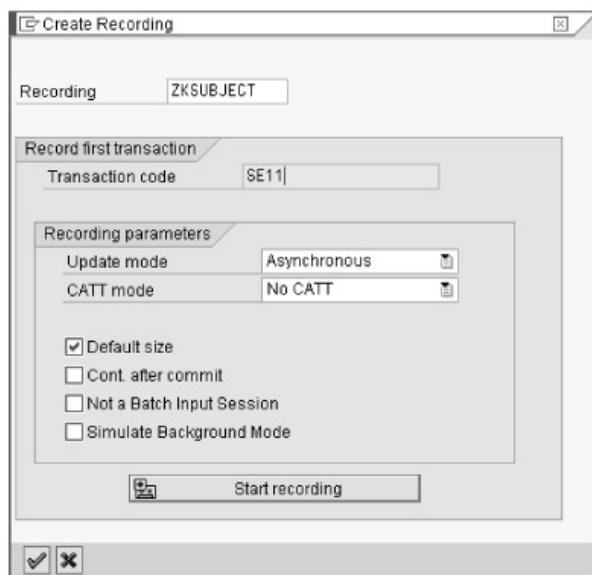
1. Start the initial screen of the batch input recorder by using the SHDB transaction code. The Transaction Recorder: Recording Overview screen appears, displaying a list of all the previously generated recordings, as shown in [Figure 11.35](#):
2. Click the New recording button, as shown in [Figure 11.35](#).



© SAP AG. All rights reserved.

Figure 11.35: The initial screen of recording

The Create Recording dialog box appears, as shown in [Figure 11.36](#):



© SAP AG. All rights reserved.

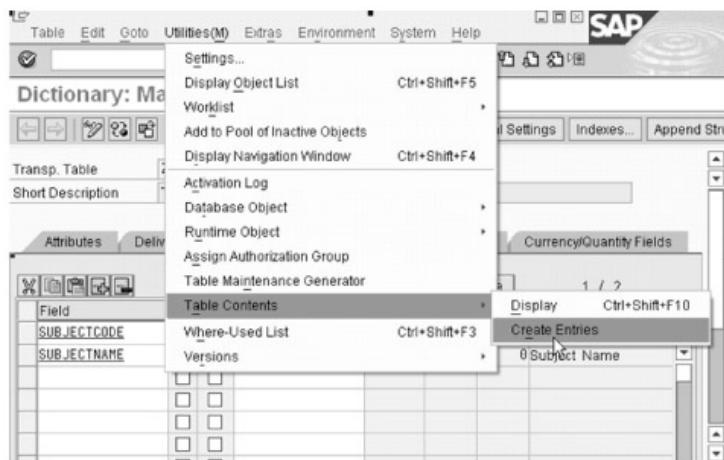
Figure 11.36: Creating the recording screen

- Enter the name of the recording, say, ZKSUBJECT, and transaction code as SE11. Click the Start recording button, as shown in [Figure 11.36](#).

The initial screen of the ABAP Dictionary appears.

- Enter the name of the table as ZSubject_Data in the Database table field and then click the Change pushbutton. A message, Recording is running, appears in the status bar. The Dictionary: Maintain Table screen appears, as shown in [Figure 11.37](#):
- Click Utilities (M) > Table Contents > Create Entries to create an entry in the ZSubject_Data table

(see Figure 11.37).



© SAP AG. All rights reserved.

Figure 11.37: Creating an entry

The Table ZSUBJECT_DATA Insert screen appears, as shown in Figure 11.38:

 A screenshot of the Table ZSUBJECT_DATA Insert screen. The title bar says 'Table ZSUBJECT_DATA Insert'. There are two input fields: 'SUBJECTCODE' containing 'A11' and 'SUBJECTNAME' containing 'Economy'. A 'Reset' button is also visible.

© SAP AG. All rights reserved.

Figure 11.38: Making an entry in the table ZSUBJECT_DATA

6. Enter A11 in the SUBJECTCODE field and Economy in the SUBJECTNAME field. Click the Save (□) icon (see Figure 11.38).
7. Click the Back (◀) icon (see Figure 11.38).

The Transaction Recorder: Change Recording ZKSUBJECT screen appears, as shown in Figure 11.39:

| | Program | Screen | St | Field name | Field value |
|----|----------|--------|------|----------------|---|
| 1 | | T | SE11 | | |
| 2 | SAPMSRDO | 0102 | X | | |
| 3 | | | | BDC_CURSOR | RSRD1-TBMA_VAL |
| 4 | | | | BDC_OKCODE | =EDIT |
| 5 | | | | RSRD1-TBMA | X |
| 6 | | | | RSRD1-TBMA_VAL | ZSubject_Data |
| 7 | SAPLSD41 | 2200 | X | | |
| 8 | | | | BDC_CURSOR | DD02D-DDTEXT |
| 9 | | | | BDC_OKCODE | =TDED |
| 10 | | | | DD02D-DDTEXT | Table Containing Subject Code with Subject Name |
| 11 | | | | BDC_SUBSCR | SAPLSD41 2201TS_SCREEN |
| 12 | SAPLSP01 | 0300 | X | | |
| 13 | | | | BDC_OKCODE | =CANC |
| 14 | SAPLSD41 | 2200 | X | | |
| 15 | | | | BDC_CURSOR | DD02D-DDTEXT |
| 16 | | | | BDC_OKCODE | =WB_SAVE |
| 17 | | | | DD02D-DDTEXT | Table Containing Subject Code with Subject Name |
| 18 | | | | BDC_SUBSCR | SAPLSD41 2201TS_SCREEN |

© SAP AG. All rights reserved.

Figure 11.39: The recording screen

Figure 11.39 shows the description of the screen, such as the screen name in the Program column, the number associated with each screen under the Screen column, the status under the Status column, and the value of each field under the Field value column.

- Click the Save () icon to save this recording (see Figure 11.39).

The message Recording was saved appears on the Status bar.

- Click the Back () icon (see Figure 11.39).

The Transaction Recorder: Recording Overview screen appears, as shown in Figure 11.40:

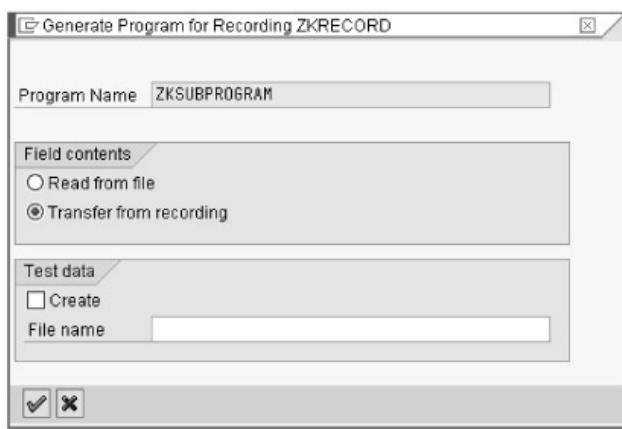
- Select the created recording (that is, ZKSUBJECT) and then click the Program button, as shown in Figure 11.40.

| Recording | CreatedBy | Date | Time | Transact. | Screens |
|-----------|-----------|------------|----------|-----------|---------|
| ZKSUBJECT | KDT | 26.10.2008 | 10:26:04 | 1 | 14 |

© SAP AG. All rights reserved.

Figure 11.40: Overview of the created recording

The Generate Program for Recording ZKRECORD dialog box appears, as shown in Figure 11.41:

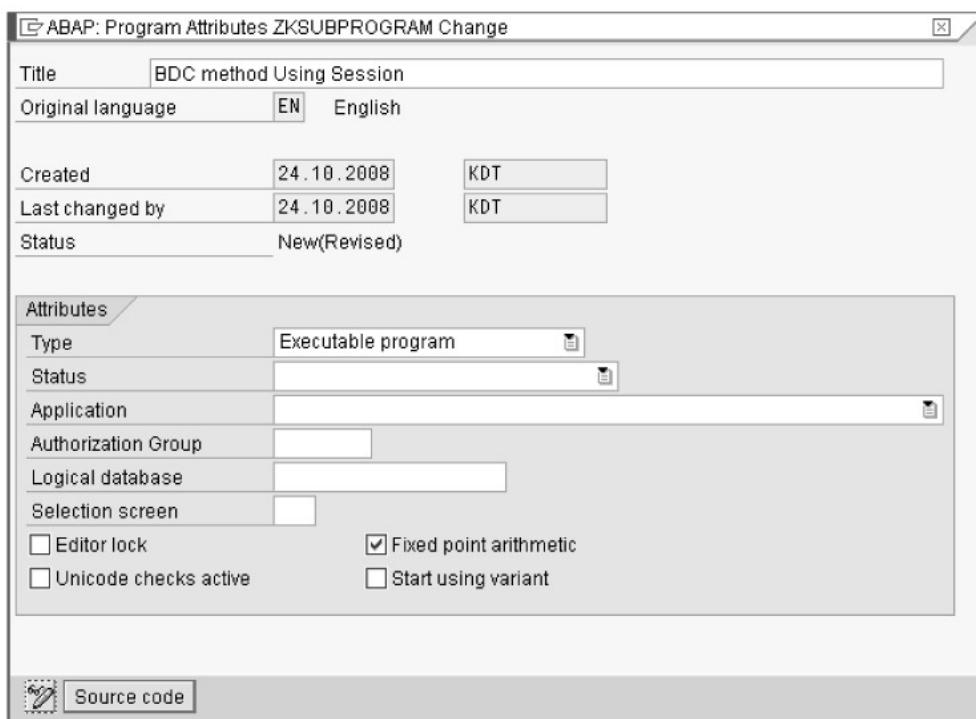


© SAP AG. All rights reserved.

Figure 11.41: The generate program for recording ZKRECORD

11. Enter the name of the program, ZKSUBPROGRAM, in the Program Name field and select the Transfer from recording radio button. Click the Continue (✓) icon (see Figure 11.41).

The ABAP: Program Attributes ZKSUBPROGRAM Change dialog box appears, as shown in Figure 11.42:



© SAP AG. All rights reserved.

Figure 11.42: The attributes screen related to the BDC program

Figure 11.42 displays the screen where you assign the attributes related to the ZKSUBPROGRAM program.

12. Enter BDC method Using Session in the Title field and Executable program in the Type field.
13. Click the Source code button and save the program in a package. In this case, we have saved the program in a ZKOG_PCKG package.

The ABAP Editor Change screen appears.

Repeat steps 12 to 20 used in the BDC method by using the call transaction method. Note that when the internal table is declared, it includes only two fields because we transfer the data related to only two fields of the ZSUBJECT_DATA table. Figure 11.43 shows the internal table and BDCDATA structure:

```

Report ZKSUBPROGRAM
no standard page heading line-size 255.

*include bdcrecx1.
*
*start-of-selection.
*
*perform open_group.

Data: begin of it occurs 0,
      subjectcode like zsubject_data-subjectcode,
      subjectname like zsubject_data-subjectname,
      end of it.

data : bdcdatalike bdcdatalike 0 with header line.

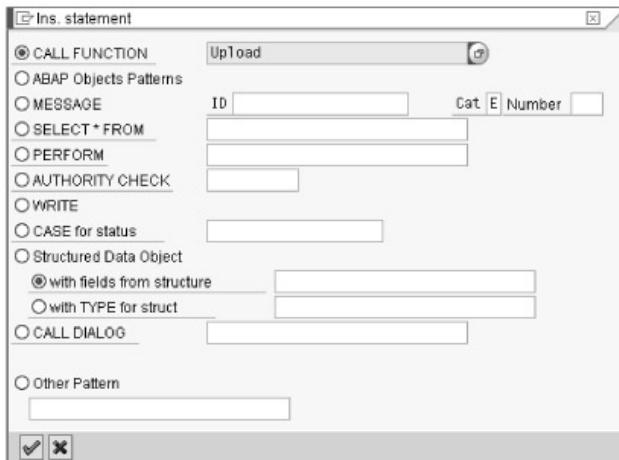
```

© SAP AG. All rights reserved.

Figure 11.43: Including the internal table and BDCDATA structure

- Click the Pattern button in ABAP Editor (see [Figure 11.43](#)).

The Ins. statement dialog box appears, as shown in [Figure 11.44](#):



© SAP AG. All rights reserved.

Figure 11.44: Uploading function module

- Enter the name of the function module, Upload, which is to be uploaded in the ZKSUBPROGRAM program ([Figure 11.44](#)).

The definition regarding the UPLOAD function module appears in the code, as shown in [Figure 11.45](#):

```

Report ZKSUBPROGRAM Active (Revised)
Data: begin of it occurs 0,
  subjectcode like zsubject_data-subjectcode,
  subjectname like zsubject_data-subjectname,
  end of it.

data : bdcdata like bdcdata occurs 0 with header line.

CALL FUNCTION 'UPLOAD'
EXPORTING
*  CODEPAGE          = ''
*  FILENAME          = ''
*  FILETYPE          = 'DAT'
*  ITEM              = ''
*  FILEMASK_MASK    = ''
*  FILEMASK_TEXT    = ''
*  FILETYPE_NO_CHANGE = ''
*  FILEMASK_ALL     = ''
*  FILETYPE_NO_SHOW  = ''
*  LINE_EXIT         = ''
*  USER_FORM         = ''
*  USER_PROG         = ''
*  SILENT            = 'S'
*  IMPORTING
*    FILESIZE        =
*    CANCEL          =
*    ACT_FILENAME    =
*    ACT_FILETYPE   =
TABLES
  DATA_TAB           = IT
*  EXCEPTIONS
*    CONVERSION_ERROR = 1
*    INVALID_TABLE_WIDTH = 2
*    INVALID_TYPE     = 3
*    NO_BATCH         = 4

```

© SAP AG. All rights reserved.

Figure 11.45: The UPLOAD function module

16. Uncomment the EXPORTING, FILENAME, and FILETYPE parameters by deleting the asterisk symbol. Write the name of the internal table in the DATA_TAB parameter (see [Figure 11.45](#)).
17. Click the Pattern button.

The `Ins.` statement screen appears again, as shown in [Figure 11.44](#). Now, we include a function module named `BDC_OPEN_GROUP`, which is used to create a new session. [Figure 11.46](#) shows the definition of the `BDC_OPEN_GROUP` function module in the editor screen:

```

Report ZKSUBPROGRAM Active (Revised)
CALL FUNCTION 'BDC_OPEN_GROUP'
EXPORTING
*  CLIENT          = SY-MANDT
*  DEST            = FILLER8
*  GROUP           = 'ZKSUBPROGRAM'
*  HOLDDATE        = FILLER8
*  KEEP             = 'X'
*  USER             = SY-UNAME
*  RECORD           = FILLER1
*  PROG             = SY-CPROG
*  DCPFM            = '%'
*  DATFM            = '%'
*  IMPORTING
*    QID             =
*  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.
*  MESSAGE ID SY-MSGID TYPE SY-MS6TY NUMBER SY-MSGNO
*  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

```

© SAP AG. All rights reserved.

Figure 11.46: The BDC_OPEN_GROUP function module

18. Now, uncomment the EXPORTING, GROUP, KEEP, USER, and PROG parameters. Enter the name of the program, ZKSUBPROGRAM, for the GROUP parameter, X for the KEEP parameter, SY-UNAME for the USER parameter, and SY-CPROG for the PROG parameter, as shown in [Figure 11.46](#).
19. Start a loop for the created internal table named IT to transfer the data one by one in an SAP system. Now, call a function named BDC_INSERT after writing the PERFORM statement by clicking the Pattern button on the application toolbar. Write the function name BDC_INSERT in the CALL FUNCTION field. The BDC_INSERT function module inserts the SE11 transaction code to the batch input session. [Figure 11.47](#) shows the definition of the BDC_INSERT function module:
20. In the definition of the BDC_INSERT function module, uncomment the EXPORTING, TCODE, TABLES, and DYNPROTAB parameters and include the function module named BDC_CLOSE_GROUP.

```

Report ZKSUBPROGRAM Active (Revised)
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

LOOP AT IT.
REFRESH BODATA.
PERFORM MAPPINGS.
CALL FUNCTION 'BDC_INSERT'
  EXPORTING
    TCODE          = 'SE11'
    POST_LOCAL     = NOBLOCAL
    PRINTING       = NOPRINT
    SIMUBATCH     = ''
    CTUPARAMS      = ''
  TABLES
    DYNPROTAB     = BODATA
  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.
    MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
  ENDIF.
ENDLOOP.

```

© SAP AG. All rights reserved.

Figure 11.47: The function BDC_INSERT is called

[Figure 11.48](#) shows the code with the definition of the BDC_CLOSE_GROUP function module:

```

Report ZKSUBPROGRAM Active (Revised)
* 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.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
ENDLOOP.

CALL FUNCTION 'BDC_CLOSE_GROUP'
* EXCEPTIONS
* NOT_OPEN        = 1
* QUEUE_ERROR     = 2
* OTHERS          = 3

IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
FORM MAPPING.

perform bdc_dynpro  using 'SAPMSRDO' '0102'.

```

© SAP AG. All rights reserved.

Figure 11.48: The BDC_CLOSE_GROUP function module

21. Now, type FORM MAPPING after the ENDIF statement, as shown in [Figure 11.48](#). In the figure, FORM represents a type of statement and MAPPING represents the subroutine name. Write the ENDFORM statement in the ABAP Editor: Change Report ZKSUBPROGRAM screen, at the place shown in [Figure 11.17](#).
22. Replace values A11 and Economy with the respective internal table fields named it-subjectcode and it-subjectname inside the definition of FORM MAPPING, as shown in [Figure 11.49](#):
23. Click the Save () icon, the Check () icon, and the Activate () icon to activate the program.
24. Now, create a notepad file and store it in a particular location.

```

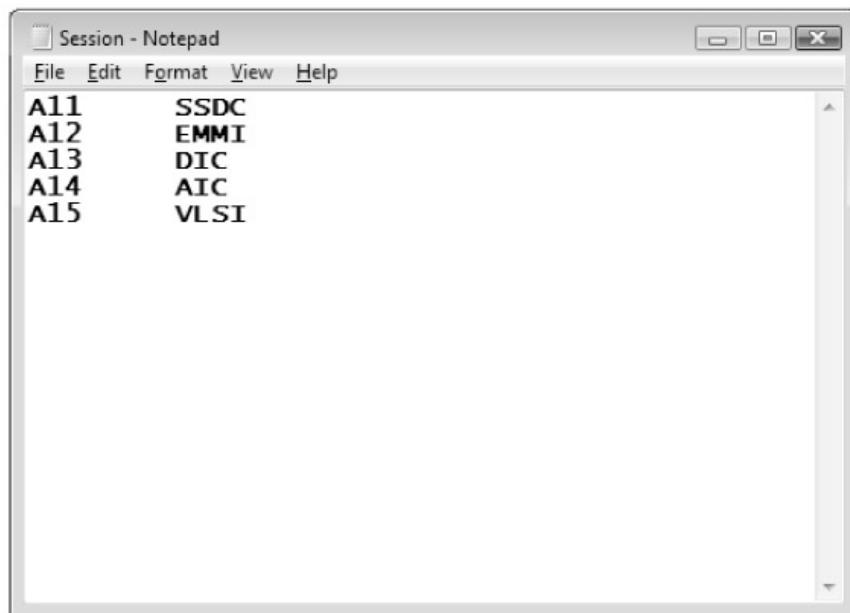
ABAP Editor: Change Report ZKSUBPROGRAM
Report ZKSUBPROGRAM Active
perform bdc_dynpro      using '/IBCDWB/DBZSUBJECT_DATA' '0101'.
perform bdc_field       using 'BDC_CURSOR'
perform bdc_field       using 'ZSUBJECT_DATA-SUBJECTCODE'.
perform bdc_field       using 'BDC_OKCODE'
           '=SAVE'.
perform bdc_field       using 'ZSUBJECT_DATA-SUBJECTCODE'
           it-subjectcode.
perform bdc_field       using 'ZSUBJECT_DATA-SUBJECTNAME'
           it-subjectname.
perform bdc_dynpro      using '/IBCDWB/DBZSUBJECT_DATA' '0101'.
perform bdc_field       using 'BDC_CURSOR'
           '/EBACK'.
perform bdc_field       using 'BDC_CURSOR'
           'ZSUBJECT_DATA-SUBJECTCODE'.
perform bdc_dynpro      using 'SAPLSD41' '2200'.
perform bdc_field       using 'BDC_CURSOR'
           'D0020-DTEXT'.
perform bdc_field       using 'BDC_OKCODE'
           '=WB_BACK'.
perform bdc_field       using 'D0020-DTEXT'
           'Table Containing Subject Code with Sub'
           & 'ject Name'.
perform bdc_dynpro      using 'SAPMSR00' '0102'.
perform bdc_field       using 'BDC_CURSOR'
           'RSRD1-TBMA_VAL'.
perform bdc_field       using 'BDC_OKCODE'
           '=BACK'.
perform bdc_field       using 'RSRD1-TBMA'
           'X'.
perform bdc_field       using 'RSRD1-TBMA_VAL'
           'ZSUBJECT_DATA'.
*perform bdc_transaction using 'SE11'.
*perform bdc_close_group.
ENDFORM.

```

© SAP AG. All rights reserved.

Figure 11.49: Changing the content with the respective field names of the internal table

In this case, we create a notepad file session and store it on the desktop. This file contains data related to the subject code and subject name. The content of this file has to be transferred into the SAP system. [Figure 11.50](#) shows the data contained in the session file:



© SAP AG. All rights reserved.

Figure 11.50: The notepad file

The session file consists of five subject names with their corresponding subject codes.

25. Click the Direct Processing (█) icon (see [Figure 11.49](#)).

The Import from a Local File dialog box appears, as shown in [Figure 11.51](#):



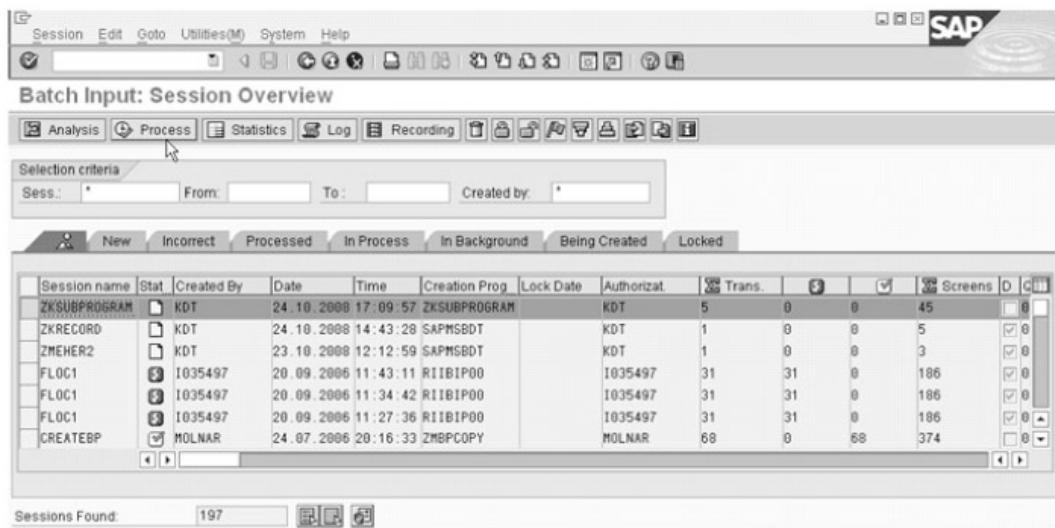
© SAP AG. All rights reserved.

Figure 11.51: Import from a local file

26. Enter the location of the notepad file that is to be transferred into the SAP system and click the Transfer button (see [Figure 11.51](#)).

The message Byte transferred appears in the status bar.

27. Execute the transaction in a new session to check whether the batch input session has been created. [Figure 11.52](#) shows the Batch Input: Session Overview screen, which displays the ZKSUBPROGRAM session name:



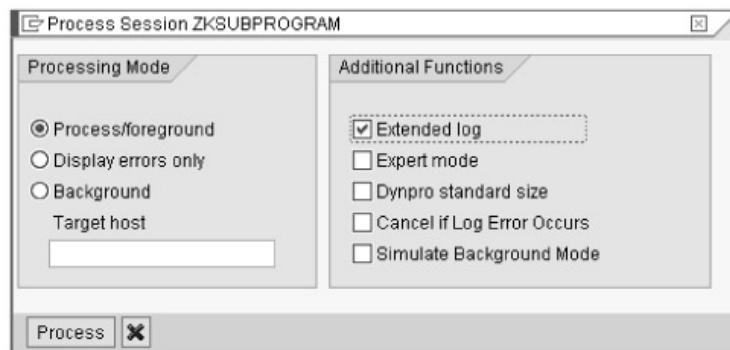
© SAP AG. All rights reserved.

Figure 11.52: The batch input session overview screen

Figure 11.52 shows the transaction details, such as the name of the user under the **Created By** column, date on which the program was created under the **Date** column, the time at which the session was created under the **Time** column, and authorization information under the **Authorization** column.

28. Select the name of the session. In this case, we select ZKSUBPROGRAM, as shown in Figure 11.52.
29. Click the **Process** pushbutton (see Figure 11.52).

The **Process Session ZKSUBPROGRAM** dialog box appears, as shown in Figure 11.53:

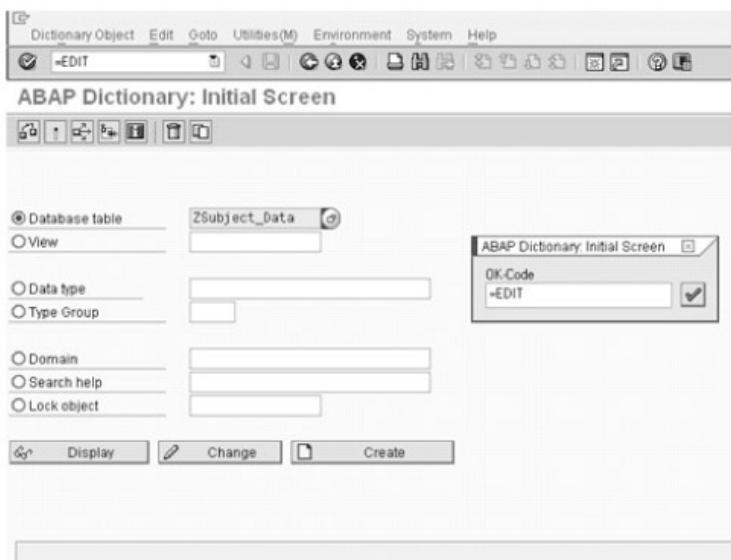


© SAP AG. All rights reserved.

Figure 11.53: Processing the created session

30. Select the **Process/foreground** radio button and the **Extended log** check box. Click the **Process** button, as shown in Figure 11.53.

The initial screen of the ABAP Dictionary appears, as shown in Figure 11.54:



© SAP AG. All rights reserved.

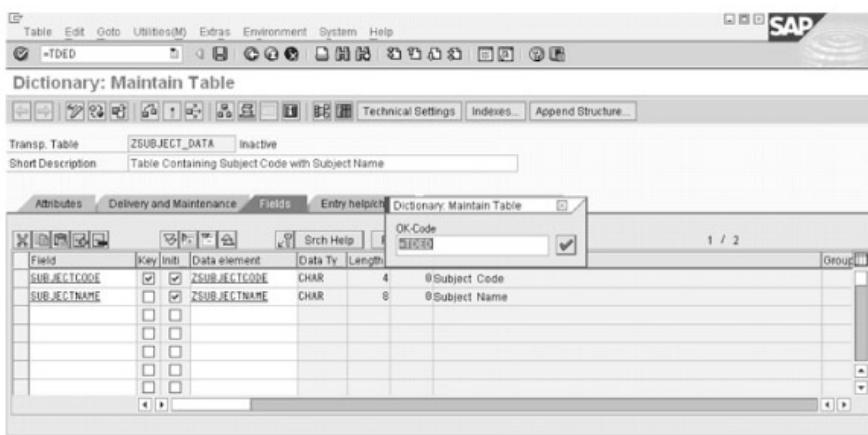
Figure 11.54: The initial screen of the ABAP dictionary

In the initial screen of ABAP Dictionary, the name of the table, ZSubject_Data, automatically appears in the Database table field. The associated OK-Code field (=EDIT) for the screen also appears.

31. Click the Continue  icon.

The Dictionary: Maintain Table screen appears, as shown in [Figure 11.55](#):

[Figure 11.55](#) displays the short description related to the ZSUBJECT_DATA table.

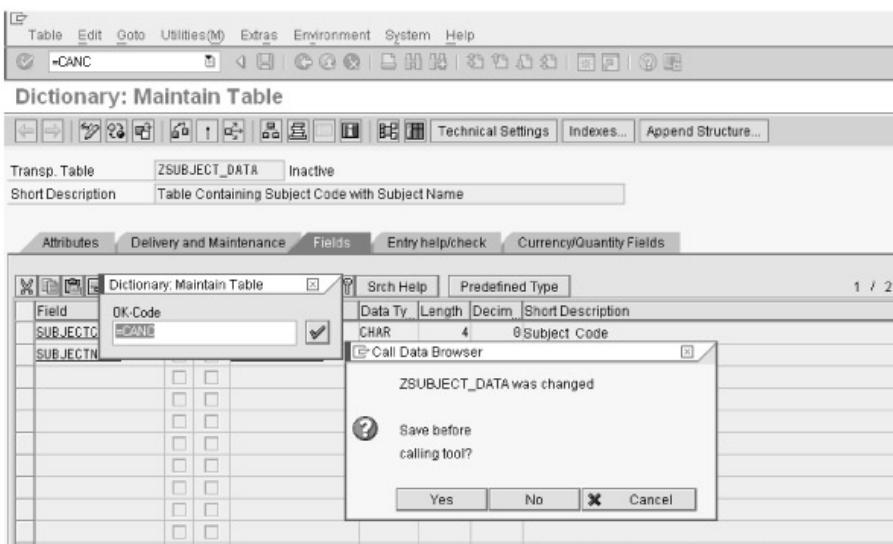


© SAP AG. All rights reserved.

Figure 11.55: The screen showing the fields of the table

32. Click the Continue  icon.

The Call Data Browser dialog box appears asking whether you want to save the ZSUBJECT_DATA table. [Figure 11.56](#) displays the Call Data Browser dialog box:



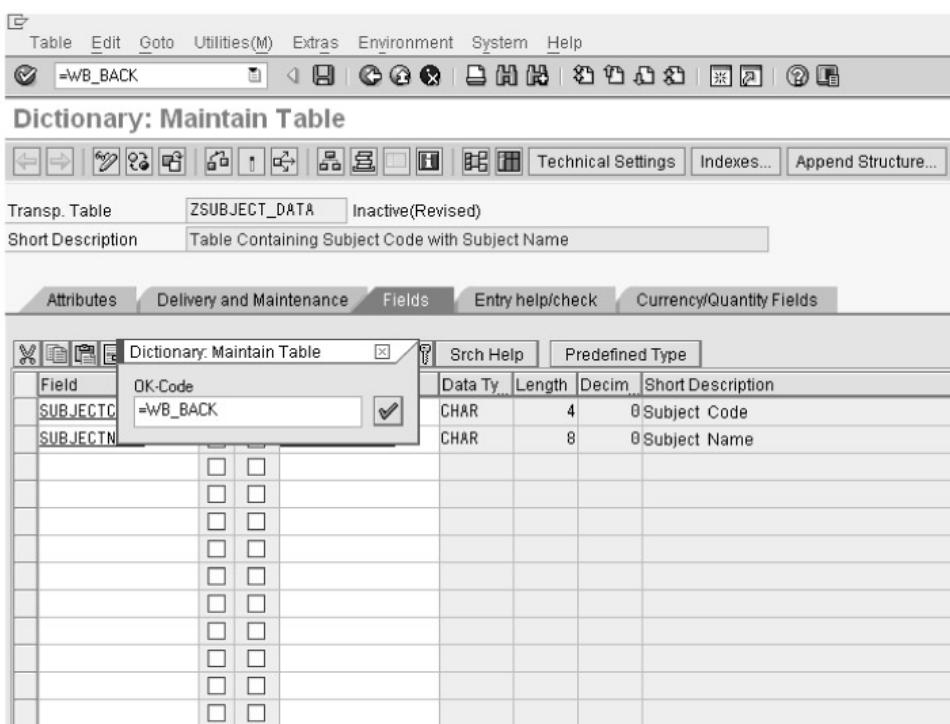
© SAP AG. All rights reserved.

Figure 11.56: The Ok-Code field for the maintain table

The value of the OK-Code for the screen, =CANC, also appears (see Figure 11.56).

33. Click the Continue (checkbox) icon.

The Dictionary: Maintain Table screen appears, as shown in Figure 11.57:



© SAP AG. All rights reserved.

Figure 11.57: The screen after saving with different OK-Code fields

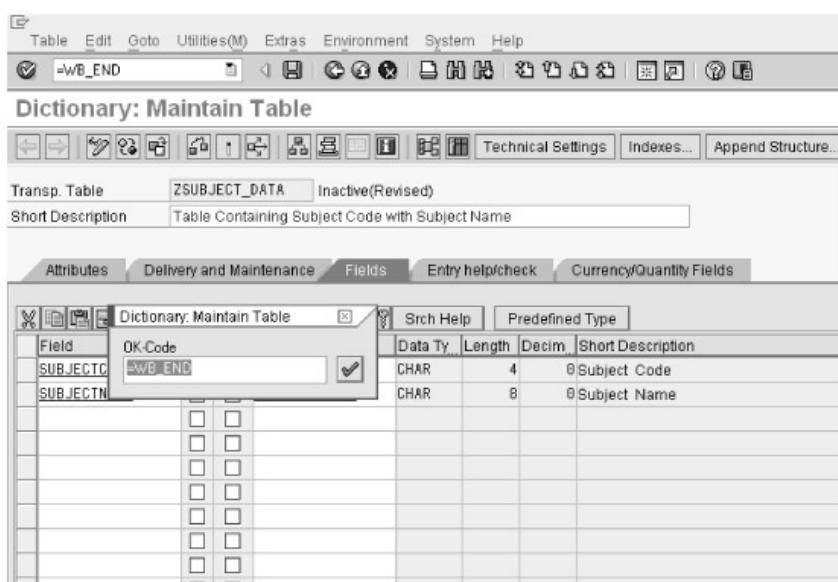
The value of the OK-Code for the screen, WB_BACK, is displayed.

34. Click the Continue (checkbox) icon.

The value of the OK-Code for the screen, =CANC, appears, as shown in Figure 11.56.

35. Click the Continue (checkbox) icon.

The Dictionary: Maintain Table dialog box appears, as shown in Figure 11.58:



© SAP AG. All rights reserved.

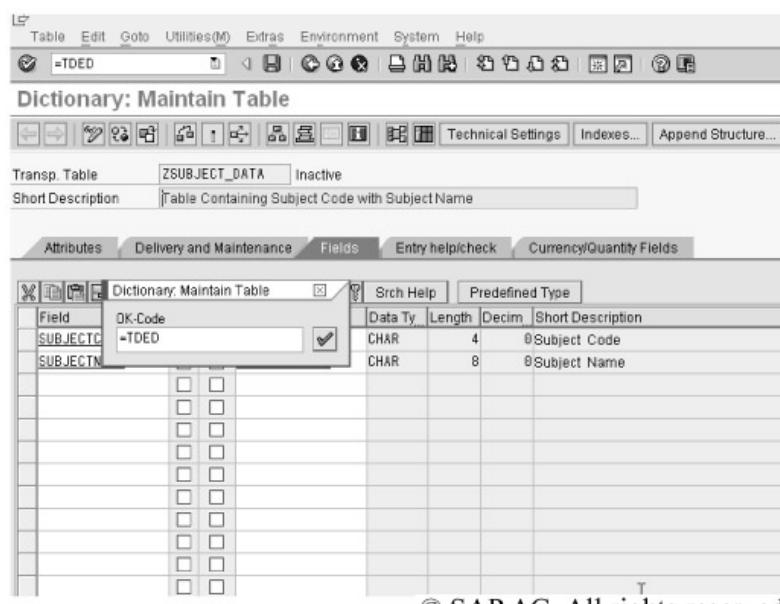
Figure 11.58: The Ok-Code field

36. Click the Continue (checkbox) icon.

The Call Data Browser dialog box appears, as shown in Figure 11.56.

37. Click the Continue (checkbox) icon for the screen containing the OK-Code field as =CANC (see Figure 11.56).

Now, the Dictionary: Maintain Table screen appears, as shown in Figure 11.59:



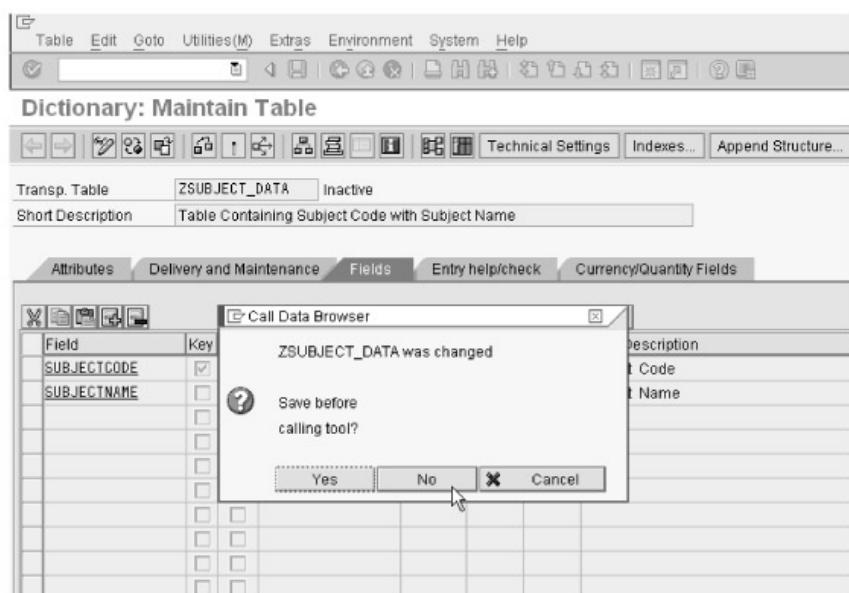
© SAP AG. All rights reserved.

Figure 11.59: The screen showing the Ok-Code (=TDED)

In Figure 11.59, the OK-Code field for the screen appears as =TDED.

38. Click the Continue (checkbox) icon.

The Call Data Browser dialog box appears, as shown in [Figure 11.60](#):



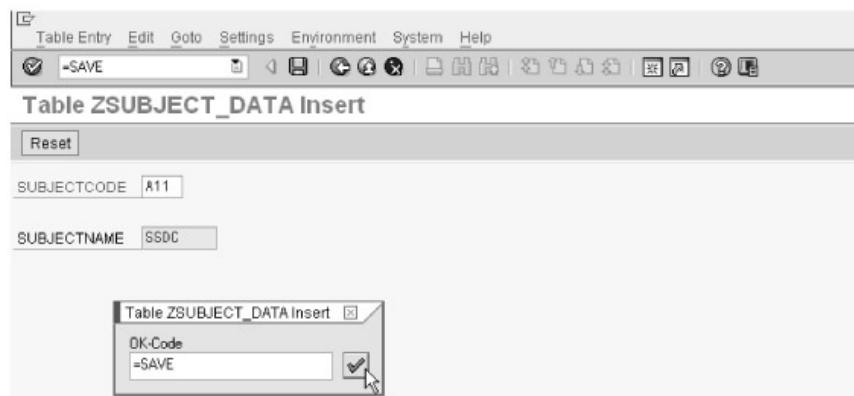
© SAP AG. All rights reserved.

Figure 11.60: The call data browser screen

A dialog box appears asking whether or not you want to save the ZSUBJECT_DATA table.

39. Click the No button, because we do not want to save the table (see [Figure 11.60](#)).

The Table ZSUBJECT_DATA Insert screen appears along with its OK-Code field, as shown in [Figure 11.61](#):



© SAP AG. All rights reserved.

Figure 11.61: Inserting first data from the notepad file into the database table

In [Figure 11.61](#), you see that the first entries related to the subject code and subject name are picked from the session notepad file and are inserted into the ZSUBJECT_DATA table.

40. Click the Continue (checkbox) icon for the OK-Code field (=SAVE).

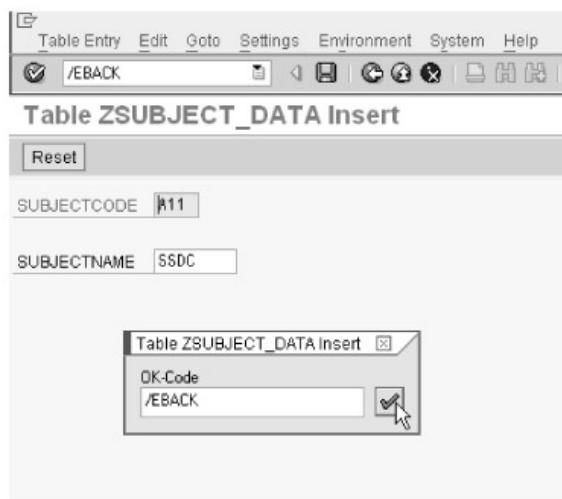
The message Database record successfully created appears in the status bar, as shown in [Figure 11.62](#):



© SAP AG. All rights reserved.

Figure 11.62: Status bar

The value of OK-Code, /EBACK, is displayed, as shown in Figure 11.63:



© SAP AG. All rights reserved.

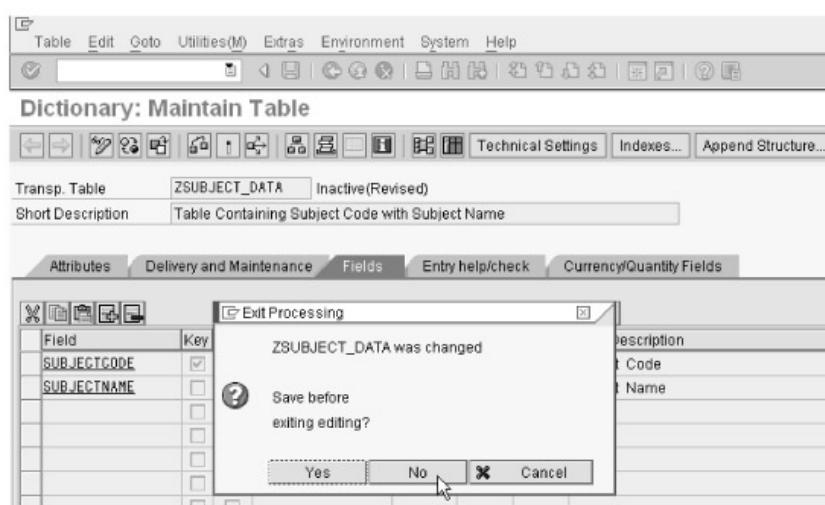
Figure 11.63: The screen showing the Ok-Code as /EBACK

41. Click the Continue (checkbox) icon.

The value of OK-Code for the screen, =WB_BACK, appears, as shown in Figure 11.57.

42. Click the Continue (checkbox) icon.

The Exit Processing dialog box appears asking whether you want to save the ZSUBJECT_DATA table, as shown in Figure 11.64:

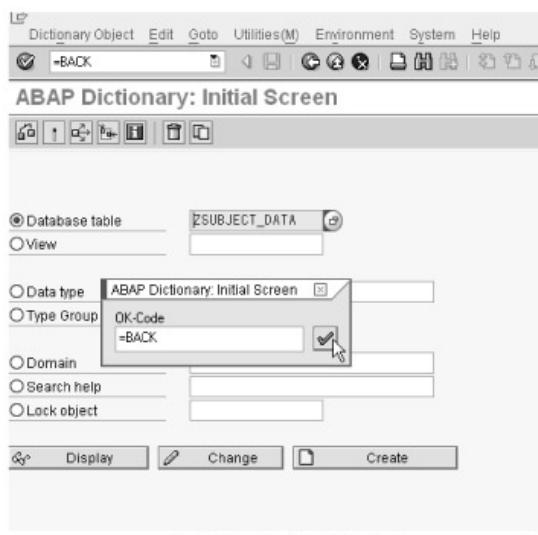


© SAP AG. All rights reserved.

Figure 11.64: The exit processing dialog box

43. Click the No button (see Figure 11.64).

The Initial screen of the ABAP Dictionary appears, as shown in Figure 11.65:



© SAP AG. All rights reserved.

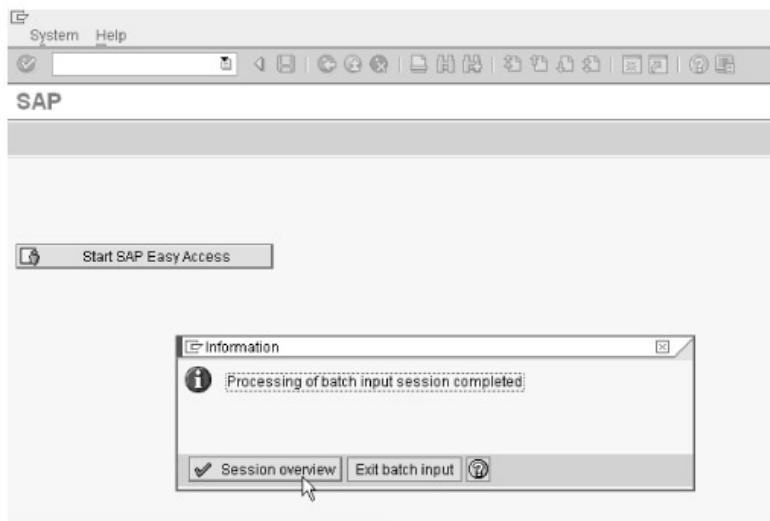
Figure 11.65: The ABAP dictionary initial screen

In the OK-Code field, the value, =BACK, for the screen is displayed.

44. Click the Continue (checkbox) icon.

45. Now, repeat Steps 30 to 43 to transfer all the records mentioned in the session notepad file into the SAP system.

An information dialog box appears after the successful transfer of the data from the notepad file to the ZSUBJECT_DATA table, as shown in [Figure 11.66](#):



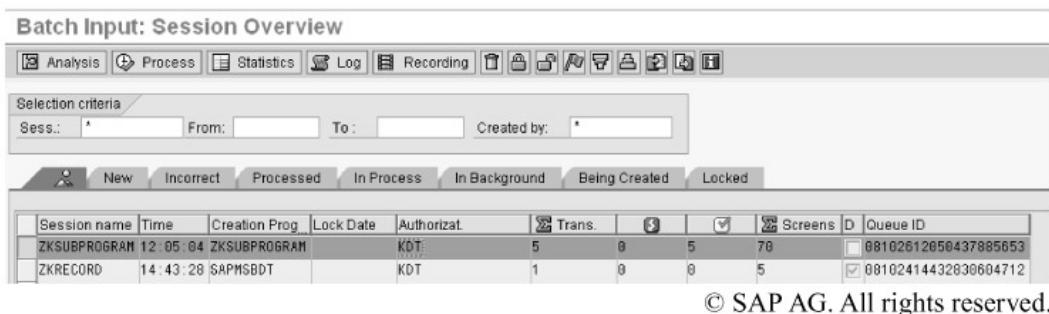
© SAP AG. All rights reserved.

Figure 11.66: Information dialog box

46. Click the Session overview button to view the generated session, as shown in [Figure 11.66](#).

The Batch Input: Session Overview screen appears, as shown in [Figure 11.67](#):

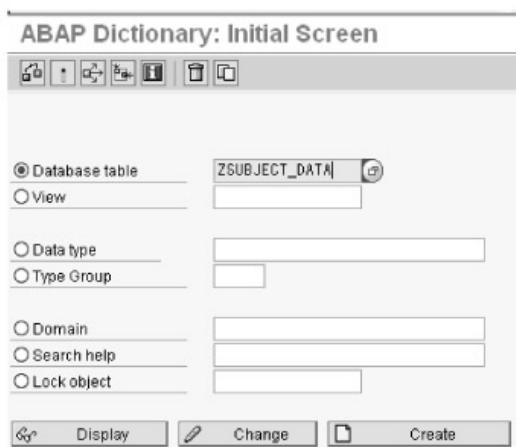
[Figure 11.67](#) displays the information related to the processed session, such as the session name, the time at which the session was processed, the name of the authorized user, the total number of transactions involved, the number of transactions processed, and the total number of screens required to process the session.



© SAP AG. All rights reserved.

Figure 11.67: Session overview

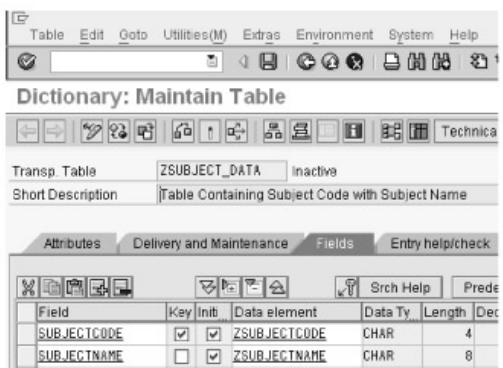
47. Now, to check whether the transactions were made in the ZSUBJECT_DATA table, open a new session and then start the ABAP Dictionary by using the SE11 transaction code. **Figure 11.68** displays the initial screen of the ABAP Dictionary:
48. Enter ZSUBJECT_DATA in the Database table field and click the Change button, as shown in **Figure 11.68**.



© SAP AG. All rights reserved.

Figure 11.68: The ABAP dictionary initial screen

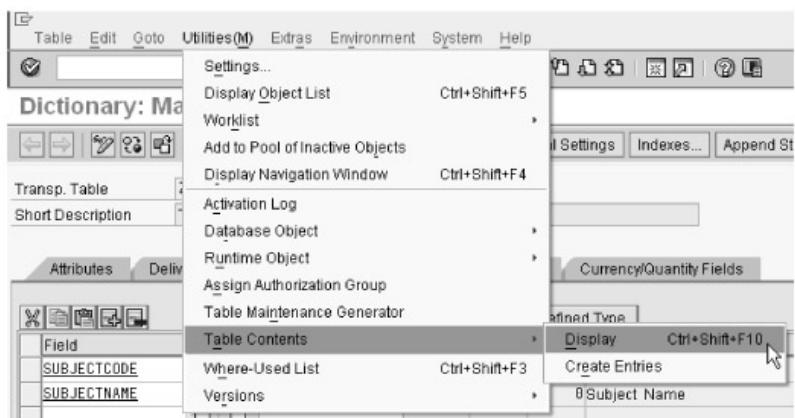
The Dictionary: Maintain Table screen appears, displaying the SUBJECTCODE and SUBJECTNAME fields of the ZSUBJECT_DATA table, as shown in **Figure 11.69**:



© SAP AG. All rights reserved.

Figure 11.69: Fields of the database table

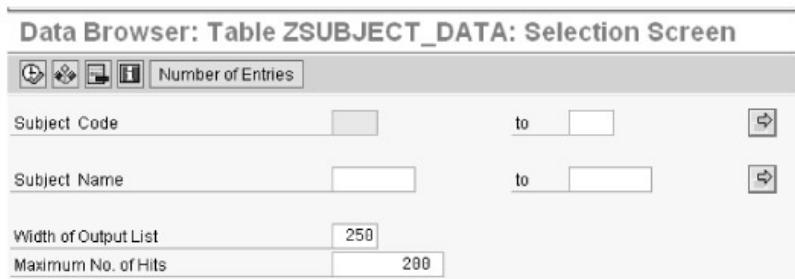
49. Select Utilities(M) > Table Contents > Display, as shown in **Figure 11.70**:



© SAP AG. All rights reserved.

Figure 11.70: Displaying the contents of ZSUBJECT_DATA

This action is used to display the contents of the ZSUBJECT_DATA table, to check whether the data from the notepad file has been transferred to the SAP system. The selection screen of Data Browser for the ZSUBJECT_DATA table appears, as shown in [Figure 11.71](#):



© SAP AG. All rights reserved.

Figure 11.71: The selection screen

50. Click the Execute (⊕) icon (see [Figure 11.71](#)).

The Data Browser: Table ZSUBJECT_DATA Select Entries screen appears, as shown in [Figure 11.72](#):

| Table: ZSUBJECT_DATA | |
|---|-------------|
| Displayed Fields: 2 of 2 Fixed Columns: | |
| SUBJECTCODE | SUBJECTNAME |
| A01 | HINDI |
| A02 | ENGLISH |
| A03 | MATHS |
| A04 | PHYSICS |
| A05 | BIOLOGY |
| A06 | HISTORY |
| A07 | STATS |
| A08 | SUPW |
| A09 | CHEM |
| A10 | SANSKRIT |
| A11 | SSDC |
| A12 | EMMI |
| A13 | DIC |
| A14 | AIC |
| A15 | VLSI |

© SAP AG. All rights reserved.

Figure 11.72: Data in table

In [Figure 11.72](#), you can see that the data of the notepad file has been transferred to the ZSUBJECT_DATA table.

Now, let's learn about another technique of transferring data into an SAP system: LSMW.

The LSMW Tool

Legacy System Migration Workbench (LSMW) is a tool used to transfer data from a legacy system to an SAP system. The data can be transferred either all at once or periodically at regular intervals. The LSMW tool, a cross-application component of the SAP system, first reads the legacy data from the spreadsheet tables or sequential files and then converts it to a format supported by the target system. Finally, the converted data is imported in the database used by the SAP system. Use the LSMW transaction code to access the LSMW interface.

You must consider the following points before you start working with the LSMW tool:

- LSMW is used only after the SAP system has been installed and application customization has been completed.
- The data in the legacy system is analyzed.
- The transactions required to transfer the data into an SAP system must be identified.
- You must process the identified transactions in the SAP system manually by using test data from the legacy system.
- The source and SAP fields must be mapped to each other.
- You must define the conversion rule, which uses the field content of the source structure and is then converted into the target structure fields.
- You must define the way in which data is extracted from the legacy system.
- You can either use any of the standard import techniques or generate a recording of the transaction with the batch input recording method.

As no prior knowledge of programming is required to work with LSMW, this tool can be employed by functional consultants directly, whereas the BDC tool can be used only by technical people because it requires programming knowledge. With the help of LSMW, you can transfer new records as well as update the existing records in an SAP system.

Now, let's learn how to update a customer's master record with LSMW by using the batch input recording method.

Updating Records by Using the Batch Input Recording Method

Sometimes, to cater to the needs of different customers in a more efficient manner, the data related to the customer master records of all the customers are linked. In terms of SAP, the linking of the customer data implies that we only change the data of the sales office, sales group, and customer group related to the customer master records.

Perform the following steps to update a customer master record by using the batch input recording method of LSMW:

1. Start the initial screen of the LSMW tool by entering `I_SMW` transaction code in the Command field of the SAP Easy Access screen, as shown in [Figure 11.73](#):
2. Press the ENTER key or click the Continue () icon.

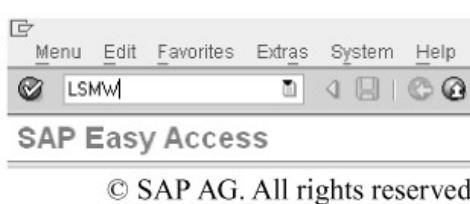
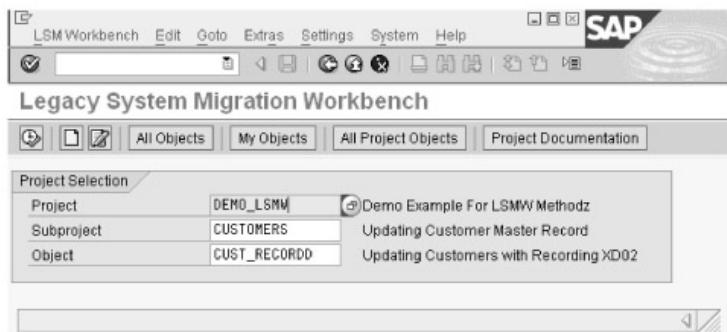


Figure 11.73: Starting LSMW

The Legacy System Migration Workbench screen appears, as shown in [Figure 11.74](#):



© SAP AG. All rights reserved.

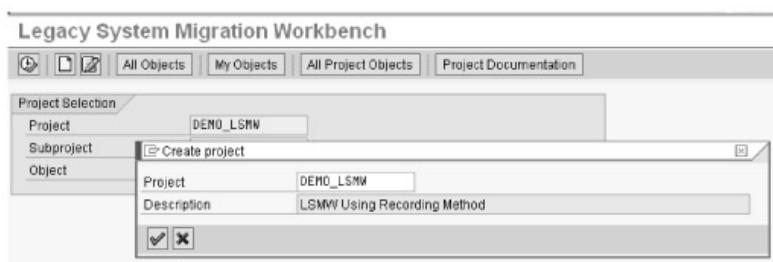
Figure 11.74: Legacy system migration workbench screen

In Figure 11.74, notice that three structures named Project, Subproject, and Object are provided.

We create a project named DEMO_LSMW next.

3. Enter DEMO_LSMW in the Project field and click the Create Entry (□) icon, as shown in Figure 11.74.

The Create project dialog box appears, as shown in Figure 11.75:

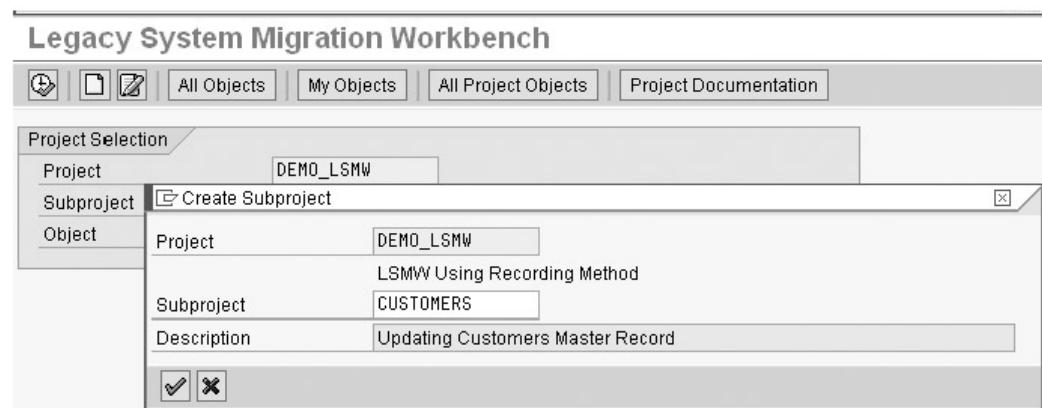


© SAP AG. All rights reserved.

Figure 11.75: Creating a project

4. Enter LSMW Using Recording Method in the Description field and click the Continue (✓) icon (see Figure 11.75).

The Create Subproject dialog box appears, as shown in Figure 11.76:

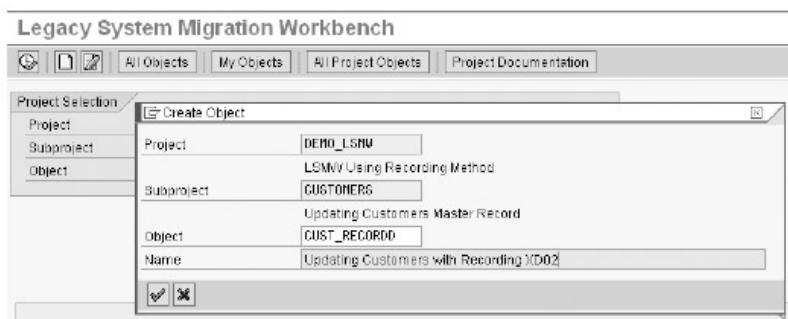


© SAP AG. All rights reserved.

Figure 11.76: Create the subproject screen

5. Enter the name of the subproject as CUSTOMERS in the Subproject field, enter Updating Customer Master Record in the Description field, and click the Continue (✓) icon (see Figure 11.76).

The Create Object dialog box appears, as shown in [Figure 11.77](#):

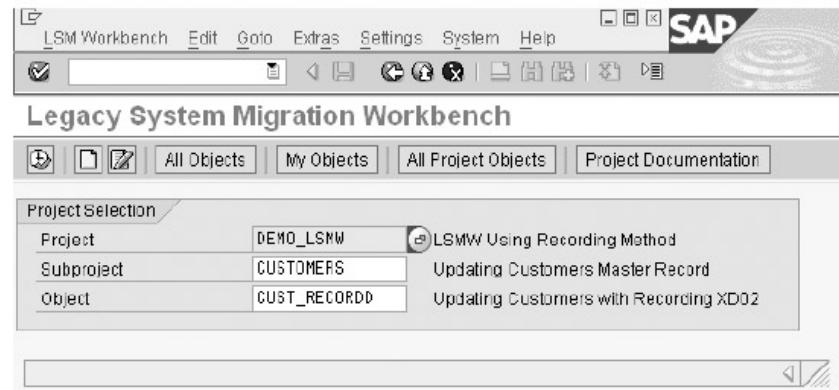


© SAP AG. All rights reserved.

Figure 11.77: Create object screen

6. Enter the name of the object as CUST_RECORD and the description Updating Customers with Recording XD02 in the Object and Name fields, respectively. Click the Continue () icon.

The initial screen of the LSMW appears, displaying the values in all the fields, as shown in [Figure 11.78](#):

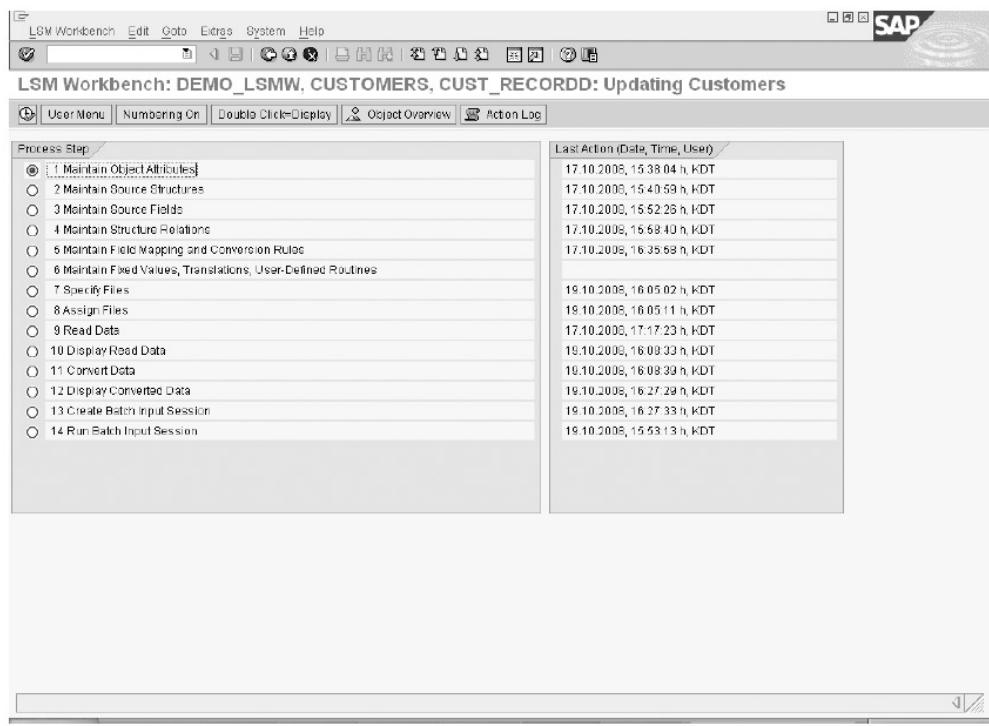


© SAP AG. All rights reserved.

Figure 11.78: The initial screen of the LSMW

7. Click the Execute (icon ([Figure 11.78](#)).

The LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDDD: Updating Customers with Recording XD02 screen appears, as shown in [Figure 11.79](#):



© SAP AG. All rights reserved.

Figure 11.79: The LSMW wizard

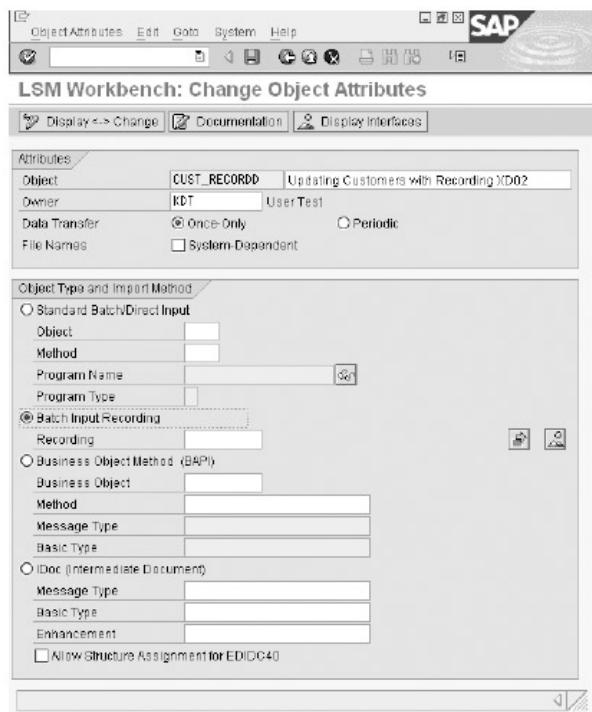
In Figure 11.79, the screen contains a sequence of process steps that you can execute to transfer data. When one process step is executed, the cursor automatically points at the next process step. Note that this screen is also referred to as the LSMW wizard.

Note The process steps depend on the settings of the User menu. These settings can be altered by clicking the User Menu button. The number related to each step can also be displayed by clicking the Numbering Off button or can be hidden by clicking the Numbers On button. Each step can be executed either by double-clicking the row or by clicking the Execute (Execute icon) button. The Double Click=Change (Double Click=Change button) displays the steps in change mode; whereas the Double Click=Display (Double Click=Display button) displays the steps in display mode.

Executing the First Process Step

The name of the first process step of the LSMW wizard is Maintain Object Attributes. In this step, an object attribute, such as object type and import method, is defined. Perform the following steps to execute the Maintain Object Attributes process step:

1. Double-click the Maintain Object Attributes process step. The LSM Workbench: Change Object Attributes screen appears, as shown in Figure 11.80:



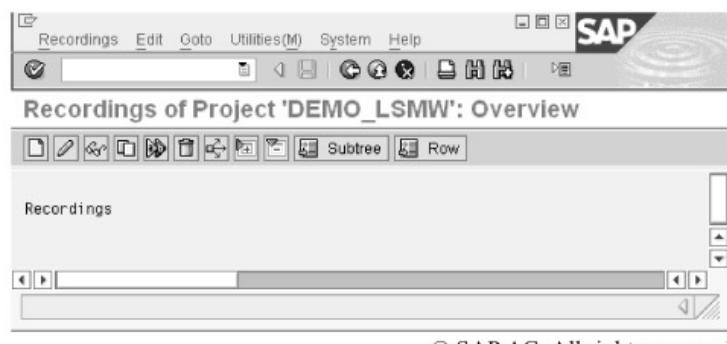
© SAP AG. All rights reserved.

Figure 11.80: Batch input recording

Figure 11.80 shows the import methods in the form of radio buttons: Standard Batch/Direct Input, Batch Input Recording, Business Object Method (BAPI), and IDoc (Intermediate Document). You can select any of these radio buttons according to your requirements. In this case, we have selected the Batch Input Recording radio button.

2. Select the Batch Input Recording radio button and then click the Recording: Overview (录) icon to record the R/3 transaction (see Figure 11.80).

The Recordings of Project 'DEMO_LSMW': Overview screen appears, as shown in Figure 11.81:

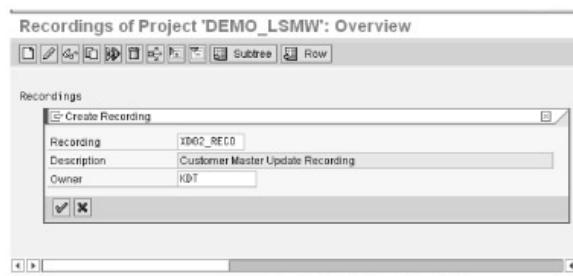


© SAP AG. All rights reserved.

Figure 11.81: Creating recording

3. Click the Create Recording (录) icon to create a recording of a transaction in the SAP system, as shown in Figure 11.81.

The Create Recording dialog box appears, as shown in Figure 11.82:

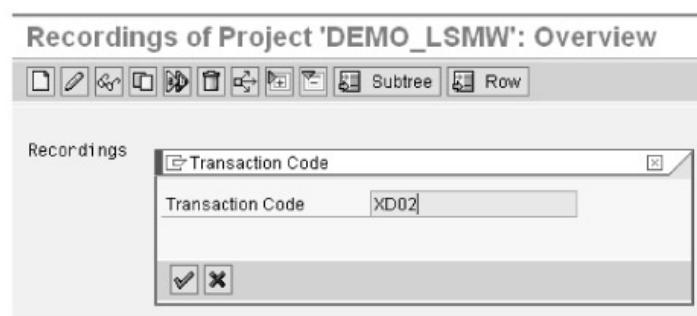


© SAP AG. All rights reserved.

Figure 11.82: Recording of the project

- Enter the XD02_RECO and Customer Master Update Recording in the Recording and Description fields, respectively. Click the Continue (✓) icon.

The Transaction Code dialog box appears, as shown in [Figure 11.83](#):

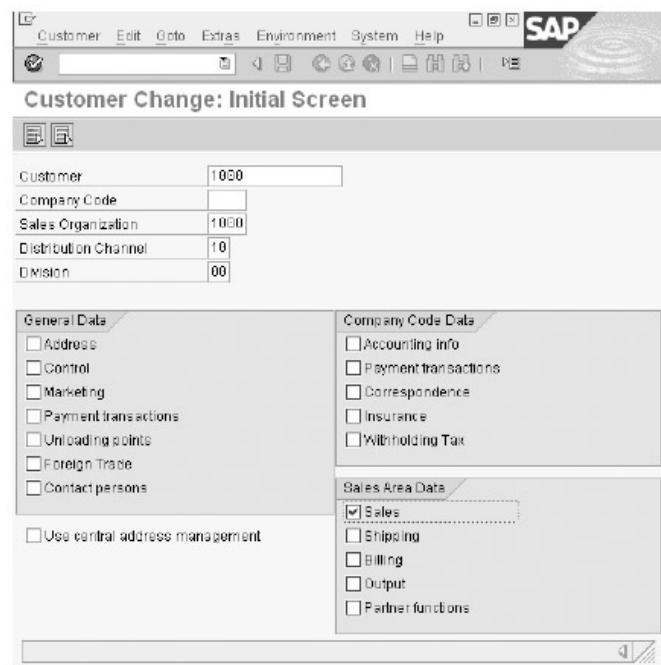


© SAP AG. All rights reserved.

Figure 11.83: Entering transaction code

- Enter the transaction code as XD02 to update the existing customer in the SAP system and then click the Continue (✓) icon, as shown in [Figure 11.83](#).

The initial screen of Customer Change appears, as shown in [Figure 11.84](#):



© SAP AG. All rights reserved.

Figure 11.84: The customer change screen

6. Enter the details of the existing customer—say, Customer number as 1000, Sales Organization as 1000, Distribution Channel as 10, and Division as 00 (see [Figure 11.84](#)). Select the Sales check box and press the ENTER key.

The message Recording running appears in the status bar, and then the Customer Change: Sales Sales Area screen appears, as shown in [Figure 11.85](#):

Customer Change: Sales Sales Area

| | | | |
|----------------|------|----------------------|--------|
| Customer | 1000 | Becker Berlin | Berlin |
| Sales Org. | 1000 | Germany Frankfurt | |
| Distr. Channel | 10 | Final customer sales | |
| Division | 00 | Cross-division | |

Sales order

| | | | |
|---------------------------------------|--------|-----------------|-------|
| Sales district | DE0010 | Order probab. | 100 % |
| Sales Office | 1010 | AuthorizGroup | |
| Sales Group | 110 | Item proposal | |
| Customer group | 01 | Acct at cust. | 3333 |
| ABC class | | UoM Group | |
| Currency | EUR | Exch. Rate Type | |
| <input type="checkbox"/> Rounding off | | PP cust. proc. | |

Pricing/Statistics

| | |
|-----------------|----|
| Price group | 01 |
| Cust.pric.proc. | 1 |
| Price List | 03 |
| Cust Stats Grp | 1 |

Agency business

| |
|--|
| <input type="checkbox"/> Agency business |
|--|

© SAP AG. All rights reserved.

Figure 11.85: Performing transaction recording for transaction XD02

[Figure 11.85](#) displays the information related to the customer sales area.

7. Click the Save (□) icon and then click the Back (○) icon.

The Create Recording screen appears, as shown in [Figure 11.86](#):

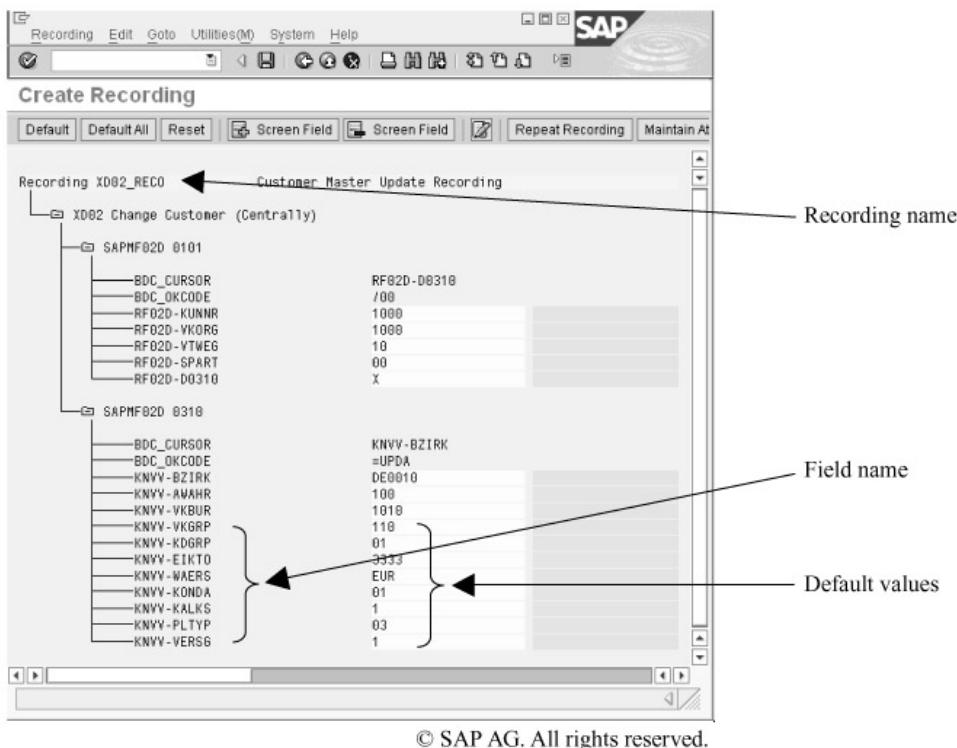


Figure 11.86: The screen showing the details regarding the recording

The Create Recording screen displays the name of the recording created (that is, XD02_RECO) along with the field names and their default values, such as BDC_CURSOR, BDC_OKCODE, RF02D-KUNNR, and RF02D-VKORG. On this screen, you can also delete the fields not needed in the recording by clicking the Remove Screen field ().

As shown in Figure 11.86, the field names are shown in a technical format. You can replace these technical names with the descriptive names by double-clicking the field. In this case, we change the technical names of the required fields.

- Double-click the field whose name you want to change. In our case, we double-click the RF02D-KUNNR field.

The Create Recording dialog box appears, as shown in Figure 11.87:



Figure 11.87: Changing the details regarding the fields

- Enter the name of the field (that is, KUNNR) in the Name field and the description related to this field (that is, Customer Account Number), in the second Name field. Similarly, we change the technical names of all these fields by following the same procedure mentioned for the field RF02D-KUNNR. After all the changes are made, the Create Recording screen appears, as shown in Figure 11.88:

- Click the Save () icon and then click the Back () icon.

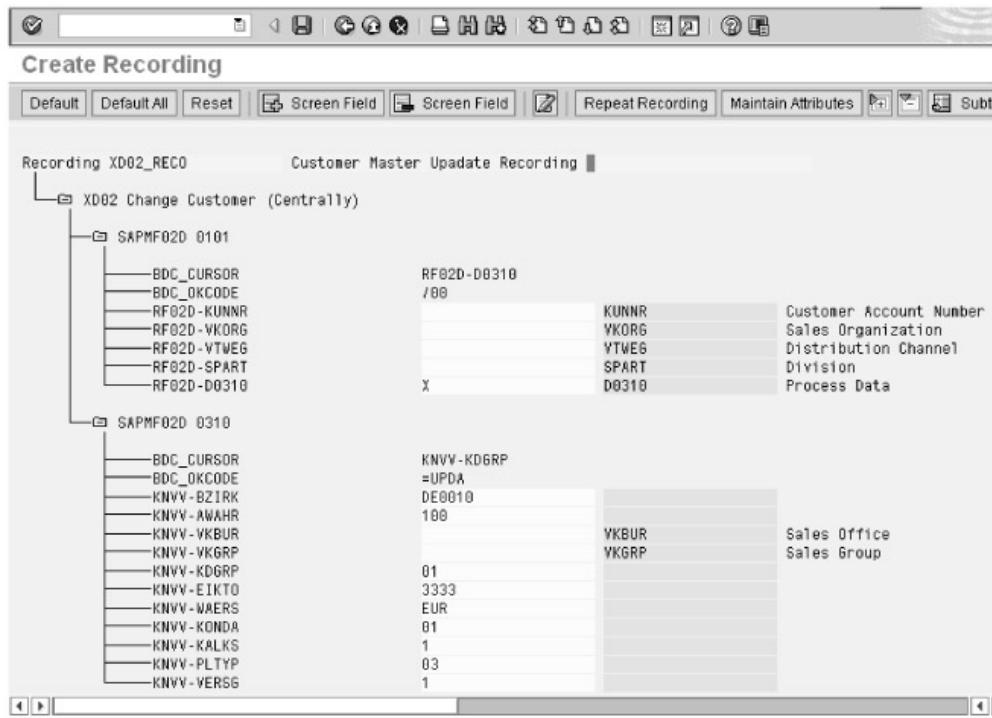


Figure 11.88: The screen showing the changes

The Recordings of Project DEMO_LSMW: Overview screen appears, as shown in Figure 11.89:

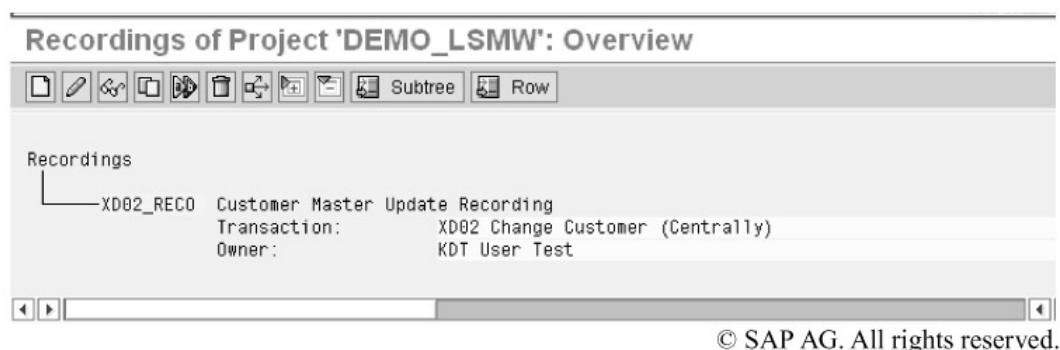
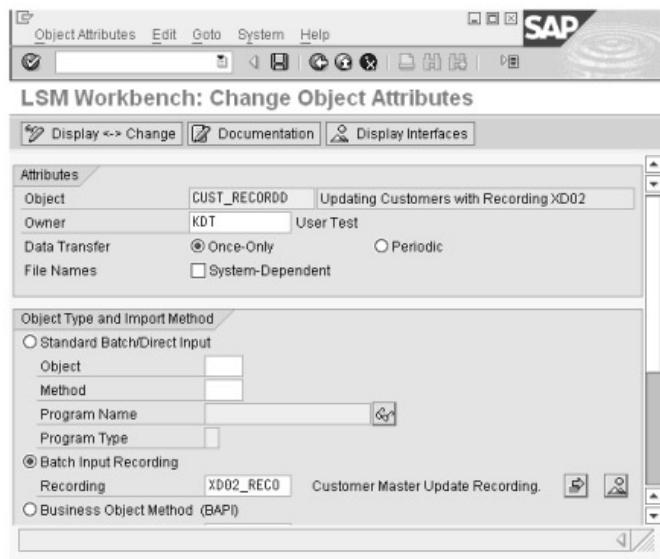


Figure 11.89: Recording overview

Figure 11.89 displays the details of the created recording, XD02_REC.

11. Click the Back (⌚) icon.

The LSM Workbench: Change Object Attributes screen appears, as shown in Figure 11.90:



© SAP AG. All rights reserved.

Figure 11.90: The change object attributes screen

12. Enter the name of the created recording as XD02_RECO in the Recording field, as shown in Figure 11.90.

13. Click the Save (□) icon and then click the Back (○) icon.

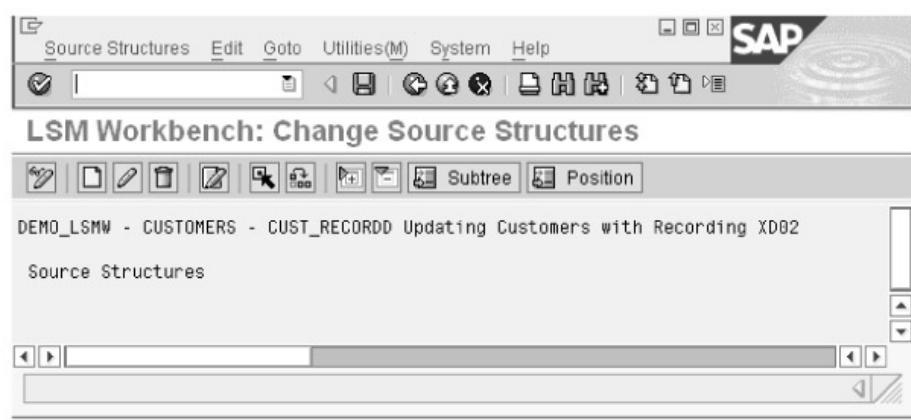
The cursor is shifted automatically to the next step in the LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDD: Updating Customers with Recording XD02 screen.

Executing the Second Process Step

The second process step of the LSMW wizard is Maintain Source Structures. In this process step, a source structure is defined. Execute this step by double-clicking it.

Note Now, you see only 14 steps on the screen, because we want to import the data with the help of the recording method, which means that the steps related to the Standard Batch/Direct Input, Direct input, and IDoc methods have been hidden.

The LSM Workbench: Change Source Structures screen appears, as shown in Figure 11.91:



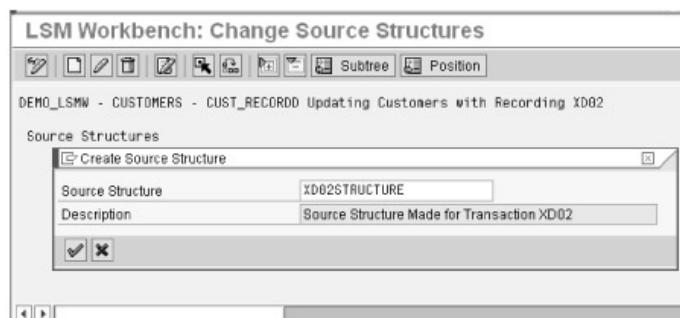
© SAP AG. All rights reserved.

Figure 11.91: The LSM workbench—change source structures screen

In the LSM Workbench: Change Source Structures screen, we create a source structure for the method, as shown in the following steps:

1. Click the Create a Structure (□) icon (see Figure 11.91).

The Create Source Structure dialog box appears, as shown in [Figure 11.92](#):

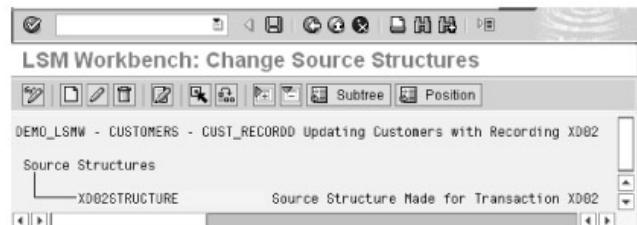


© SAP AG. All rights reserved.

Figure 11.92: The create source structure screen

2. Enter the name of the source structure in the Source Structure field and a related description in the Description field. In this case, we enter XD02STRUCTURE in the Source Structure field and Source Structure Made for Transaction XD02 in the Description field, as shown in [Figure 11.92](#).
3. Click the Continue (✓) icon.

The Change Source Structures screen appears, as shown in [Figure 11.93](#):



© SAP AG. All rights reserved.

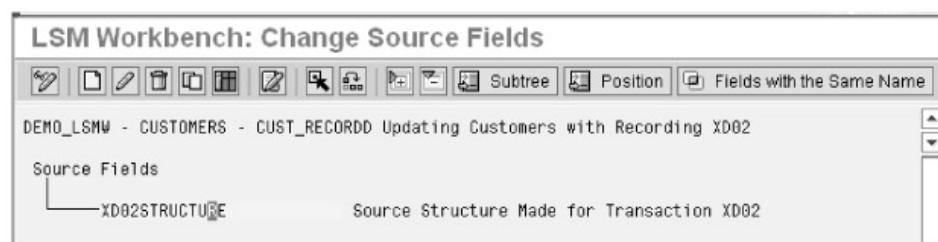
Figure 11.93: Description of the source structure

4. Click the Save (■) icon and then click the Back (⌚) icon.

Executing the Third Process Step

The third process step of the LSMW wizard is Maintain Source Fields. In this process step, source fields related to the source structure are defined. Execute this process step by double-clicking it.

The LSM Workbench: Change Source Fields screen appears, as shown in [Figure 11.94](#):



© SAP AG. All rights reserved.

Figure 11.94: The change source fields screen

In the LSM Workbench: Change Source Fields screen, you need to list all the fields you want to include in the source structure To do so, perform the following steps:

- Select the created source structure, XD02STRUCTURE, and then click the Table Maintenance () icon, as shown in [Figure 11.94](#).

The Source Fields for Source Structure XD02STRUCTURE screen appears, as shown in [Figure 11.95](#):

| Field Name | Type | Length | Field description |
|-------------|------|--------|-------------------|
| CUSTOMER | C | 10 | CUSTOMER |
| SALESORG | C | 4 | SALESORG |
| DISTCHANNEL | C | 2 | DISTCHANNEL |
| DIVISION | C | 2 | DIVISION |
| SALESOFFICE | C | 4 | SALESOFFICE |
| SALESGROUP | C | 3 | SALESGROUP |

© SAP AG. All rights reserved.

Figure 11.95: Maintaining source fields for the source structure

- Enter the values for different fields present on the screen, as shown in [Figure 11.95](#).

In addition to this, you also need to update two fields, SALESOFFICE and SALESGROUP.

- Click the Save () icon and then click the Back () icon.

The LSM Workbench: Change Source Fields screen appears, as shown in [Figure 11.96](#):

| Source Fields | Source Structure Made For Transaction XD02 |
|---------------|--|
| XD02STRUCTURE | CUSTOMER SALESORG DISTCHANNEL DIVISION SALESOFFICE SALESGROUP |

© SAP AG. All rights reserved.

Figure 11.96: Description of source fields of the source structure

The names of the fields, along with their respective descriptions, appear on the screen.

- Click the Save () icon and then click the Back () icon.

Executing the Fourth Process Step

The fourth process step of the LSMW wizard is Maintain Structure Relations. In this step, you establish the relationship between the source structure and target structure. Execute this process step by double-clicking it.

The LSM Workbench: Change Structure Relationships screen appears, as shown in Figure 11.97:

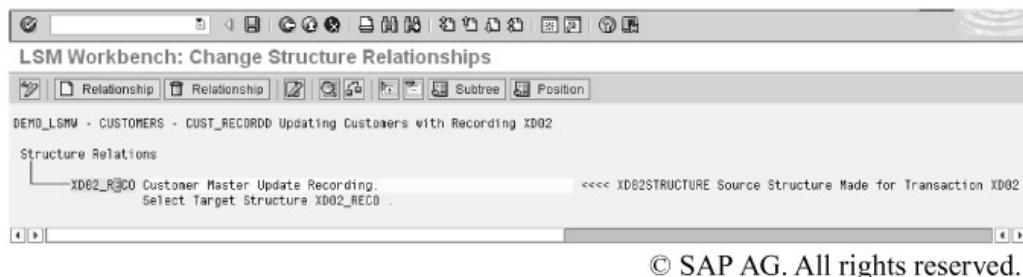


Figure 11.97: Maintaining the relationship for the structure

In this case, only one source structure and target structure is present; therefore, the relationship between them already exists by default. In case there are more source structures and target structures, select the target structure name and then click the Relationship button.

Then, click the Save (□) icon and then click the Back (⊕) icon.

Executing the Fifth Process Step

The fifth process step of the LSMW wizard is Maintain Field Mappings and Conversion Rules. In this step, you first assign the source structure fields to the target structure fields and then define how the field content is to be converted. For each field of the target structure, you define the field description and assign the source structure fields, the type of rule, and the code. Execute this process step by double-clicking it.

The LSM Workbench: Change Field Mappings and Conversion Rules screen appears, as shown in Figure 11.98:

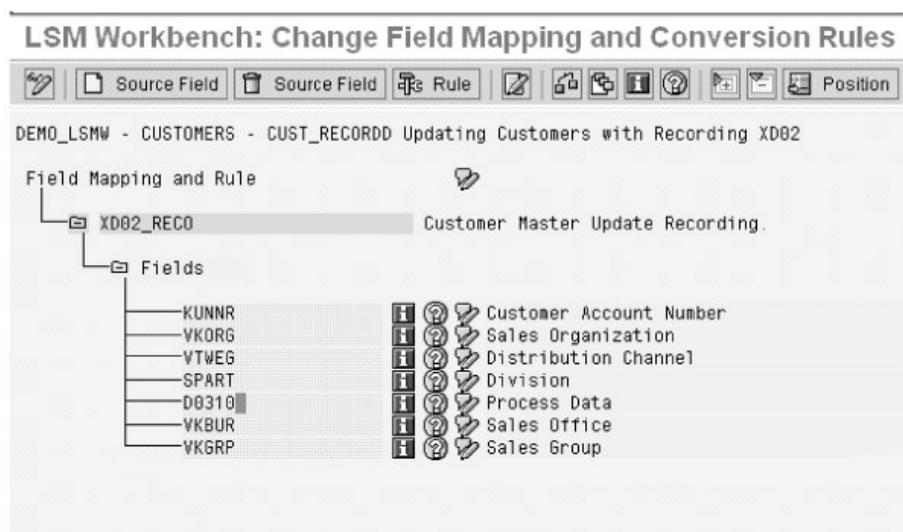


Figure 11.98: The screen for entering the value of the XD02_REC_D0310 field

On this screen, the D0310 field represents that you have selected Sales View for the customer master screen and its value should be set to X so that data conversion is not affected.

Now, perform the following steps to execute the fifth process step:

1. Click the D0310 field and then click the Constant (button) present on the application toolbar.

The Value For XD02_RECO-D0310 dialog box appears, as shown in [Figure 11.99](#):

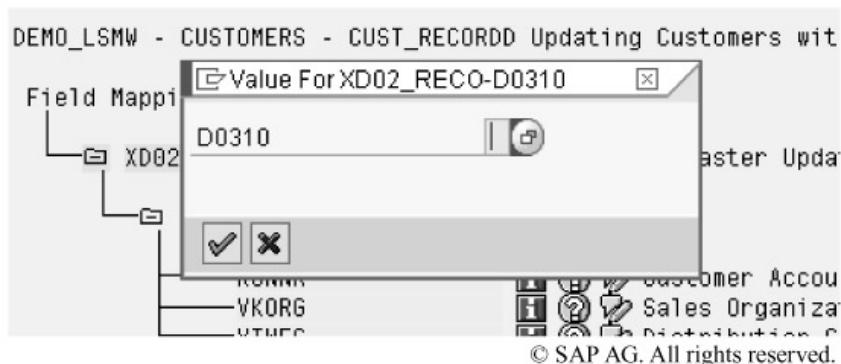


Figure 11.99: The screen for entering the value

2. Click the Search Help (icon beside the D0310 field (see [Figure 11.99](#)).

The Process data? dialog box appears, as shown in [Figure 11.100](#):

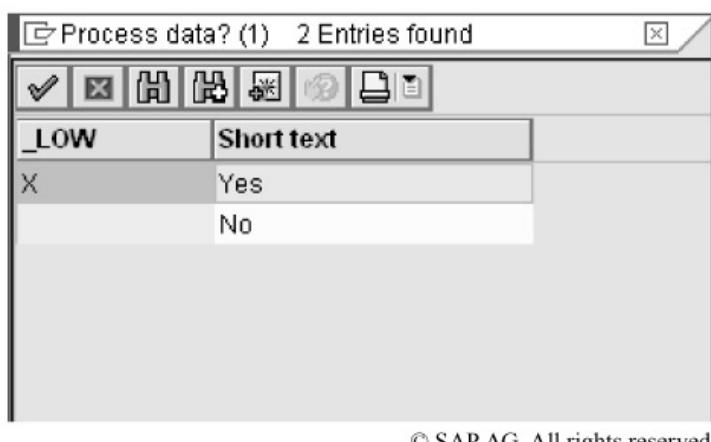
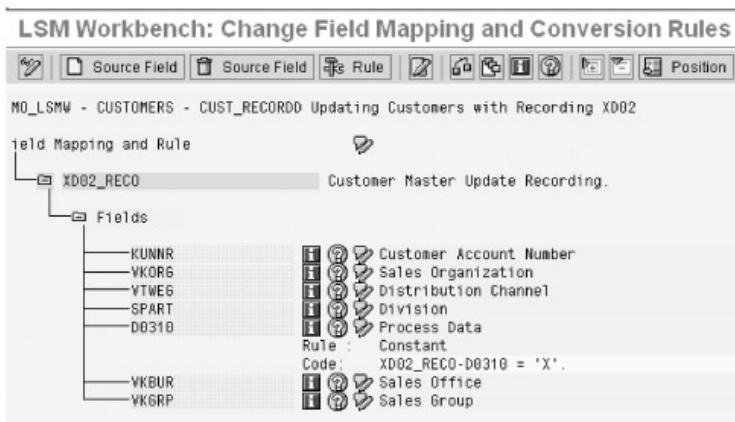


Figure 11.100: The process data dialog box

3. Select the Yes option and then click the Copy (icon.

The corresponding value for the field, X, appears in the D0310 field, as shown in [Figure 11.101](#):



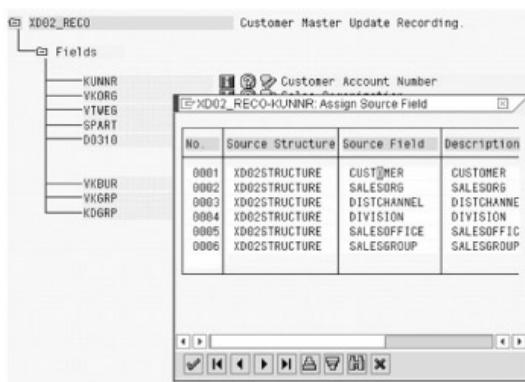
© SAP AG. All rights reserved.

Figure 11.101: Mapping of the field

Figure 11.101 shows that the rule and the corresponding code are assigned to the D0310 field.

- Click the KUNNR field and then click the Source Field ()

The XD02_RECO-KUNNR: Assign Source Field dialog box appears, as shown in Figure 11.102:



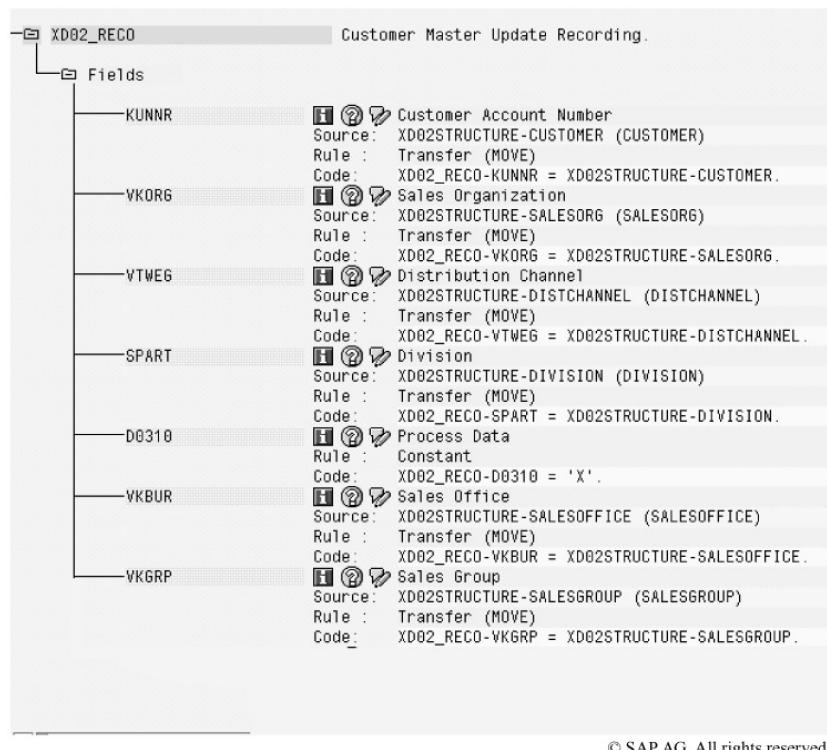
© SAP AG. All rights reserved.

Figure 11.102: The screen for assigning the source field

- Select the source field as CUSTOMER and then click the Choose ()

Follow the same procedure for the rest of the fields, such as VKORG, VTWEG, SPART, VKBUR, and VKGRP, to define the code to transfer data from the fields of the source data structure to the respective fields in the target structure.

Finally, the LSM Workbench: Change Field Mapping and Conversion Rules screen appears, as shown in Figure 11.103:



© SAP AG. All rights reserved.

Figure 11.103: Mapping the fields and defining source structures

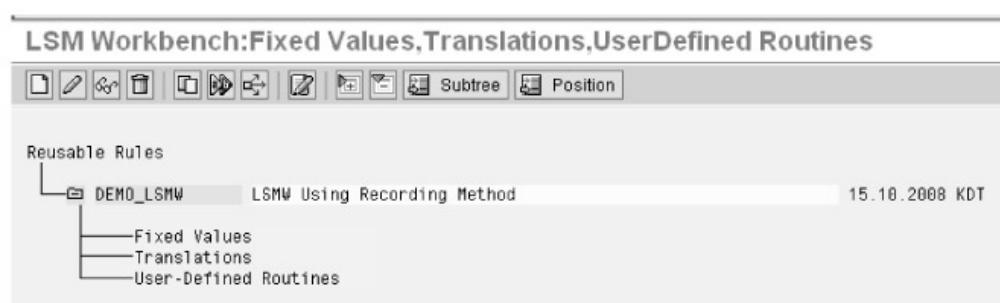
In **Figure 11.103**, note that the relations are established between the fields of the source and target structures.

6. Click the Save () icon and then click the Back () icon.

Executing the Sixth Process Step

The sixth process step of the LSMW wizard is Maintain Fixed Values, Translations, User-Defined Routines. This step allows you to process the reusable rules, such as fixed values, translations, and user-defined routines related to the project. This process step also allows you to define fixed values; that is, a fixed value object is assigned to the field of the target structure. In addition, you can also define reusable translations; that is, the code to carry out field content conversion must be assigned to the target field. You can also use a user-defined routine in different objects related to a project. Execute this process step by double-clicking it.

The LSM Workbench: Fixed Values, Translations, UserDefined Routines screen appears, as shown in **Figure 11.104**:



© SAP AG. All rights reserved.

Figure 11.104: Maintaining fixed values, translations, and user-defined routines

In this case, we do not process any reusable rules; therefore, the sixth step has been skipped. Click the Back () icon.

Executing the Seventh Process Step

The seventh process step of the LSMW wizard is Specify Files. In this step, the location of the file containing legacy data is

specified. Execute this step by double-clicking it.

The LSM Workbench: Specify Files (Change) screen appears, as shown in Figure 11.105:

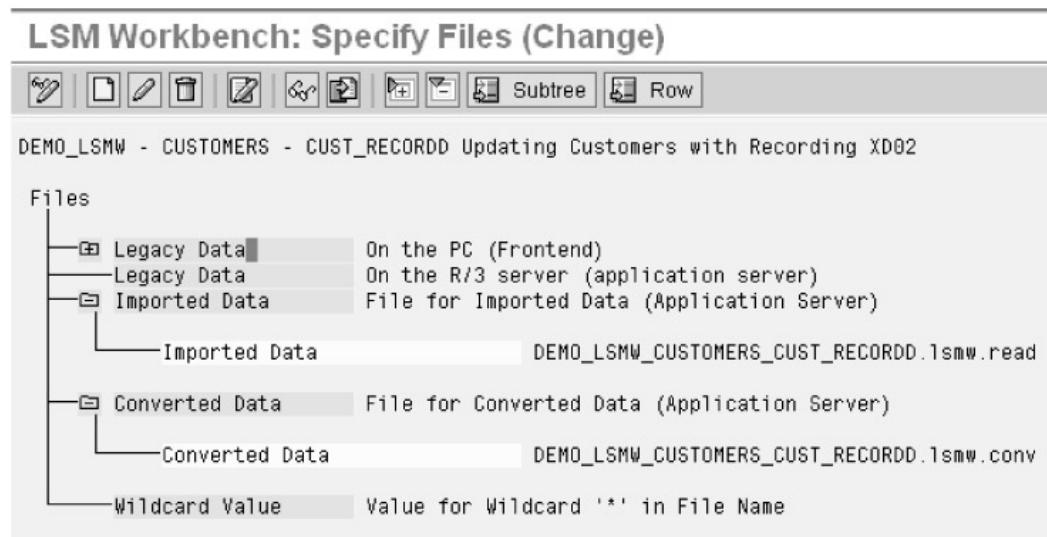


Figure 11.105: Specifying the input file

In the LSM Workbench: Specify Files (Change) screen, the layout of the input file (a notepad file) is specified. This file contains the field names of the XD02 transaction in the first line. Save this input file in the [Tab] delimited file category. Note that this file will be saved in the local drive on your computer. In this case, the file is saved at the location C:\XD02.txt.

Perform the following steps to execute the seventh process step:

1. Double-click Legacy Data on the PC (Frontend), as shown in Figure 11.105.

The File on Front End: Maintain Properties dialog box appears, as shown in Figure 11.106:

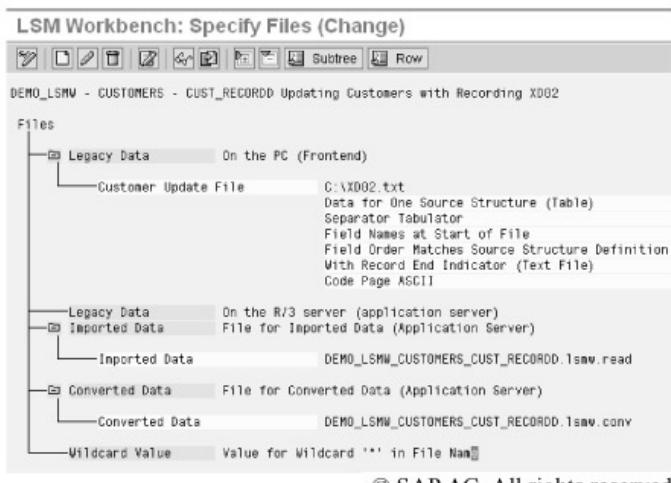


© SAP AG. All rights reserved.

Figure 11.106: Specifying the input file

2. Select the delimiter as Tabulator and the file structure as Field Names at Start of File. Click the Continue (checkbox) icon.

The LSM Workbench: Specify Files (Change) screen appears, as shown in Figure 11.107:



© SAP AG. All rights reserved.

Figure 11.107: Details of the input file

All the selected attributes of the file are displayed on the screen.

3. Click the Save (S) icon and then click the Back (B) icon to return to the LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDD: Updating Customers with Recording XD02 screen.
4. Create an Excel file containing data related to customers, as shown in Figure 11.108:

| A | B | C | D | E | F |
|----------|----------|----------|----------|----------|----------|
| Customer | SalesOrg | DistChnl | Division | SalesOff | SalesGrp |
| 1000 | 1000 | 10 | 00 | 1010 | 110 |
| 1005 | 1000 | 10 | 00 | 1010 | 110 |
| 1007 | 1000 | 10 | 00 | 1010 | 110 |
| 1030 | 1000 | 10 | 00 | 1010 | 110 |
| 1031 | 1000 | 10 | 00 | 1010 | 110 |
| 1171 | 1000 | 10 | 00 | 1010 | 110 |
| 1172 | 1000 | 10 | 00 | 1010 | 110 |
| 1200 | 1000 | 10 | 00 | 1010 | 110 |
| 1235 | 1000 | 10 | 00 | 1010 | 110 |
| 1280 | 1000 | 10 | 00 | 1010 | 110 |

© SAP AG. All rights reserved.

Figure 11.108: An input excel file

Figure 11.108 shows the records of the customer that need to be updated. Now, save this Excel file as a Tab-delimited text file named XD02 on the local drive of your computer (in this case, it is C:\).

Executing the Eighth Process Step

The next process step of the LSMW wizard is Assign Files. In this step, the file name is assigned to the source structure. Execute this step by double-clicking it.

The LSM Workbench: Assign Files (Change) screen appears, as shown in Figure 11.109:

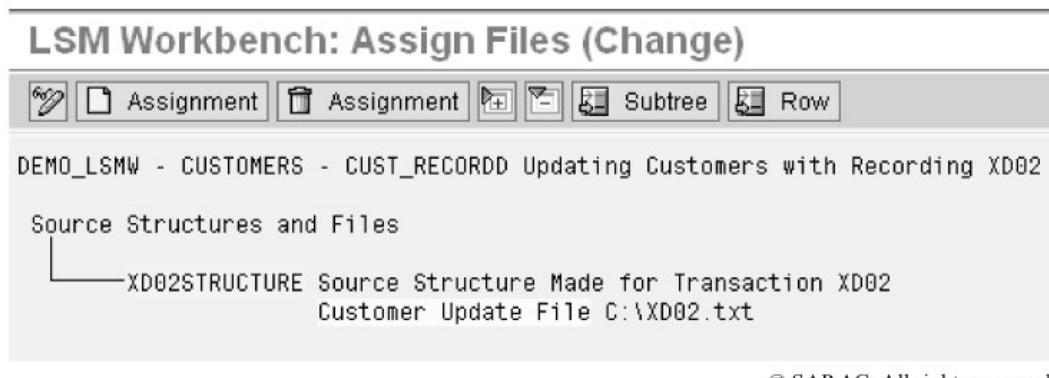


Figure 11.109: Assigning input files

The SAP system automatically assigns the file name to the source structure. In this case, the file name is XD02.txt.

Click the Save () icon and then the Back () icon.

Executing the Ninth Process Step

The next process step of the LSMW wizard is Read Data. In this step, the data is read from the source file. Execute this step by double-clicking it.

The LSM Workbench: Import Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD screen appears, as shown in [Figure 11.110](#):

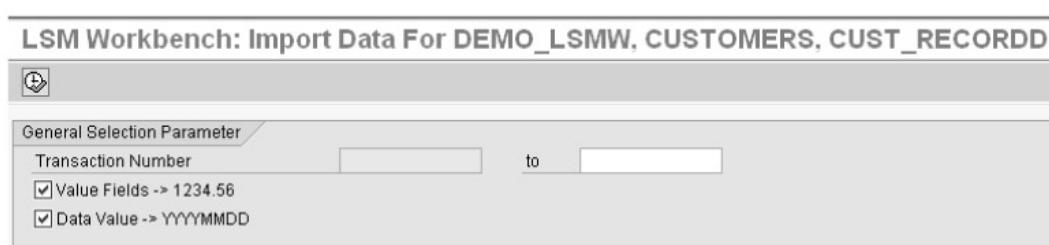


Figure 11.110: Importing data

In the LSM Workbench: Import Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD, screen LSMW reads the data from the source input file; that is, XD02.

Now, perform the following steps to execute the ninth process step:

1. Click the Execute () icon.

The LSM Workbench: Import Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD screen appears, as shown in [Figure 11.111](#):

| LSM Workbench: Import Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD | | | |
|---|------|---------|-------------|
| LSM Workbench: Import Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD | | | |
| 16.10.2008 - 19:23:30 | | | |
| File(s) Read: C:\XD02.txt File Written: DEMO_LSMW_CUSTOMERS_CUST_RECORDD.lsmw.read | | | |
| Source Structure | Read | Written | Not Written |
| XD02STRUCTURE | 10 | 10 | 0 |
| Transactions Read: | 10 | | |
| Records Read: | 10 | | |
| Transactions Written: | 10 | | |
| Records Written: | 10 | | |

© SAP AG. All rights reserved.

Figure 11.111: Displaying the details of the read data

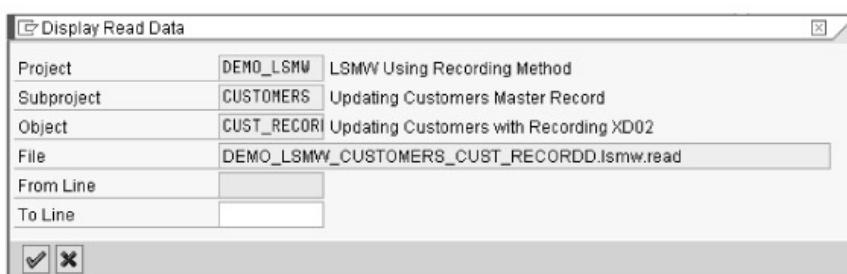
In [Figure 11.111](#), note that the number of records maintained in the input file (XD02.txt) is read and written in the given transaction.

- Click the Back (⌚) icon twice. The LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDD: Updating Customers with Recording XD02 screen appears.

Executing the Tenth Process Step

The tenth process step of the LSMW wizard is Display Read Data. This step is used to read the converted data. Execute this step by double-clicking it.

The Display Read Data dialog box appears, as shown in [Figure 11.112](#):



© SAP AG. All rights reserved.

Figure 11.112: Displaying the read data

This screen allows you to provide the number of lines in the From Line and To Line fields that you want to display. If you do not enter any number in these fields, all the lines of the notepad file (XD02. txt) will be displayed. In this case, we want to display all the lines; therefore, we leave these fields blank.

Perform the following steps to execute the tenth process step:

- Click the Continue (checkbox checked) icon.

The LSM Workbench: Imported Data screen appears, as shown in [Figure 11.113](#):

LSM Workbench: Imported Data

| File DEMO_LSMW_CUSTOMERS_CUST_RECORDD.1smw.read | | | | | |
|---|---------------|-----------|-----------------|-----|----------------------|
| Row | Struct. | Conts. | | | |
| | LSMWDEMO_LSMW | CUSTOMERS | CUST_RECORDD | DMT | 80020081017171638KDT |
| 1 | XD02STRUCTURE | 1000 | 100010001010110 | | |
| 2 | XD02STRUCTURE | 1005 | 100010001010110 | | |
| 3 | XD02STRUCTURE | 1007 | 100010001010110 | | |
| 4 | XD02STRUCTURE | 1030 | 100010001010110 | | |
| 5 | XD02STRUCTURE | 1031 | 100010001010110 | | |
| 6 | XD02STRUCTURE | 1171 | 100010001010110 | | |
| 7 | XD02STRUCTURE | 1172 | 100010001010110 | | |
| 8 | XD02STRUCTURE | 1200 | 100010001010110 | | |
| 9 | XD02STRUCTURE | 1235 | 100010001010110 | | |
| 10 | XD02STRUCTURE | 1280 | 100010001010110 | | |

© SAP AG. All rights reserved.

Figure 11.113: Imported data

In the LSM Workbench: Imported Data screen, the details of the imported data are displayed. In addition to this, you can view the detailed description of any record by double-clicking it. **Figure 11.114** shows the details of the imported data:

LSM Workbench: Imported Data

| File DEMO_LSMW_CUSTOMERS_CUST_RECORDD.1smw.read | | |
|---|-------------|-------------|
| Structure XD02STRUCTURE | | |
| Field Name | Field Text | Field Value |
| CUSTOMER | CUSTOMER | 1000 |
| SALESORG | SALESORG | 1000 |
| DISTCHANNEL | DISTCHANNEL | 10 |
| DIVISION | DIVISION | 00 |
| SALESOFFICE | SALESOFFICE | 1010 |
| SALESGROUP | SALESGROUP | 110 |

© SAP AG. All rights reserved.

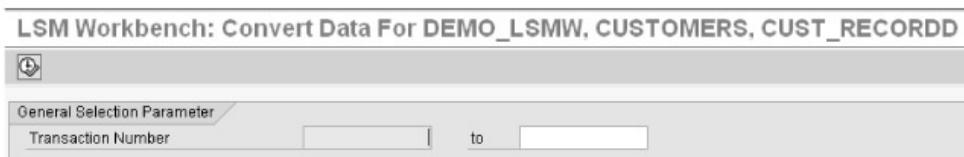
Figure 11.114: Detailed description of the data

- Click the Back (Back) icon twice. The LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDD: Updating Customers with Recording XD02 screen appears.

Executing the Eleventh Process Step

The eleventh process step of the LSMW wizard is Convert Data. This step is used to convert the data from the source format into the target format. Execute this step by double-clicking it.

The LSM Workbench: Convert Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD screen appears, as shown in **Figure 11.115**:



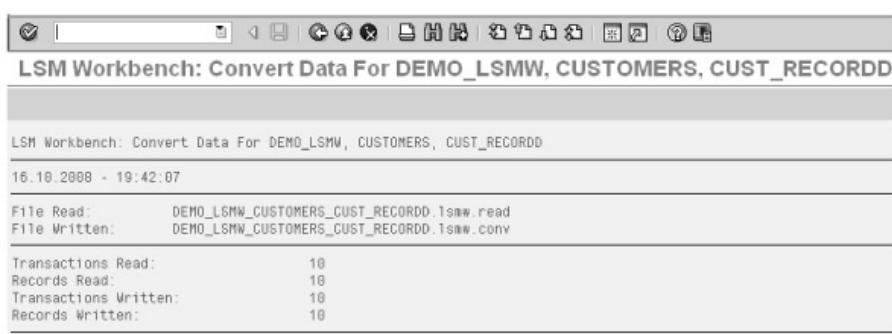
© SAP AG. All rights reserved.

Figure 11.115: Converting data

Perform the following steps to execute the eleventh process step:

1. Click the Execute (Execute icon) icon.

The LSM Workbench: Convert Data For DEMO_LSMW, CUSTOMERS, CUST_RECORDD screen appears, as shown in [Figure 11.116](#):



© SAP AG. All rights reserved.

Figure 11.116: The screen showing the converted data

In [Figure 11.116](#), information, such as Transactions Read, Records Read, Transactions Written, and Records Written, is displayed.

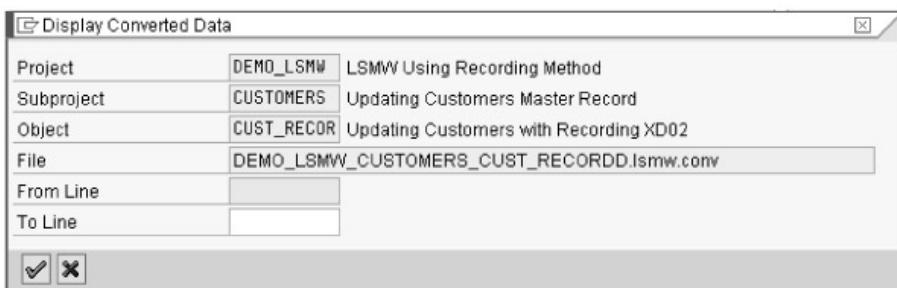
2. Click the Back (Back icon) icon twice. The LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDD: Updating Customers with recording XD02 screen appears.

Now, let's learn about the next process step.

Executing the Twelfth Process Step

The twelfth process step of the LSMW wizard is Display Converted Data. This step is used to display the converted source data. Execute this process step by double-clicking it.

The Display Converted Data dialog box appears, as shown in [Figure 11.117](#):



© SAP AG. All rights reserved.

Figure 11.117: Converted data

Perform the following steps to execute the twelfth process step:

1. Click the Continue (Continue icon) icon.

The LSM Workbench: Converted Data screen appears, as shown in [Figure 11.118](#):

| LSM Workbench: Converted Data | | | | | |
|---|----------------|----------------|-----------------------|------|------------------|
| | Field Contents | Change Display | Display Colour Legend | | |
| File DEMO_LSMW_CUSTOMERS_CUST_RECORDD.lsmw.conv | | | | | |
| Row | Struct. | Contents | | | |
| 1 | XD02_REC0 | XD02_REC0 | XD02 | 1003 | 10001000X1010110 |
| 2 | XD02_REC0 | XD02_REC0 | XD02 | 1005 | 10001000X1010110 |
| 3 | XD02_REC0 | XD02_REC0 | XD02 | 1007 | 10001000X1010110 |
| 4 | XD02_REC0 | XD02_REC0 | XD02 | 1033 | 10001000X1010110 |
| 5 | XD02_REC0 | XD02_REC0 | XD02 | 1031 | 10001000X1010110 |
| 6 | XD02_REC0 | XD02_REC0 | XD02 | 1171 | 10001000X1010110 |
| 7 | XD02_REC0 | XD02_REC0 | XD02 | 1172 | 10001000X1010110 |
| 8 | XD02_REC0 | XD02_REC0 | XD02 | 1209 | 10001000X1010110 |
| 9 | XD02_REC0 | XD02_REC0 | XD02 | 1235 | 10001000X1010110 |
| 10 | XD02_REC0 | XD02_REC0 | XD02 | 1280 | 10001000X1010110 |

© SAP AG. All rights reserved.

[Figure 11.118](#): Displaying data after conversion

In the LSM Workbench: Converted Data screen, you can view the detailed description of any record by double-clicking it. [Figure 11.119](#) shows the details of the converted data in the tabular format:

| LSM Workbench: Converted Data | | |
|---|-------------------------|-----------|
| File DEMO_LSMW_CUSTOMERS_CUST_RECORDD.lsmw.conv | | |
| Structure XD02_REC0 | | |
| Fld Name | Fld Text | FldValue |
| TABNAME | Table Name | XD02_REC0 |
| TCODE | Transaction Code | XD02 |
| KUNNR | Customer Account Number | 1000 |
| VKORG | Sales Organization | 1000 |
| VTWEG | Distribution Channel | 10 |
| SPART | Division | 00 |
| D0310 | Process Data | X |
| VKBUR | Sales Office | 1010 |
| VKGRO | Sales Group | 110 |

© SAP AG. All rights reserved.

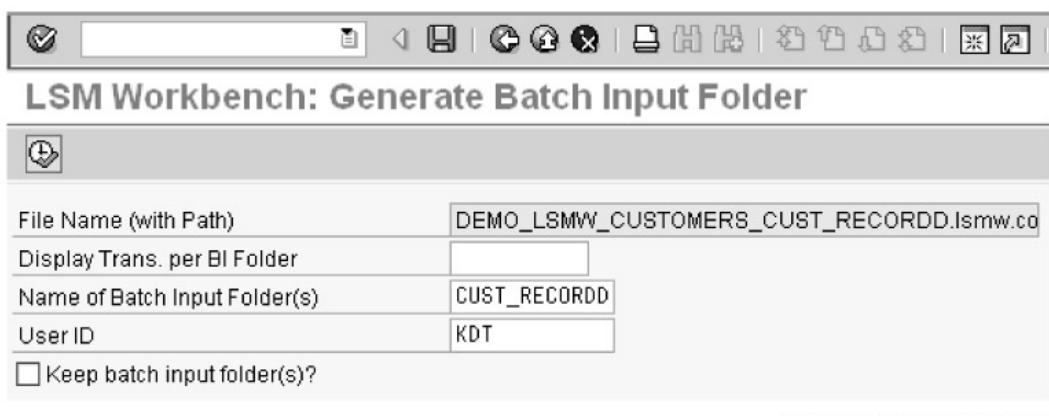
[Figure 11.119](#): Data of customer 1000

- Click the Back (Back) icon to navigate back to the LSM Workbench: DEMO_LSMW, CUSTOMERS, CUST_RECORDD: Updating Customers with Recording XD02 screen.

Executing the Thirteenth Process Step

The thirteenth process step of the LSMW wizard is Create Batch Input Session. This step is used to create a batch input session. Execute this step by double-clicking it.

The LSM Workbench: Generate Batch Input Folder screen appears, as shown in [Figure 11.120](#):



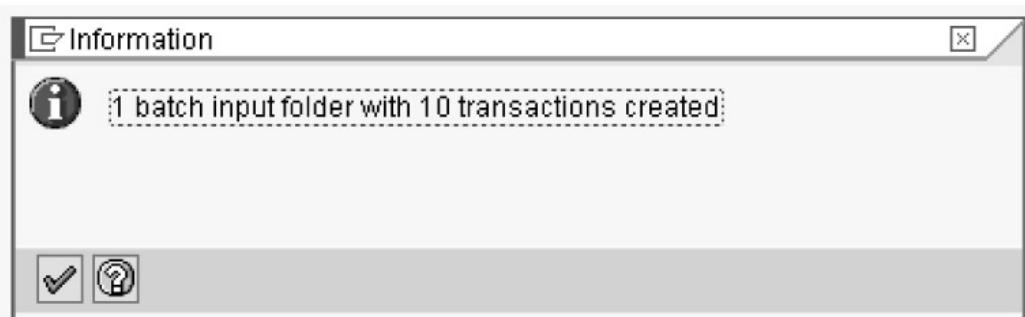
© SAP AG. All rights reserved.

Figure 11.120: Creating the batch input folder

Now, perform the following steps to execute the thirteenth process step:

1. Click the Execute (icon).

An Information dialog box appears, as shown in [Figure 11.121](#):



© SAP AG. All rights reserved.

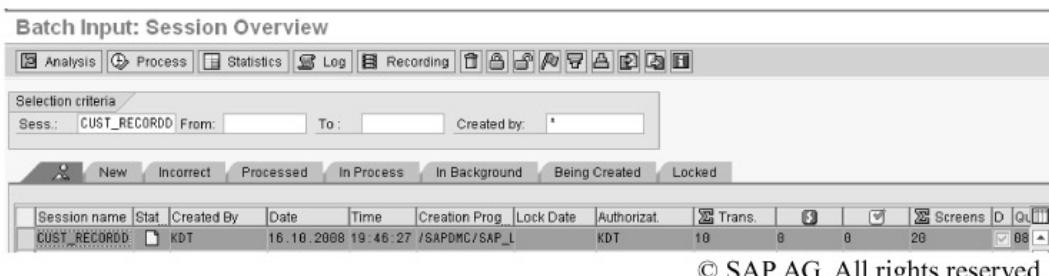
Figure 11.121: The information dialog box

2. Click the Continue (icon).

Executing the Fourteenth Process Step

The fourteenth process step of the LSMW wizard is Run Batch Input Session. This step is used to process the created batch input session. Execute this step by double-clicking it.

The Batch Input: Session Overview screen appears, as shown in [Figure 11.122](#):



© SAP AG. All rights reserved.

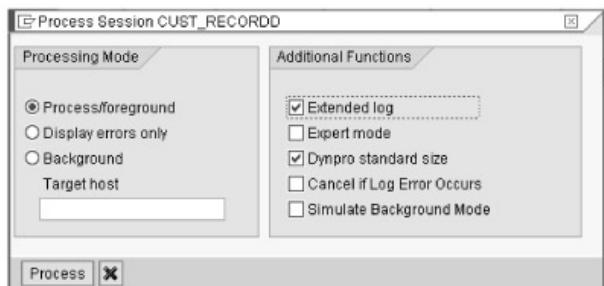
Figure 11.122: Session overview

[Figure 11.122](#) displays information related to the CUST_RECORDD session, such as Status, Created By, Date, Time, Creation Program, Lock Date, Authorization, Total Number of Transactions, Transactions with errors, Transactions

processed, Screens, Deletion Idx., and Queue ID.

1. Select the session name and click the Process button.

The Process Session CUST_RECORDD dialog box appears, as shown in Figure 11.123:



© SAP AG. All rights reserved.

Figure 11.123: Processing screen

This screen allows you to select any additional functions required to process the CUST_RECORDD session.

2. Select the Process/foreground radio button, and the Extended log and Dynpro standard size check boxes. Now, click the Process button.

The initial screen of Customer Change appears, as shown in Figure 11.124:

© SAP AG. All rights reserved.

Figure 11.124: Changing customer

The OK-Code for the initial screen of Customer Change is also displayed in Figure 11.124.

3. Click the Continue () icon.

The Customer Change: Sales Sales Area screen appears, as shown in Figure 11.125:

Customer Change: Sales Sales Area

| | | | |
|----------------|------|-------------------|--------------------------------|
| Customer | 1005 | Becker Berlin | Berlin |
| Sales Org. | 1000 | Germany Frankfurt | |
| Distr. Channel | 10 | Final ch. | Customer Change: Sales Sale... |
| Division | 00 | Cross- | OK-Code =UPDA |

Sales order

| | | | |
|---------------------------------------|--------|-----------------|-----|
| Sales district | DE0010 | Order probab. | 100 |
| Sales Office | 1010 | AuthorizGroup | |
| Sales Group | 110 | Item proposal | |
| Customer group | 02 | Acct at cust. | 33: |
| ABC class | | UoM Group | |
| Currency | EUR | Exch. Rate Type | |
| <input type="checkbox"/> Rounding off | | PP cust. proc. | |

Pricing/Statistics

| | |
|-----------------|----|
| Price group | 01 |
| Cust.pric.proc. | 1 |
| Price List | 03 |
| Cust.Stat.Grp | 1 |

© SAP AG. All rights reserved.

Figure 11.125: Changing customer group

4. Update the Customer group field from 01 to 02, as shown in Figure 11.125.

The OK-Code for the screen is displayed as =UPDA.

5. Click the Continue (checkbox) icon of the Customer Change: Sales Sales Area dialog box.

Customer details corresponding to the customer number 1005 are displayed, as shown in Figure 11.126:

The OK-Code for this screen is displayed as /00.

Customer Change: Initial Screen

| | |
|----------------------|------|
| Customer | 1005 |
| Company Code | |
| Sales Organization | 1000 |
| Distribution Channel | 10 |
| Division | 00 |

General Data

| | |
|---|---|
| <input type="checkbox"/> Address | <input type="checkbox"/> Accounting info |
| <input type="checkbox"/> Control | <input type="checkbox"/> Payment transactions |
| <input type="checkbox"/> Marketing | <input type="checkbox"/> Correspondence |
| <input type="checkbox"/> Payment transactions | <input type="checkbox"/> Insurance |
| <input type="checkbox"/> Unloading points | <input type="checkbox"/> Withholding Tax |
| <input type="checkbox"/> Foreign Trade | |
| <input type="checkbox"/> Contact persons | |

Company Code Data

| |
|--------------------------------|
| <input type="checkbox"/> Sales |
|--------------------------------|

© SAP AG. All rights reserved.

Figure 11.126: Details of customer

6. Click the Continue (checkbox) icon of the Customer Change: Initial Screen dialog box (Figure 11.126).

The Customer Change: Sales Sales Area screen for customer number 1005 appears, as shown in Figure 11.127:

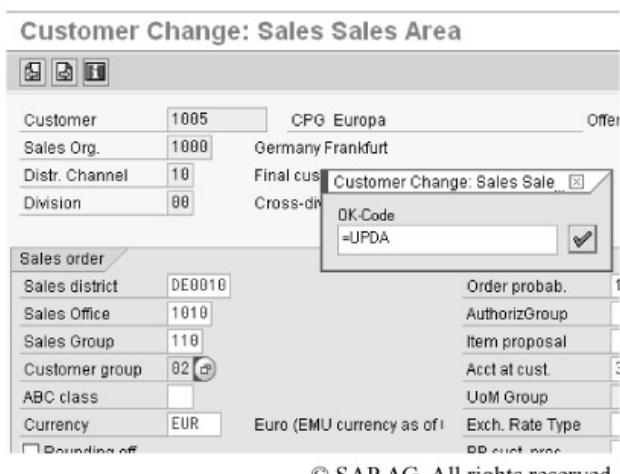


Figure 11.127: The sales data screen

In Figure 11.127, note that the OK-Code for the screen is also shown as = UPDA.

7. In Customer Change: Sales Sales Area screen, change the value in the Customer group field from 01 to 02 (see Figure 11.127).
8. Click the Continue (checkbox) icon (Figure 11.127) of the Customer Change: Sales Sales Area dialog box.

The initial screen of Customer Change for the next customer (that is, 1007), appears.

9. Repeat Steps 2 to 5 to update the Customer group field for the remaining customers as mentioned in the Excel file (see Figure 11.108).

After making the changes for the remaining customers in the Customer group field, an information dialog box appears, as shown in Figure 11.128:

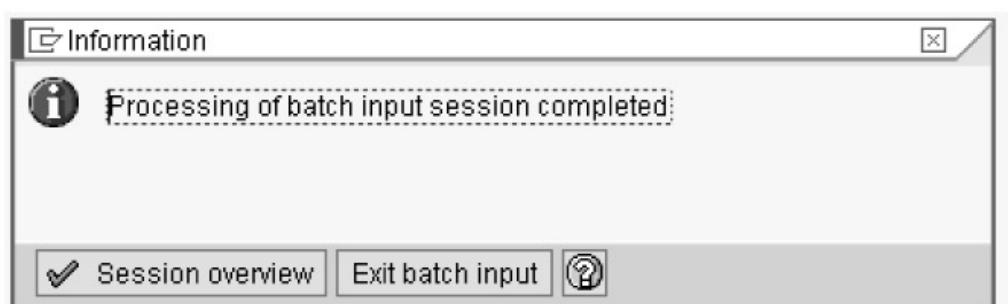


Figure 11.128: The information dialog box

10. Click the Session overview pushbutton (see Figure 11.128).

The Batch Input: Session Overview screen appears, as shown in Figure 11.129:

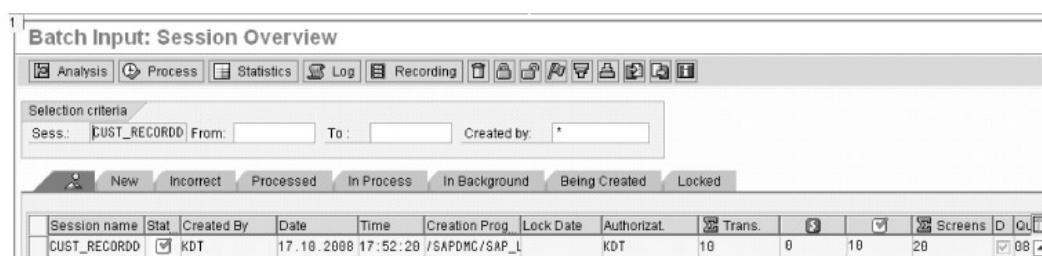


Figure 11.129: Session overview

Note that the status of the CUST_RECORDD session appears as processed, shown by a tick mark. The total number of screens involved during the processing of the session is 20. The number of transactions processed is 10. If any errors occur in processing the session, they are displayed under the Transactions with errors ( column).

After the batch input session has been executed successfully, the customer master records are updated in the SAP system. The updated customer master records can be viewed by using the XD03 transaction code.

Summary

In this chapter, you have learned about the methods of data transfer from a non-SAP system to an SAP system or between two SAP systems. You have also learned how to transfer data related to material from a non-SAP system to an SAP system by using the BDC tool. Next, you have learned how to update the customer master records by using the LSMW tool. In addition, various methods, such as the direct method, the call transaction method, and the batch input with session method, have been discussed in detail.