



Data Migration of Non-SAP Systems to SAP-Systems

*Quick Introduction to Working with the
Legacy System Migration Workbench
Version 1.8.0*

January 28, 2003



Table of Contents

| | | |
|----------|---|-----------|
| 1 | INTRODUCTION..... | 4 |
| 1.1 | PURPOSE OF THIS INTRODUCTION..... | 4 |
| 1.2 | LSM WORKBENCH: WHAT IS IT? | 4 |
| 1.3 | SUPPORTED RELEASES | 4 |
| 1.4 | COSTS | 4 |
| 1.5 | DELIVERY | 4 |
| 1.6 | LSM WORKBENCH VERSIONS..... | 4 |
| 1.7 | SUPPORT | 5 |
| 1.8 | SIGNIFICANCE OF DATA MIGRATION..... | 5 |
| 1.9 | BASIC PRINCIPLES OF THE LSM WORKBENCH | 5 |
| 2 | PRECONDITIONS..... | 7 |
| 3 | STARTUP AND PREPARATIONS..... | 8 |
| 3.1 | AUTHORIZATIONS | 8 |
| 3.2 | INITIAL TRANSACTION | 8 |
| 3.3 | PROJECT, SUBPROJECT AND OBJECT | 9 |
| 3.4 | USER GUIDANCE | 10 |
| 3.5 | FIELD MAPPING ON PAPER | 12 |
| 3.6 | CREATE OBJECT OVERVIEW..... | 13 |
| 3.7 | ADMINISTRATION..... | 17 |
| 3.8 | RECORDINGS | 17 |
| 3.9 | PREPARATIONS FOR USING IDOC INBOUND PROCESSING | 18 |
| 4 | GENERAL TIPS FOR THE PROCEDURE IN A MIGRATION USING LSMW AND DX- WORKBENCH | 22 |
| 4.1 | SAP-CUSTOMIZING..... | 23 |
| 4.2 | WHICH DATA SHOULD BE MIGRATED?..... | 23 |
| 4.3 | RUN THE MANUAL PROCESS | 23 |
| 4.4 | WHICH IMPORT TECHNIQUE WILL BE USED? | 23 |
| 4.5 | CREATE THE RECORDING..... | 23 |
| 4.6 | FILL OBJECT ATTRIBUTES | 24 |
| 4.7 | DETERMINE THE SOURCE STRUCTURES..... | 24 |
| 4.8 | CREATE TEST DATA IN DX-WORKBENCH..... | 24 |
| 4.9 | MAPPING ON PAPER..... | 24 |
| 4.10 | MAINTAIN STRUCTURE RELATIONS | 24 |
| 4.11 | DATA EXPORT | 25 |
| 4.12 | ENTER THE RULES AND TEST OUT OF LSMW | 25 |
| 4.13 | CREATE PROJECT, SUBPROJECT AND RUN IN DX-WORKBENCH..... | 25 |
| 4.14 | IMPORT THE DATA | 25 |
| 5 | DATA MIGRATION – STEP BY STEP..... | 26 |
| 5.1 | MAINTAIN OBJECT ATTRIBUTES | 26 |
| 5.2 | MAINTAIN SOURCE STRUCTURES..... | 27 |
| 5.3 | MAINTAIN SOURCE FIELDS | 29 |
| 5.3.1 | <i>Create Individual Source Fields.....</i> | 29 |
| 5.3.2 | <i>Maintain Source Fields in Table Form.....</i> | 31 |
| 5.3.3 | <i>Copy Source Fields from Other Sources.....</i> | 32 |
| 5.4 | MAINTAIN STRUCTURAL RELATIONSHIPS | 33 |
| 5.5 | MAINTAIN FIELD MAPPING AND CONVERSION RULES | 35 |
| 5.5.1 | <i>For the Advanced User: Display Variant, Processing Times</i> | 38 |
| 5.5.2 | <i>For the Advanced User: Global Variables.....</i> | 40 |
| 5.5.3 | <i>For the Advanced User: Global Functions.....</i> | 41 |
| 5.5.4 | <i>For the Advanced User: Reusable Rules — Naming Conventions.....</i> | 42 |
| 5.6 | MAINTAIN FIXED VALUES, TRANSLATIONS AND USER-WRITTEN ROUTINES..... | 43 |
| 5.7 | SPECIFY FILES | 46 |
| 5.8 | USE WILDCARDS IN FILE NAMES | 51 |

| | | |
|-----------|--|-----------|
| 5.9 | ASSIGN FILES | 53 |
| 5.10 | READ DATA | 53 |
| 5.10.1 | Display Read Data | 54 |
| 5.11 | CONVERT DATA | 54 |
| 5.11.1 | General Remarks..... | 54 |
| 5.11.2 | Additional Function for BAPI/IDoc | 55 |
| 5.12 | DISPLAY CONVERTED DATA | 56 |
| 5.13 | IMPORT DATA | 56 |
| 5.13.2 | Import Data with Direct Input | 56 |
| 5.13.3 | Import Data with BAPI or IDoc Technique | 57 |
| 6 | RECORDINGS..... | 58 |
| 6.1 | DETAILED DESCRIPTION OF THE PROCESS | 58 |
| 7 | TRANSPORT LSMW PROJECTS..... | 61 |
| 7.1 | GENERATE CHANGE REQUEST | 61 |
| 7.2 | EXPORT PROJECT | 61 |
| 7.3 | IMPORT PROJECT | 62 |
| 8 | PERIODIC DATA TRANSFER..... | 63 |
| 9 | LONG TEXTS..... | 66 |
| 9.1 | LONG TEXTS IN THE SAP SYSTEM | 66 |
| 9.2 | DETERMINE TEXT KEY STRUCTURE..... | 67 |
| 9.3 | DEVELOP OBJECTS FOR LONG TEXTS VIA OBJECT 0001..... | 68 |
| 9.4 | DEVELOP OBJECTS FOR LONG TEXTS VIA OBJECT 2000..... | 71 |
| 9.5 | IMPORT TEXTS | 75 |
| 10 | TIPS AND TRICKS..... | 76 |
| 10.1 | DETERMINE THE TRANSACTION CODE AT RUNTIME..... | 76 |
| 10.2 | SKIP A RECORD | 76 |
| 10.3 | SKIP ALL RECORDS OF A TRANSACTION | 76 |
| 10.4 | DUPLICATE A RECORD | 76 |
| 10.5 | ERROR MESSAGES IN THE CONVERSION LOG..... | 77 |
| 10.6 | EXTRA HANDLING FOR “POS-IDOCs” | 77 |
| 11 | UPGRADE FROM LSMW 1.0 TO LSMW 1.8..... | 79 |
| 11.1 | DIFFERENCES BETWEEN VERSION 1.0 AND VERSION 1.8 OF THE LSM WORKBENCH | 79 |
| 12 | TRANSFER OF LSMW DATA FROM VERSION 1.0 TO VERSION 1.8..... | 81 |
| 13 | UPGRADE FROM LSMW 1.5 TO LSMW 1.8..... | 82 |
| 13.1 | NOTES ON THE UPGRADE TO LSMW 1.8..... | 82 |
| 13.2 | CORRECTIONS | 82 |
| 13.3 | DEVELOPMENTS | 82 |
| 14 | FINAL REMARKS..... | 84 |

1 Introduction

1.1 Purpose of this Introduction

This introduction is intended to allow a quick entry into the work with the Legacy System Migration Workbench Version 1.8 ("LSM Workbench").

1.2 LSM Workbench: What is it?

The LSM Workbench is an SAP-based tool that supports You when transferring data from non-SAP systems ("Legacy Systems") to SAP systems¹ once or periodically.

The tool supports conversion of data of the legacy system in a convenient way. The data can then be imported into the SAP system via batch input, direct input, BAPIs or IDocs.

Furthermore, the LSM Workbench provides a recording function that allows to generate a "data migration object" in an entry or change transaction.

1.3 Supported Releases

LSMW Version 1.8 can be implemented for systems that use the following releases: 4.6A, 4.6B, 4.6C and 4.6D. LSMW Version 1.72 is available for releases from 4.0 up to (but not including) 4.6.

LSMW Version 1.0 can be implemented for releases older than 4.0.

1.4 Costs

SAP makes this tool available to their customers and partners **free of charge**.

1.5 Delivery

The LSM Workbench is not part of the standard SAP system. If you are interested in this product, please contact SAP via:

- SAPNet (Online Service System): component BC-SRV-DX-LSM or
- SAPNet: <http://service.sap.com/lsmw>

There you find all available information and documentation and the software itself (transport file). Among others, the following accompanying material is available:

- Check list for the usage of the LSM Workbench
- Presentation of the LSM Workbench (PowerPoint presentation)

1.6 LSM Workbench Versions

Version 1.0 of the LSM Workbench was made available to about 350 interested customers and partners in the frame of the First Customer Shipment between March 1998 (CeBIT) and middle of August 1998. The resulting experiences and feedback were taken into account in the further development.

¹ SAP-System includes SAP-R/3, APO 2.0 and CRM 3.0; LSMW has its own namespace and there are no problems known with the industry solutions yet

LSMW Version 1.0 was made globally available in August 1998 and since then has been requested more than 1000 times. LSMW Versions 1.5 and 1.6 were published in 1999.

LSMW version 1.7 has been available since September 2000 and was replaced by LSMW Version 1.8 in February 2003. (SAP R/3 Basis Release 4.6 is mandatory for Version 1.8.)

For all persons already familiar with version 1.0 of the LSM Workbench, section 11 provides a compact overview of the modifications in version 1.7 compared with version 1.0.

1.7 Support

For problem messages via SAPNet (Online Service System), entry "**BC-SRV-DX-LSM**" is available in the component hierarchy. When you enter a message, please specify the number of the LSM Workbench version you are using. (To display the version number, select *Extras* → *Display LSMW version* in the initial screen of transaction LSMW.)



Note: If problems occur after step *Convert data*, please directly contact the special department responsible for the module (FI, CO, MM, SD, HR, etc.). All steps following the data conversion are not LSM Workbench functions. Therefore the LSMW team cannot provide support for these functions.

1.8 Significance of Data Migration

Data migration comes in the end of SAP implementation. At this time, the SAP system is normally installed and application customizing is finished (in the best case).

An examination of SAP implementation projects by SAP Consulting has shown that data migration is about 20% of the total implementation expenses. This portion may rise up to 40% in smaller implementation projects.

A significant reduction of the expenses for data migration results in a corresponding significant reduction of the total project budget and the project runtime.

The experiences with the **LSM Workbench** in SAP implementation projects are very promising: Both the expenses and the costs of data migration could be reduced significantly.

1.9 Basic Principles of the LSM Workbench

The **LSM Workbench** was developed on the basis of the **R/2-R/3 Migration Workbench** that has been used many hundred times in the past and is still used presently.

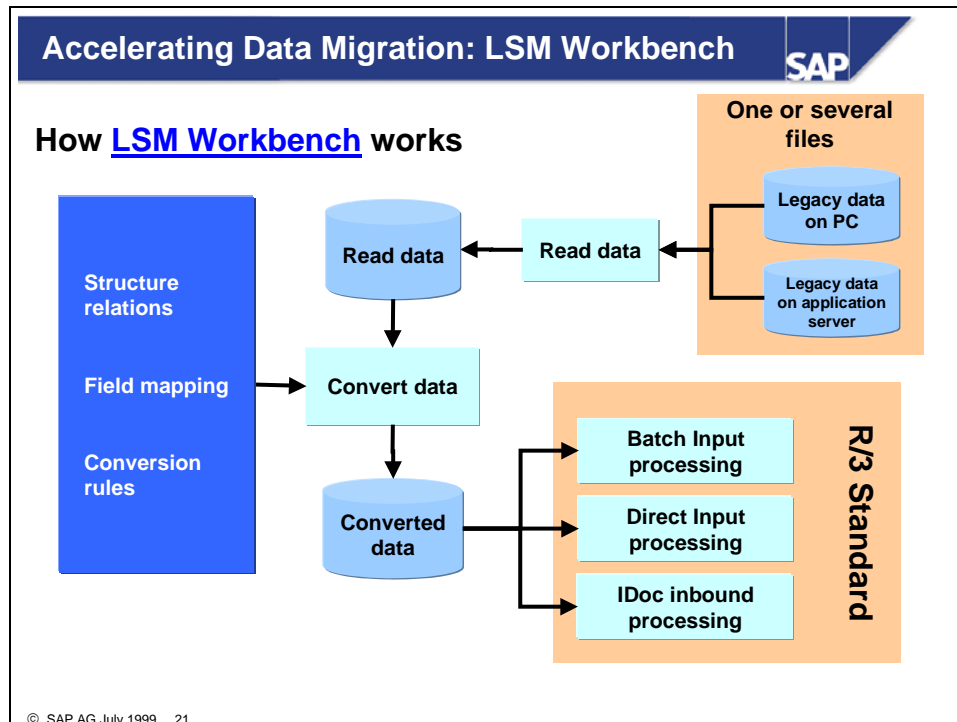
The following concepts and techniques from the **R/2-R/3 Migration Workbench** were adopted:

- Business objects instead of individual tables or field contents are migrated.
- The conversion rules to be defined are flexible and can be customized in the frame of migration customizing to meet the actual situation in the project in the user system.
- Preprogrammed conversion programs are not delivered. These programs are rather generated on the basis of the defined rules.

The **LSM Workbench** was developed on the basis of the following principles:

- Most of the functions should reside in the SAP system. No collection of individual programs on different platforms.
- The quality and consistence of the data imported into the SAP system should be more important than speed and performance of data migration.
- Existing knowledge and coding should be used.
- The developed "mapping" and rules should be reusable and thus be used repeatedly in projects.

On this basis, a concept was developed that is represented in the following chart:



Schematic Flow of Data Migration with the LSM Workbench

The main advantages of the **LSM Workbench**:

- Part of the SAP system and thus independent of individual platforms
- A variety of technical possibilities of data conversion:
- Data consistency due to standard import techniques:
 - Batch input
 - Direct input
 - BAPIs (Business Application Programming Interfaces)
 - IDocs (Intermediate Documents)

The import technique to be used in an individual case depends on the business object.

- Generation of the conversion program on the basis of defined rules
- Clear interactive process guide
- Interface for data on frontend and on application server
- Creation of data migration objects on the basis of recorded transactions
- Charge-free for SAP customers and SAP partners

2 Preconditions

The LSM Workbench is a tool that supports data transfer from non-SAP systems to SAP systems. The main functions of the LSM Workbench are:

1. Import data (legacy data in spreadsheet tables and/or sequential files)
2. Convert data (from source format to target format)
3. Import data (into the database of the SAP application)

Before you can apply the LSM Workbench, you absolutely need a data migration **concept**. The following items should be considered in particular:

- Make sure that SAP customizing is finished.
- Determine the data contained in your legacy system and which of the data will be required in the future (with respect to business operation).
- Decide whether it makes sense to use this tool with respect to the data volume to be transferred. It may be easier to transfer very small data volumes into the SAP system manually. In the case of a very large data volume, batch input or IDoc techniques may cause extremely long runtimes. A rough landmark for estimating the required time: 10 000 records per hour where this value may vary considerably depending on the hardware available.
- Identify the transaction(s) in the SAP system via which you want to import the data into your SAP system. It may be relevant whether you need the data for statistical analysis or for further processing in the system.
- Run the relevant transaction in the SAP system manually with test data from the legacy system and see which fields must be filled. There may be required fields that do not correspond to data fields in the legacy system. In such a case, you should better assign a fixed value or establish an optional field for data transfer.
- Map the fields in advance in written form: Assign the source fields to the target fields.
- Determine the form in which non-SAP data will be transferred into the SAP system (e.g. via "Move" or according to a rule).
- If applicable, define the translation rules (LSMW-internal name: "translation").
- In which way will the data be extracted from the non-SAP system? Note: The LSMW does not extract data.
- In which form is the legacy data available? Determine accordingly which of the LSMW functions will be applied.
- If only a part of your legacy system will be replaced by SAP, determine which function will be provided by the SAP system and which by the legacy system. If applicable, set up a concept of data flows and interface architecture.

These questions will be answered differently for every customer and **must absolutely be answered before the tool will be used!**

3 Startup and Preparations

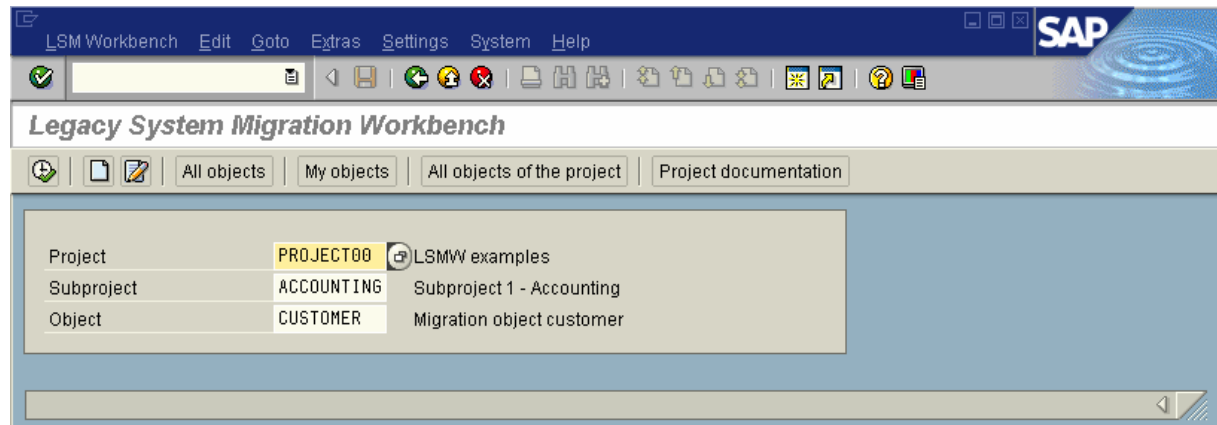
3.1 Authorizations

| Authorization level | Profile | Function |
|---------------------|-------------|---|
| Display | B_LSMW_SHOW | The user can display all projects he has the authorization for and their work steps. He/she cannot switch to change mode. |
| Execute | B_LSMW_EXEC | The user can display data, and read, convert and import data. |
| Change | B_LSMW_CHG | The user has "Execute" authorization, and can change and copy objects. |
| Administrate | B_LSMW_ALL | The user can use all functions made available by the tool. |

Please note: The profiles listed above are not included in the profiles of the standard SAP system, but are added to the installation client of the SAP system during the installation of LSMW. Step 2 of the installation procedure (client cascade) distributes the profiles to the clients of the SAP system. Therefore you have to adopt them (enter the project names) or create new profiles and add the required profiles to your user master record.

3.2 Initial Transaction

To start working with the LSM Workbench, use transaction **LSMW**:



LSM Workbench – Initial Screen

3.3 Project, Subproject and Object

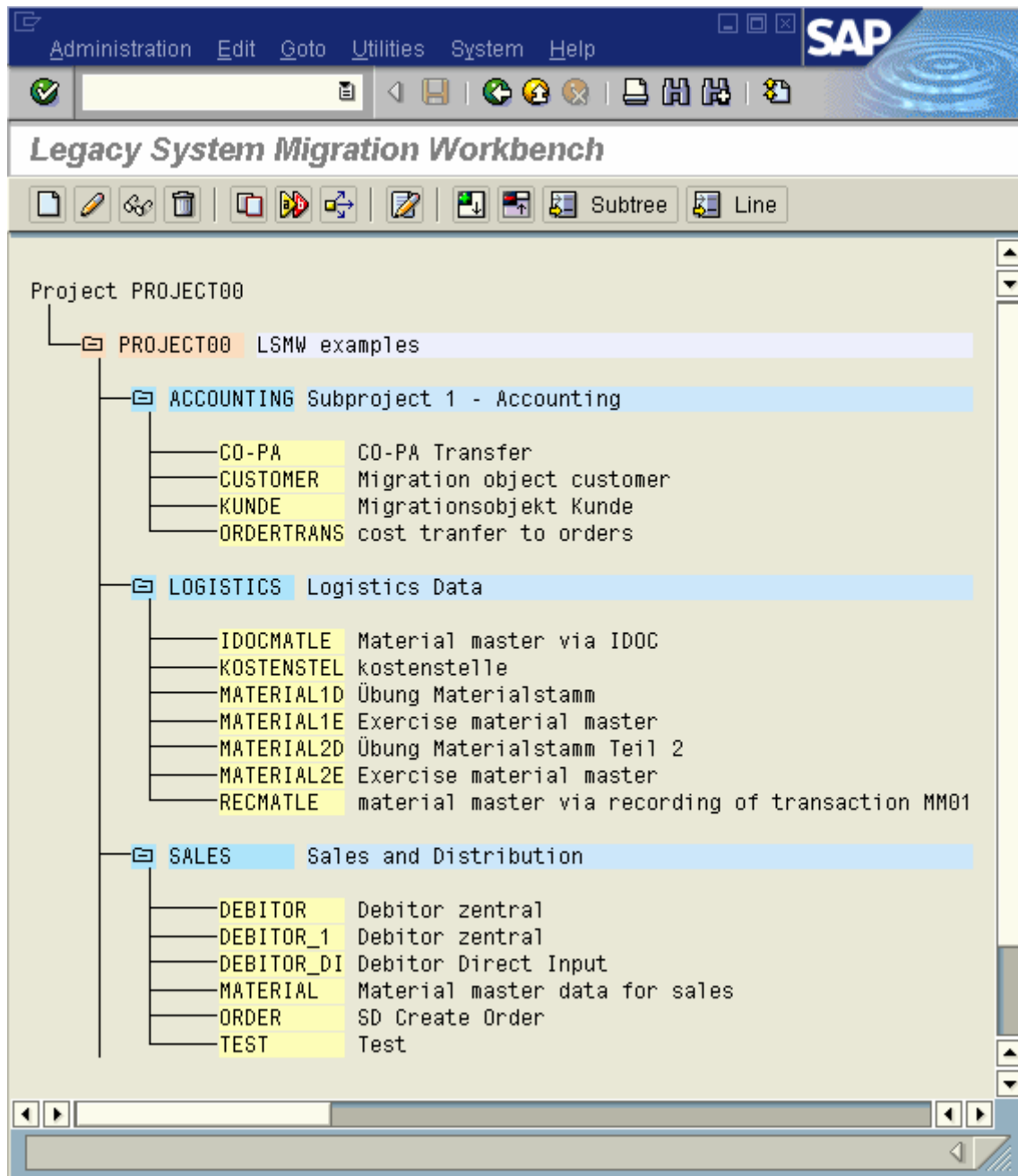
On the initial screen, you can create a new project, corresponding subprojects and objects via *Edit -> Create new entry*.

- **Project:** An ID with a maximum of 10 characters to name your data transfer project. If you want to transfer data from several legacy systems, you may create a project e.g. for every legacy system.
- **Subproject:** An ID with a maximum of 10 characters that is used as further structuring attribute.
- **Object:** An ID with a maximum of 10 characters to name the business object.

In the initial screen, *All objects* provides a list of all projects created already. *My objects* displays a list of all objects you created personally. *All objects of the project* displays all objects of the selected project as tree structure. *Project documentation* displays any documentation written for the individual popups and processing steps. you can print the project documentation out, send it and save it in various file formats.

Select *Documentation* to enter your notes. After clicking, a popup is displayed in which you can write down your personal documentation. The documentation function is additionally available under *Administration* and *Recordings* in the first seven steps of data migration.

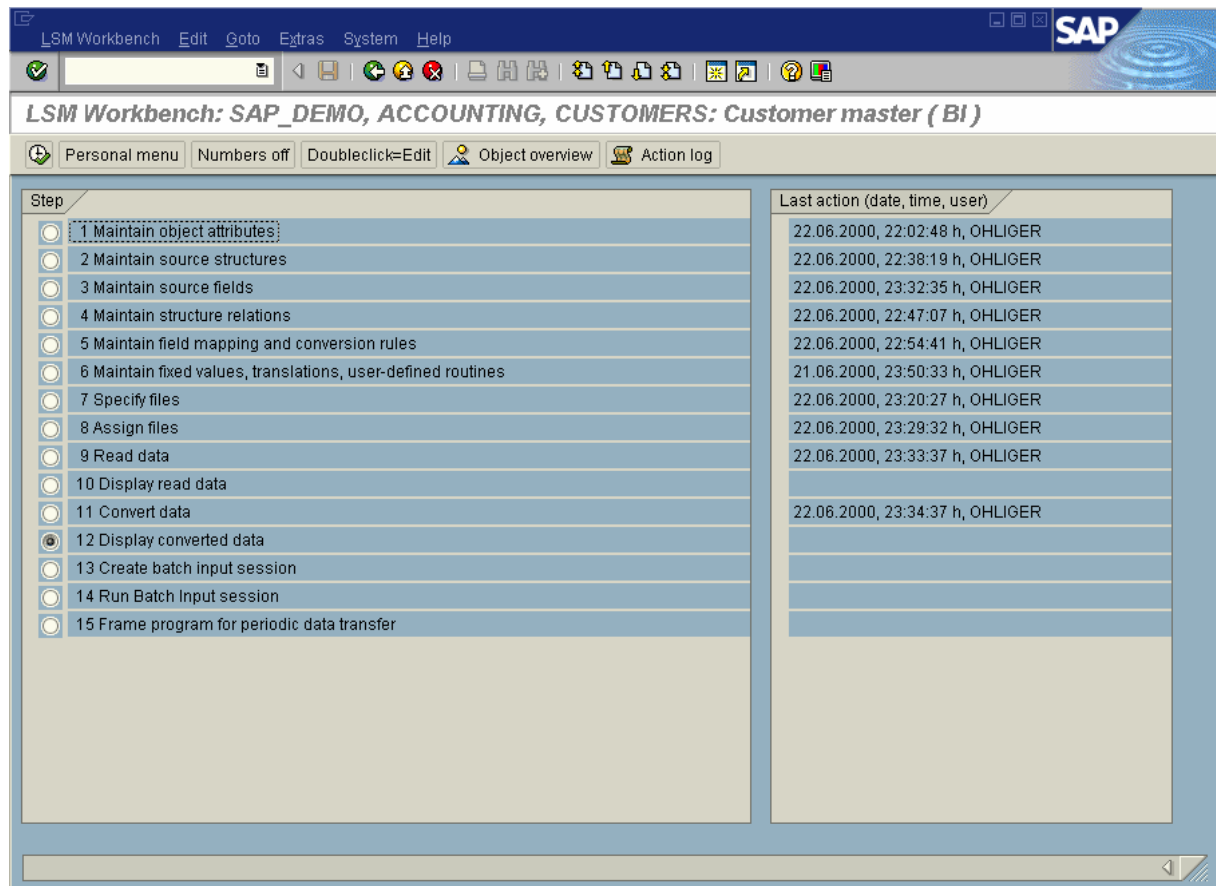
Below, you find an example for a project with several subprojects and objects. This representation is displayed by pushing the button *All objects of the project*.



Example for a Project Structure

3.4 User Guidance

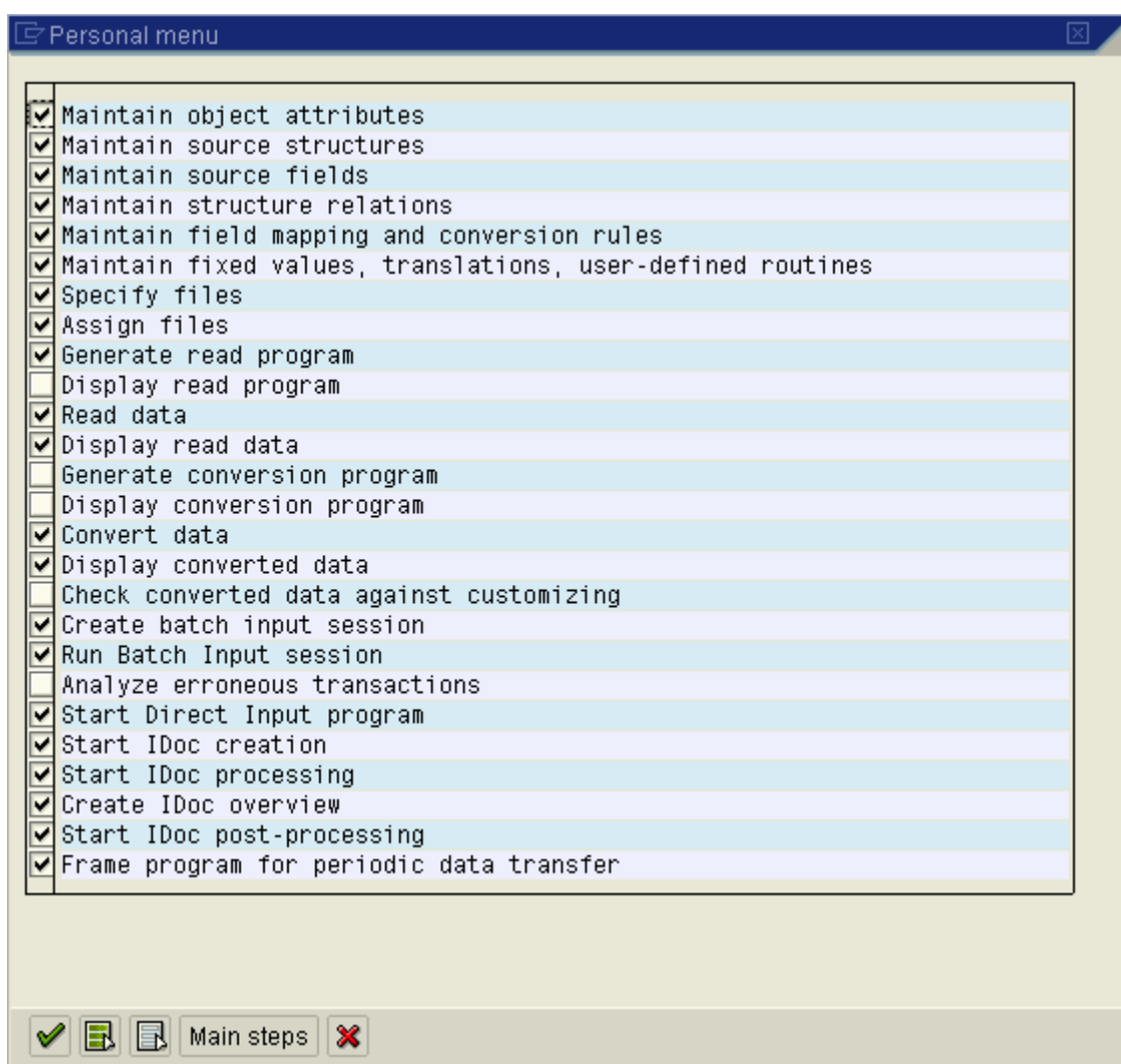
After selecting an object, *ENTER* or *CONTINUE* leads you to the interactive process guide. Here you are guided through the individual steps of data migration.



LSM Workbench: The Main Steps of Data Migration

This popup provides the following functions:

- *Execute*: Executed the selected processing step.
- *Personal menu*: Here you can make an individual selection from the displayed processing steps. Pressing button "Main steps" automatically activates all processing steps mandatory for a data conversion.
- *Numbers On or Off*: you can activate or deactivate the numbering of the individual processing steps.
- *Double click = Display* or *Double click = Change*: Here, you can determine whether display mode or change mode is selected by double clicking.
- *Object overview*: Displays all information on the selected object.
- *Action log*: Displays a detailed overview (date, user name, time) for all processing steps already carried out. you can reset the action log via *Extras → Reset action log*. This action is stored with a reference to the user and the date.



Personal Menu: All Processing Steps Available

3.5 Field Mapping on Paper

Before you start working with the LSM Workbench, you should first map the required object on paper. To do this, create and print out the "object overview".

At this time, the overview only displays the list and description of the SAP structures and their fields. You may use it as a guideline for assigning the corresponding structures and fields of the source system to these target structures and target fields.

You can also download the overview in table form and fill the table in Excel to have the mapping as a file on your PC.

3.6 Create Object Overview

This function is available as pushbutton in order to enable you to create an object overview at any time:

Legacy System Migration Workbench

System Help

LSM Workbench: Object Overview (List)

Overview as table Overview for reusable rules

Legacy System Migration Workbench: Object information

This list was created 21.06.2000 at 21:29:33 h in the R/3 System U46 .

Project: SAP_DEMO SAP Demo
 Subproject: ACCOUNTING Accounting
 Object: CUSTOMERS Customer master (BI)
 Data transfer program: RFBIDE00
 Data transfer method: B (Batch input)

Source structures

CUSTOMER_HEADER - Customer Master: Header
 CUSTOMER_CONTACTS - Customer master: Contact persons

Target struct.

BGR00 - Batch Input Structure for Session Data
 BKN00 - Customer Master Record Transaction Data for Batch Input
 BKNA1 - General Customer Master Record Part 1 (Batch Input)
 BKNB1 - Customer Master Record Company Code Data (Batch Input)
 BKNVK - Debitor Master Contact Person (Batch Input Structure)

Structure relations

BGR00 <=== CUSTOMER_HEADER
 BKN00 <=== CUSTOMER_HEADER
 BKNA1 <=== CUSTOMER_HEADER
 BKNB1 <=== CUSTOMER_HEADER
 BKNVK <=== CUSTOMER_CONTACTS

Source fields

CUSTOMER_CONTACTS - Customer master: Contact persons

Object Overview in list form: General Data, Structures, Structure Relations

System Help

LSM Workbench: Object Overview (List)

Overview as table Overview for reusable rules

Source fields

CUSTOMER_CONTACTS - Customer master: Contact persons

| Field | Description | Type | N | Lgth | Position | Offset |
|----------------|-----------------|---------|---|------|----------|--------|
| CUSTOMERNUMBER | Customer Number | Type: N | | 004 | 0001 | 000 |
| TITLE | Title | Type: C | | 004 | 0002 | 004 |
| FIRSTNAME | FirstName | Type: C | | 030 | 0003 | 008 |
| LASTNAME | LastName | Type: C | | 030 | 0004 | 038 |
| CONTACTPHONE | ContactPhone | Type: C | | 014 | 0005 | 068 |
| DEPARTMENT | Department | Type: C | | 020 | 0006 | 082 |
| FUNCTION | Function | Type: C | | 020 | 0007 | 102 |

CUSTOMER_HEADER - Customer Master: Header

| Field | Description | Type | N | Lgth | Position | Offset |
|----------------|-----------------|---------|---|------|----------|--------|
| CUSTOMERNUMBER | Customer number | Type: N | | 004 | 0001 | 000 |
| NAME | Customer name | Type: C | | 060 | 0002 | 004 |
| COUNTRY | Country code | Type: C | | 002 | 0003 | 064 |
| ZIPCODE | Zip code | Type: C | | 010 | 0004 | 066 |
| CITY | City | Type: C | | 020 | 0005 | 076 |
| STREET | Street | Type: C | | 060 | 0006 | 096 |
| PHONE | Phone | Type: C | | 030 | 0007 | 156 |
| FAX | Fax | Type: C | | 030 | 0008 | 186 |
| VATNR | VAT number | Type: C | | 020 | 0009 | 216 |

Target fields

BGR00 - Batch Input Structure for Session Data

| Field | Description | Type | Lgth |
|--------|--|------------|------|
| STYPE | Batch Input Interface Record Type | Type: CHAR | 001 |
| GROUP | Queue group name | Type: CHAR | 012 |
| MANDT | Client | Type: CLNT | 003 |
| USNAM | Queue user ID / for historical reasons | Type: CHAR | 012 |
| START | Queue start date | Type: DATS | 010 |
| XKEEP | Indicator: Hold batch input session after processing ? | Type: CHAR | 001 |
| NODATA | No Batch Input Exists for this Field | Type: CHAR | 001 |

BKN00 - Customer Master Record Transaction Data for Batch Input

| Field | Description | Type | Lgth |
|-------|-----------------------------------|------------|------|
| STYPE | Batch Input Interface Record Type | Type: CHAR | 001 |
| TCODE | Transaction code | Type: CHAR | 020 |
| KUNNR | Customer number | Type: CHAR | 010 |

Object Overview: Source Structures / Target Structures

System Help

LSM Workbench: Object Overview (List)

Overview as table | Overview for reusable rules

Struct.: BKN00

BKN00-STYPE
Rule type: Default settings
BKN00-STYPE = '1'.

BKN00-TCODE
Rule type: Constant
BKN00-TCODE = 'XD01'.

BKN00-KUNNR <== CUSTOMER_HEADER-CUSTOMERNUMBER
Rule type: Prefix
concatenate 'C_'
CUSTOMER_HEADER-CUSTOMERNUMBER
into BKN00-KUNNR.

This is the customer number

BKN00-BUKRS
Rule type: Fixed value (reusable)
BKN00-BUKRS = FV_BUKRS.

For company code a fixed value is used

BKN00-VKORG
Rule type: Constant
BKN00-VKORG = '0001'.

BKN00-KTOKD
Rule type: Constant
BKN00-KTOKD = 'KUNA'.

Struct.: BKN11

BKN11-STYPE
Rule type: Default settings

Object Overview: Rules

System Help

LSM Workbench: Object Overview (Table)

Overview as list | Overview for reusable rules

| Target fld | Field name | Field description | Type | Length | Source fields | Field name | Length | Conversion | method | Codin |
|--|------------|--|------|--------|---------------|----------------------|--------|------------------------|--------|-------|
| BGR00 - Batch Input Structure for Session Data | | | | | | | | | | |
| STYPE | | Batch Input Interface Record Type | CHAR | 001 | | | | Default settings | | BGR00 |
| GROUP | | Queue group name | CHAR | 012 | | | | Default settings | | BGR00 |
| MANDT | | Client | CLNT | 003 | | | | Default settings | | BGR00 |
| USNAM | | Queue user ID / for historical reasons | CHAR | 012 | | | | Default settings | | BGR00 |
| START | | Queue start date | DATS | 010 | | | | Default settings | | BGR00 |
| XKEEP | | Indicator: Hold batch input session after processing ? | CHAR | 001 | | | | Default settings | | BGR00 |
| NODATA | | No Batch Input Exists for this Field | CHAR | 001 | | | | Default settings | | BGR00 |
| BKN00 - Customer Master Record Transaction Data for Batch Input | | | | | | | | | | |
| STYPE | | Batch Input Interface Record Type | CHAR | 001 | | | | Default settings | | BKN00 |
| TCODE | | Transaction code | CHAR | 020 | | | | Constant | | BKN00 |
| KUNNR | | Customer number | CHAR | 010 | | CUSTOMER_HEADER-CUST | 004 | Prefix | | conca |
| | | | | | | | | | | Das 1 |
| BUKRS | | Company code | CHAR | 004 | | | | Fixed value (reusable) | | BKN00 |
| VKORG | | Sales organization | CHAR | 004 | | | | Constant | | BKN00 |
| VTWEG | | Distribution channel | CHAR | 002 | | | | Constant | | BKN00 |
| SPART | | Division | CHAR | 002 | | | | Constant | | BKN00 |
| KTOKD | | Customer Account Group | CHAR | 004 | | | | Constant | | BKN00 |
| KKBER | | Credit control area | CHAR | 004 | | | | Constant | | BKN00 |
| BKN11 - General Customer Master Record Part 1 (Batch Input) | | | | | | | | | | |
| STYPE | | Batch Input Interface Record Type | CHAR | 001 | | | | Default settings | | BKN11 |

Object overview in table form



Note: you may use this before the development of field mapping and rules to print out the SAP structures for an object including the record description in order to carry out "mapping on paper".

System Help

SAP

LSM Workbench: Overview for reusable rules

Overview as list Overview as table

Reusable rules - Fixed Values

Fixed value: **BUKRS**

Value: 0001

Reusable rules - Translations

Rule : **LAND1**

1:1 Translation table

| Old value | New value |
|-----------|-----------|
| A | AT |
| CA | CA |
| D | DE |
| I | IT |

Interval translation table

| From value | To value | New value |
|------------|----------|-----------|
| XX | ZZ | ??? |

Reusable rules - User routines

Routine: **PSTLZ**

ABAP Coding

```
* Eigene Routine
form ur_PSTLZ
  using p_in
  changing p_out.
  concatenate 'DE' p_in into p_out.
endform.
```

Overview of Reusable Rules

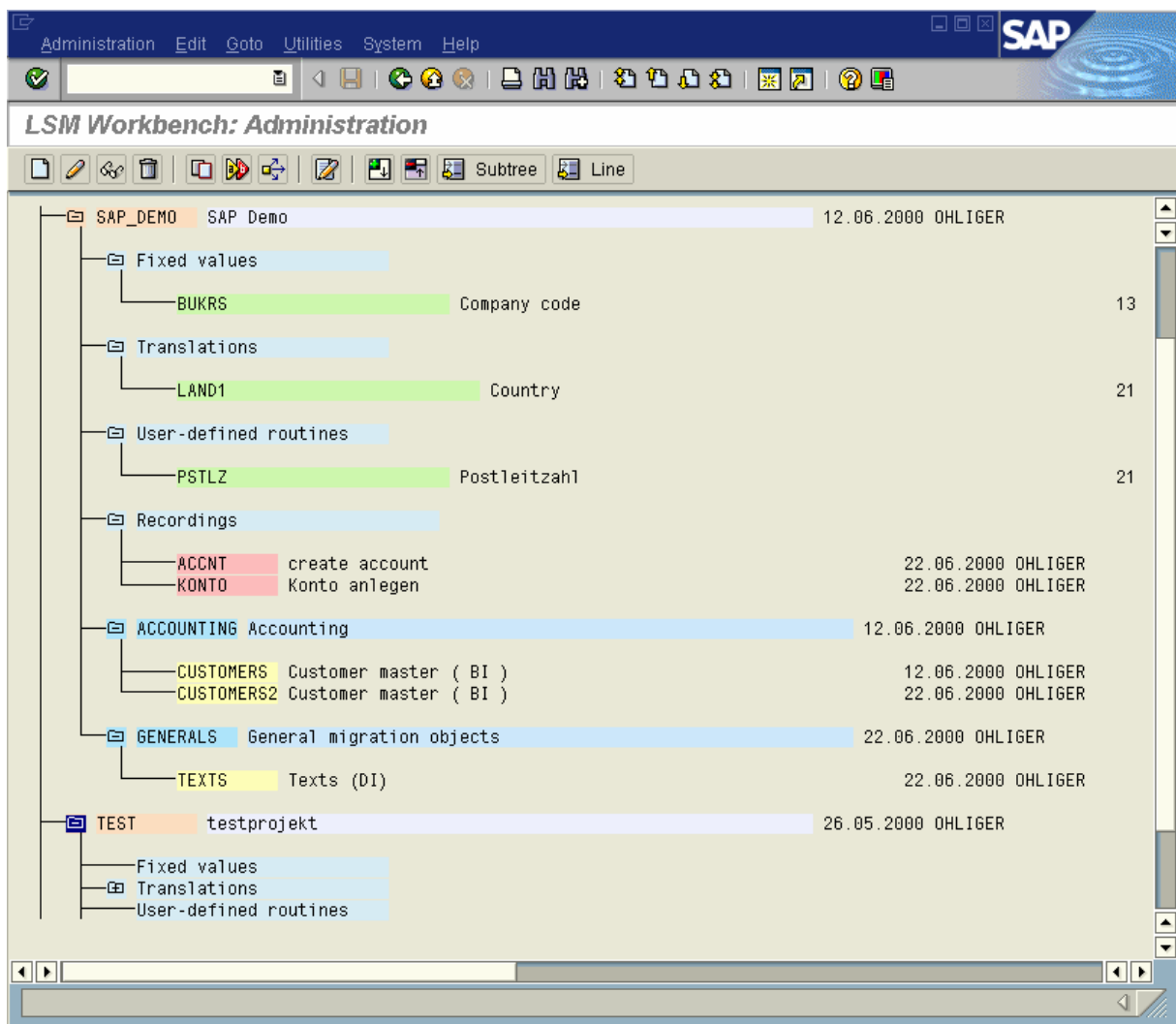
3.7 Administration

In the initial screen, you can display the administration functions via *Goto*→ *Administration*. Here you can find a list of all existing projects.

It enables you to create, process, display, delete, copy or rename projects, subprojects, objects and reusable rules.

By double-clicking on an entry you can branch to the entry display.

By positioning the cursor on an entry, you can store a personal note via *Documentation*. For every processing action, the name of the person who made the last change and the date of the last change are stored.



LSM Workbench: Administration

3.8 Recordings

In the initial screen, you can display the recording functions via *Goto*→ *Recordings*.

If neither a standard batch input program nor a standard direct input program nor an IDoc is available for a data object, you can create a new object using the recording function of the *LSM Workbench*. However, also in cases where a standard program is available, it may make sense to use the recording function in order to reduce the number of target fields.



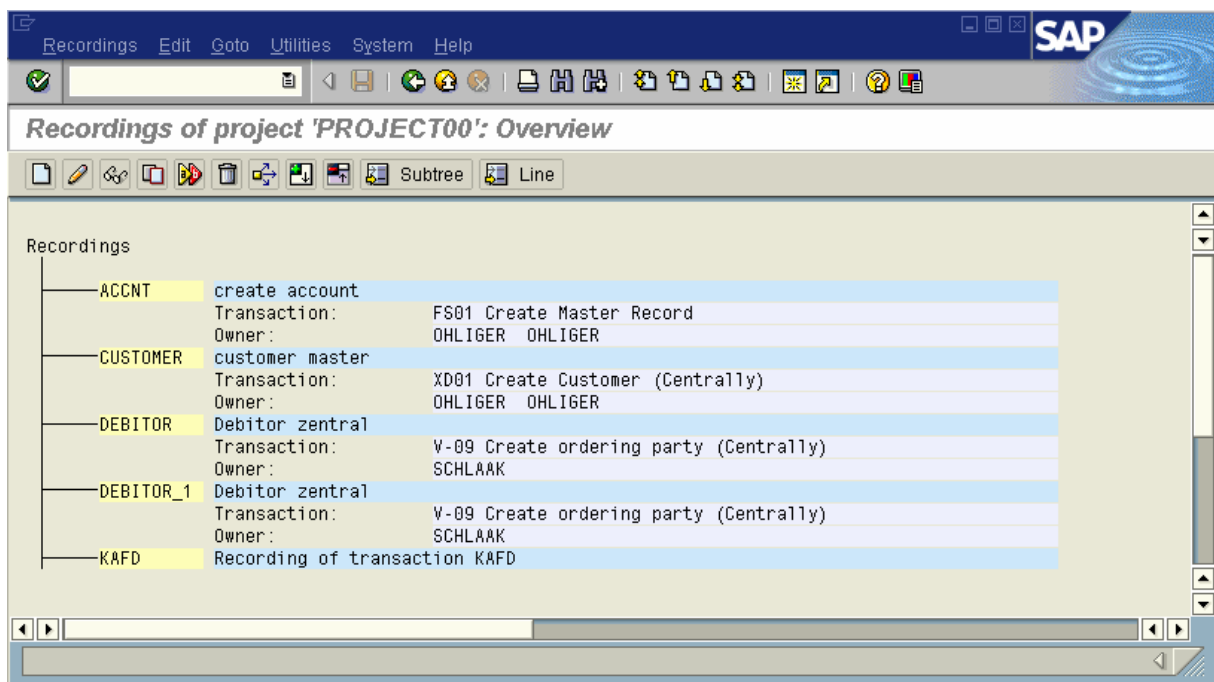
Note: The recording function records a fixed screen sequence. It cannot be used for migrating data containing a variable number of items or for transactions with dynamic screen sequences !



Tip: It is possible to create a recording via SHDB, generate a program out of this recording, adopt the program to your needs and registrate the program to be able to use it in LSMW.

Use the documentation function: Make sure that you are working in change mode. Position the cursor on an entry and select *Documentation*. A popup is displayed where you can write down your own notes.

For a detailed description of the recording function see section 6.



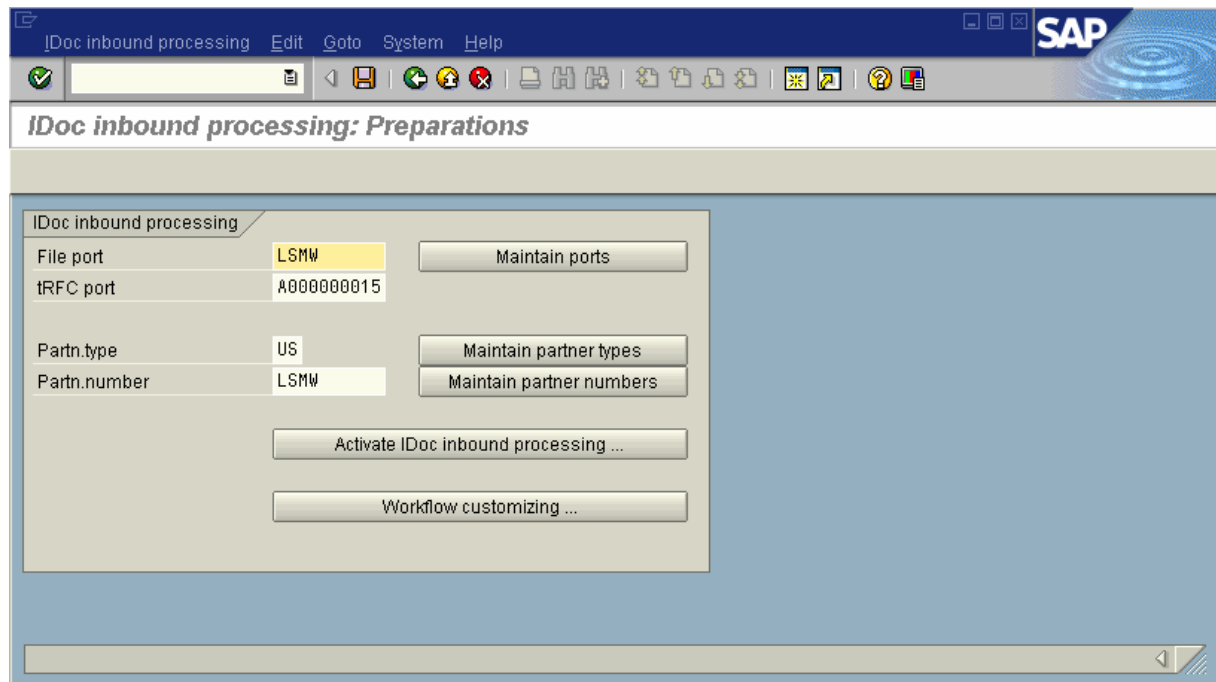
LSM Workbench: Recordings

3.9 Preparations for Using IDoc Inbound Processing

IDocs (Intermediate Documents) were developed for exchanging messages between different systems (R/3 ⇔ R/3; R/3 ⇔ R/2; R/3 ⇔ non-SAP system).

Since it is a standard interface to the SAP applications, this technique can also be used for transferring data.

To do this, however, some presettings and preparations are required (settings have to be done for each project). For a summary of these requirements see *Settings → IDoc inbound processing* in the LSM Workbench.



LSM Workbench: Settings for IDoc Inbound Processing

- The first requirement is a file port for the file transfer. If required, create a port of the file type via *Maintain ports*. To do this, position the cursor on "File" and press *Create*. You should be in change mode. SAP recommends:

| | |
|----------------------|--|
| Port: | LSMW |
| Name: | Legacy System Migration Workbench |
| Version: | 3 (IDoc record types SAP Release 4.x) |
| Outbound file | Please enter a (dummy) physical directory and a file name, i.e. 'filelsmw' |

- As an addition, you can specify a **tRFC port**. This port is required, if you do not want to create a file during data conversion but submit the data in packages directly to function module IDoc_Inbound_Aynchronous. SAP recommends:

| | |
|-------------------------|---------------------------------------|
| Port: | assigned by the system |
| Version: | 3 (IDoc record types SAP Release 4.x) |
| RFC destination: | Name of the SAP System |
| Name of port: | Legacy System Migration Workbench |

- Then the *partner type* should be defined or selected. SAP recommends:

| | |
|----------------------|-------------|
| Partner type: | "US" (User) |
|----------------------|-------------|

As of release 4.5A, this partner type is available in the standard system. Up to release 4.0B inclusive, this partner type is not available in the standard program and should be added. SAP recommends:

| | |
|----------------------|-----------|
| Partner type: | Create US |
|----------------------|-----------|

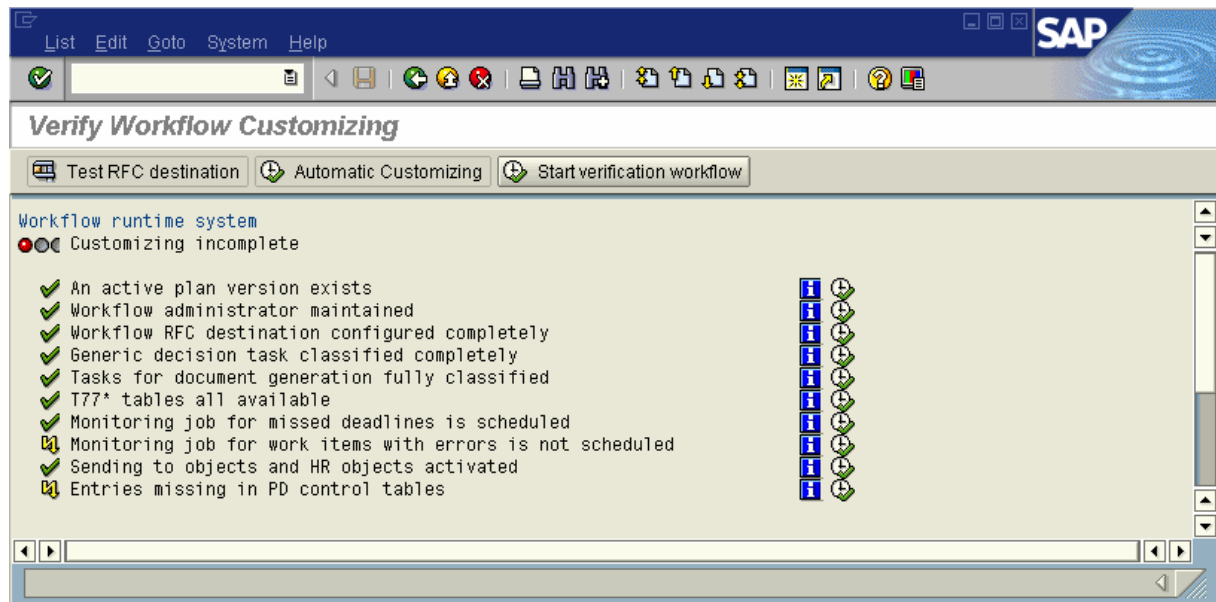
| | |
|---------------------------|-------------------------------|
| Report name: | /SAPDMC/SAP_LSMW_PARTNERTYPES |
| Form routine: | READ_USER |
| Short description: | any |

- Finally a partner number should be defined or selected. SAP recommends:

| | |
|--------------------------|--------------|
| Partner number: | LSMW |
| Partner type: | US |
| Partner status: | A (active) |
| Type: | US |
| Language: | DE or EN |
| Person in charge: | Your user ID |

- Activate IDoc inbound processing**
 - Confirm with "Yes" (to be done once for each system)
- Verify workflow customizing** (to be done once for each system)
 - The following entries of the workflow runtime system should be marked with a green check mark:
 - Workflow Administrator maintained
 - Workflow RFC destination completely configured
 - Generic decision task classified completely
 - Sending to objects and to HR objects is active

- To do this, you can start automatic customizing. After this you should set item "Monitoring job for work items with errors" to "not scheduled". (This means that you unmark the ID "Monitoring of WIs with temporary errors".) If you do not do this, the SAP system tries over and over again to post incorrect IDocs created during data migration.
- Check the function with *Test RFC destination*. The following message should be displayed:
'Ping' executed successfully. The RFC destination for the SAP Business Workflow is fully configured.



Verifying Workflow Customizing

4 General Tips for the Procedure in a migration using LSMW and DX-Workbench

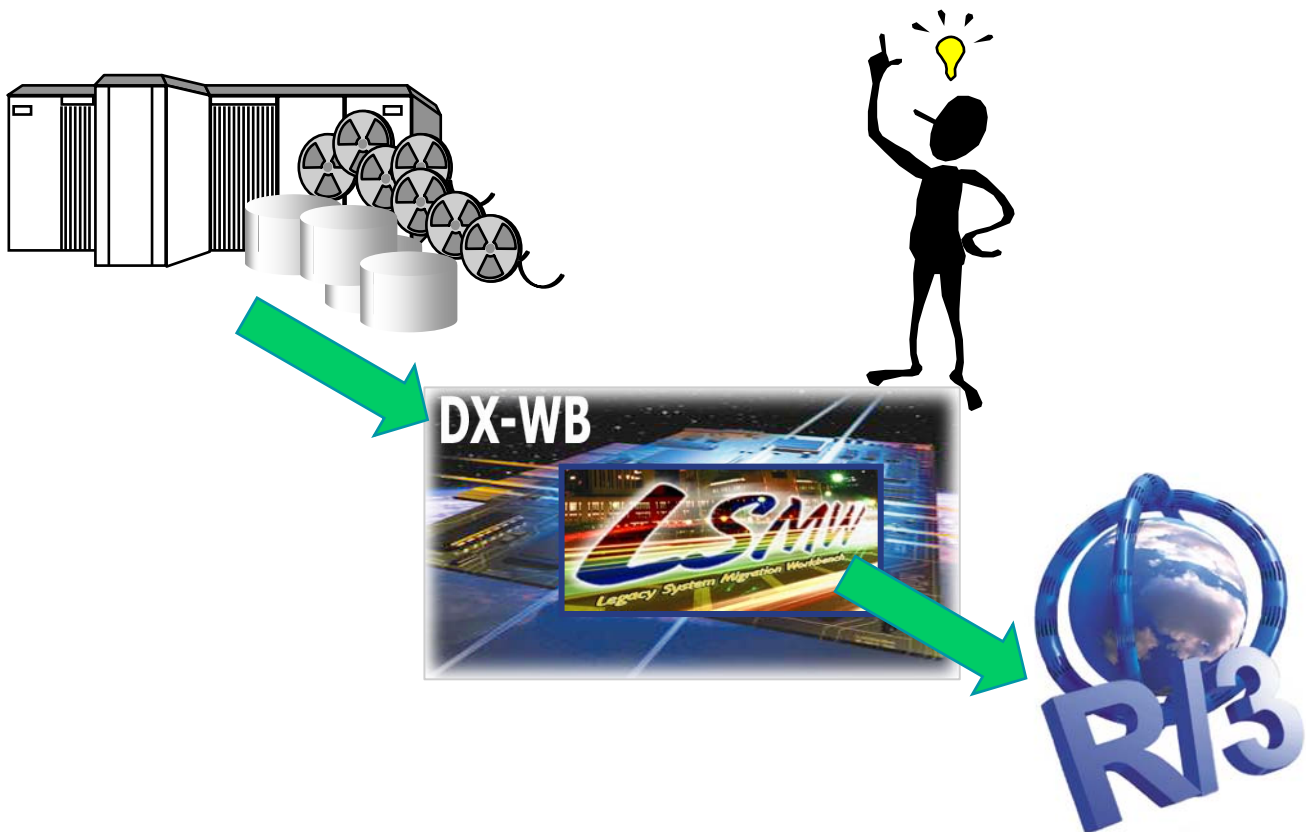
The LSM Workbench / DX-Workbench is a tool that supports the transfer of data from non-SAP systems to SAP systems.

Core functions of the LSM Workbench:

1. Import data from a legacy system
2. Converting data from its original (legacy system) format to the target (SAP) format
3. Importing the data using the standard interfaces of SAP (IDoc inbound processing, batch input, direct input)

Core functions of the DX Workbench:

1. Data import
2. Monitoring for data import
3. Create and edit test data
4. Restart functionality



LSMW and DXWB should be used in combination

➔ Recommendation

- For batch input / direct input: use the test functionality in DX-Workbench and run the whole import process from LSMW
- For BAPI / IDoc: use DX-workbench and do mapping and conversion via an LSMW object – LSMW can be called from DX-workbench as a task (from 4.6C on)

Before using the LSM Workbench or DX Workbench, you need a **concept** for data migration.

4.1 SAP-Customizing

- Make sure that the **Customizing** of your SAP system is finished.
 - The "ideal project":
 - First finish customizing
 - Then, run data migration

4.2 Which data should be migrated?

Analyze the data existing in the legacy system to determine which data will be needed in the future (also from a business-operational point of view).

4.3 Run the manual process

- Identify the transaction(s) in the SAP system you want to use for bringing the data into the SAP System. Here, it may also be relevant whether the data is required for statistical (evaluation) purposes or for further processing in the system.
- Test the relevant transaction in the SAP system manually with test data from the old system and make sure that all required fields are filled. There may be required fields that do not correspond to any data window in the legacy system. In such a case, assigning a fixed value or defining the field as optional for data transfer may be appropriate.
- Get acquainted with the terminology of the relevant data object.
 - E.g. XD01: Create customer master (see also the F1 help)

4.4 Which import technique will be used?

- Check the interfaces provided by the application. Is there a batch input program and an IDoc (for example)? You might have a look at the program library in the DX-Workbench at this point. Which method should be used in your project?
- In case of very small data quantities, it may be easier to carry out the transfer manually.
- With very large data volumes, however, batch input technology may lead to excessively long runtimes. Rough estimate for the required time: 10000 records per hour; this value, however, may vary strongly depending on the hardware.
- Batch input sessions are 'easy' in postprocessing
- Is a recording needed? Decide whether you want to use an existing import program (batch input, direct input, BAPIs, IDocs) or a recording:
 - Advantages of standard migration objects:
 - Includes screen sequences that may vary (e.g. with different material types)
 - Advantages of recordings:
 - Smaller number of target fields
 - Available for almost every transaction

4.5 Create the recording

- If you use a recording: Record the transaction and process the recording.
 - Specify
 - Field names
 - Field description
 - Default values

- If necessary:
 - Create the recording via transaction SHDB
 - Generate the program
 - Adopt the program to your needs
 - Register the program in DX-Workbench

4.6 Fill object attributes

Fill the object attributes in LSM Workbench according to the import technique chosen.

4.7 Determine the source structures

- Determine the source structures and fields
 - Is the export done into multiple files or into one sequential file?
 - Note: LSMW or DX-Workbench do not do exports from legacy systems
 - Define the **record structures** of the legacy data and introduce them to the SAP system.
- Case 1: Data is available in one or more files.
 - Introduce these structures to the SAP system.
 - Case 2: Data (still) resides in the legacy system and the legacy system provides a function for exporting the data.
 - Introduce this (these) record structure(s) to the SAP system.
 - Case 3: Data (still) resides in the legacy system and the legacy system **does not** provide a function for exporting the data..
 - Define the record structure of the data you need.
 - Export this data by means of a program to be written in the legacy system.
 - Introduce this (these) record structure(s) to the SAP system.

4.8 Create test data in DX-Workbench

Via 'Goto -> Analyze files and data structures' you get to a transaction where an example import file can be created to test the import. This file can be filled manually for test purposes. For most of the business objects you have the possibility to create a test file out of data already posted in the SAP system.

4.9 Mapping on paper

- Develop a mapping plan in written form: Assign the legacy system fields to the SAP fields. A printed object overview from LSMW might help at this point.
- Determine the form (e.g. via „MOVE“ or assigned according to a rule) in which the legacy system data shall be transferred to the SAP System.
- If applicable, define the allocation rules (LSM-internal name: „translation rules“).

4.10 Maintain structure relations

Maintain structure relations in LSM Workbench.

4.11 Data export

- In which way will the data be extracted from the non-SAP system? Note: The LSMW does not extract data.
- If only a part of your legacy system will be replaced by SAP, determine which function will be provided by the SAP system and which by the legacy system. If applicable, set up a concept of data flows and interface architecture.

4.12 Enter the rules and test out of LSMW

Enter the rules and test steps reading and converting out of LSMW

- **Read data** – automatically by pushing a button
- **Convert data**
 - The left column of the translation table is filled automatically, if this was set accordingly in translation control.
 - A sequential file is created.
- Maintain the **reusable rules**:
 - Maintain the translation tables (F4 help for right-hand column).
 - Specify your fixed values.
- Maintain the translation tables and generate a new conversion. Please note: at this point the processing steps are not sequential.

4.13 Create project, subproject and run in DX-Workbench

Create project, subproject and run definition in the DX-Workbench und define the tasks, for example:

1. task: mapping and converting
2. task: import

For task mapping and converting the migration object created in LSMW can be called; the import is done with the output file of LSMW (xxx.lsmw.conv)

4.14 Import the data

Start the run in DX workbench

Depending on the object type:

- Batch input / recording:
 - Batch input session is generated.
 - Run batch input session via SM35.
- Direct input
 - Direct input session is started.
- IDocs / BAPI:
 - Converted data is transferred to IDoc inbound processing.
 - Check inbound processing!!

5 Data Migration – Step by Step

If you want to create or change objects, make sure that you are working in change mode. To activate this mode, click *Change* in the corresponding processing step. Only this mode provides all functions required for changing.

5.1 Maintain Object Attributes

In this step, object type and import technique are selected.

Object attributes Edit Goto System Help

LSM Workbench: Change object attributes

Display <-> Change Documentation Display available interfaces

Attributes

Object: CUSTOMER Migration object customer

Owner: OHLIGER OHLIGER

Data transfer: ☒ once ☐ periodic

File names: ☐ system dependent

Object type and import technique

☒ Standard Batch/Direct Input

Object: 0050 Customer master

Method: 0000

Program name: RFBIDE00

Program type: B Batch input

☐ Batch Input Recording

Recording:

☐ Business Object Method (BAPL)

Business object:

Method:

Message type:

Basic type:

☐ IDoc (Intermediate Document)

Message type:

Basic type:

Enhancement:

☐ Enable structure relation for EDIDC40

Maintain Object Attributes

- Name your object. By entering data into field *Owner*, add the project to the list of all projects you created. You can display it afterwards in the initial screen under *My objects*.

- Choose whether data transfer is one-time or periodic. In the case of periodic transfer, files cannot be read from the frontend. This adds processing step *Frame program for the periodic data transfer*.
- Flag whether the file names are system dependant (this gives you the chance to later on enter file names per system id)
- Select the object type and import technique. Here, an F4 help is available for the input field. This help displays the relevant lists from which you can select the objects.
 - In the case of batch input and direct input, a documentation is available for the program under *Program name* (see symbol glasses).
 - If you want to carry out batch input recording, you can enter further recordings by clicking the arrow.^
 - Additionally you can use button 'Display available interfaces' to show the interfaces registered for a business object

Display available interfaces



Caution

If you apply import technique BAPI or IDoc, the program checks during the save operation whether a so-called partner agreement is already available for the preset partner (see section 3.9) and the selected message type. If this is not the case, the system tries to create them (see also section 5.13.3).



Note: Concerning flag "enable structure relation for EDIDC40" please have a look at chapter **Error! Reference source not found.**

5.2 Maintain Source Structures

In this step you define the structures of the object with name, description and the hierarchical relationships:

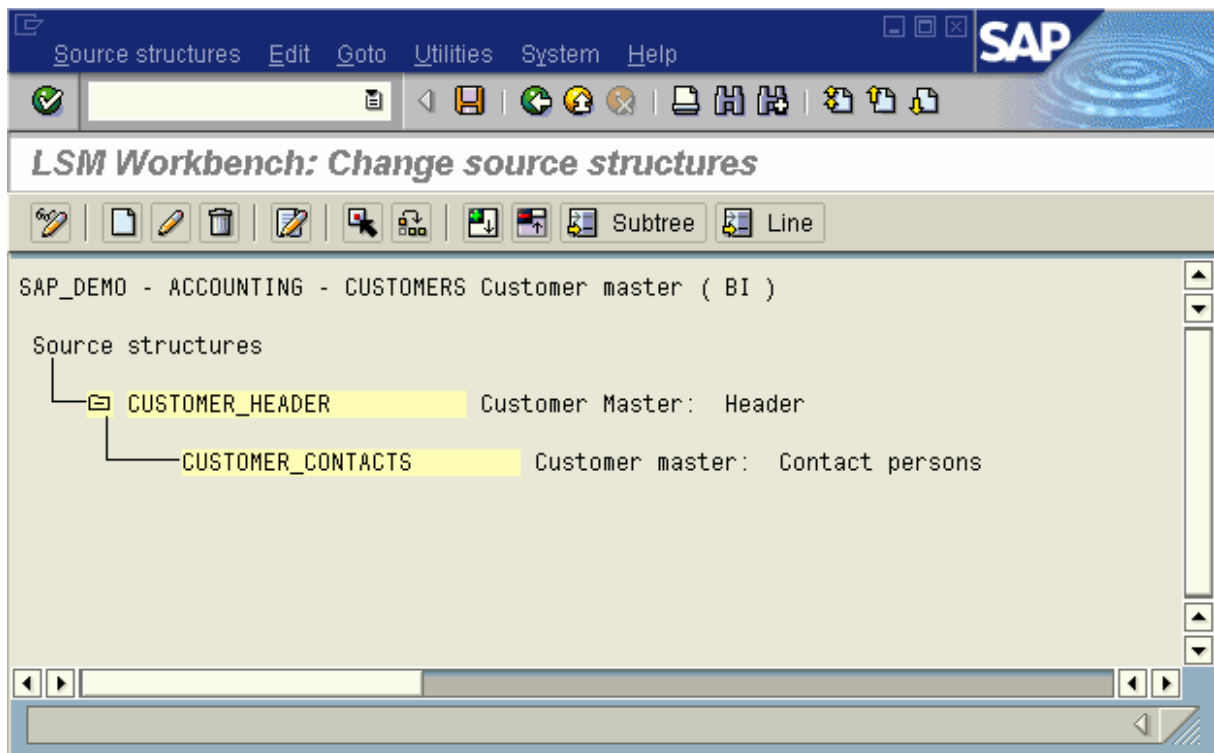
In the popup, click *Change*. You can now define, change, relink or remove structures. All these functions are available via pushbuttons.

When you define more than one structure, a popup is displayed querying the relations between the structures: same level/subordinated?



Caution

For migration objects created via transaction recording, you may only define one structure per recording here, since only one flat target structure per recording is available.



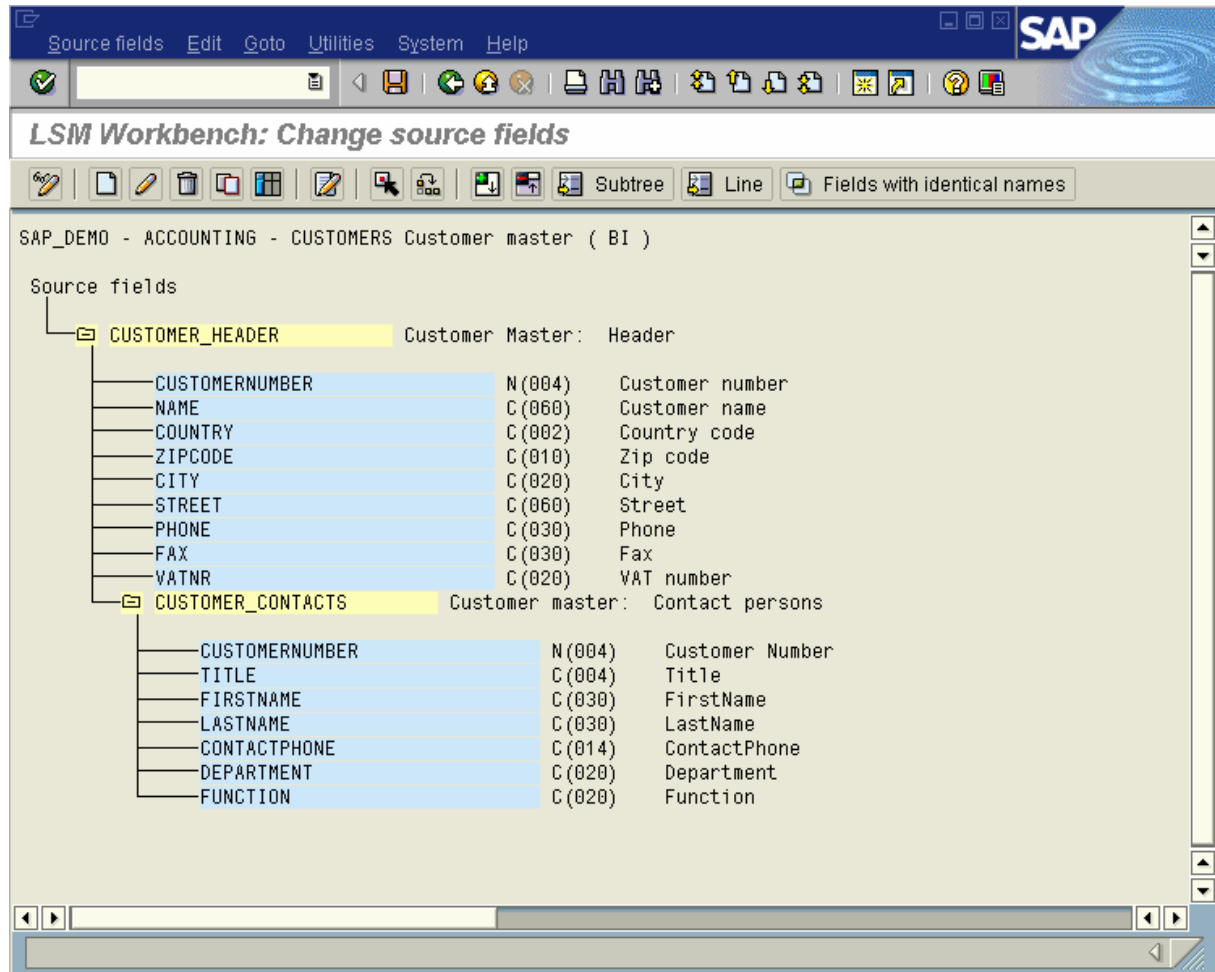
Maintain Source Structures

In the above example, one or several (or no) item records CUSTOMER_CONTACTS may exist for each header record CUSTOMER_HEADER.

Here, it is not determined yet whether these records are stored in one file or in two files.

5.3 Maintain Source Fields

In this section, fields are created and maintained for the structures defined in the preceding step.



Maintain Source Fields

There are several possibilities of defining and maintaining the source fields.

5.3.1 Create Individual Source Fields

Make sure that you are in change mode and the cursor is positioned on a source structure or an existing source field. Clicking on *Create field* displays the following popup:

Create source field

Source structure: CUSTOMER_HEADER
Customer Master: Header

Field name:

Field description:

Field length: 10

Field type: C

Identifying field value:

☐ Selection parameter for "Read data/convert data"

✓ ✗

You can select the field type from an underlying list with field type categories and the corresponding field description:

(2) 20 Entries found

| Type | Short text |
|------|---|
| C | Field with alphanumeric contents (character field) |
| N | Field with numerical contents (0-9) |
| X | Field with hexadecimal contents |
| DDMY | Date in format day-month-year (with/without delimiter) |
| DMDY | Date in format month-day-year (with/without delimiter) |
| DYMD | Date in format year-month-day (with/without delimiter) |
| AMT1 | Amount field of the form 1234,56 or -1234,56 or 1234,56- |
| AMT2 | Amount field of the form 1.234,56 or -1.234,56 or 1.234,56- |
| AMT3 | Amount field of the form 1234.56 or -1234.56 or 1234.56- |
| AMT4 | Amount field of the form 1,234.56 or -1,234.56 or 1,234.56- |
| PAC0 | Field with packed contents without decimal places |
| PAC1 | Field with packed contents with 1 decimal place |
| PAC2 | Field with packed contents with 2 decimal places |
| PAC3 | Field with packed contents with 3 decimal places |
| PAC4 | Field with packed contents with 4 decimal places |
| PAC5 | Field with packed contents with 5 decimal places |
| PAC6 | Field with packed contents with 6 decimal places |
| PAC7 | Field with packed contents with 7 decimal places |
| PAC8 | Field with packed contents with 8 decimal places |
| PAC9 | Field with packed contents with 9 decimal places |

Source Fields: Possible Field Types

During data read, you can specify whether date values are converted into the internal date format (YYYYMMDD) and amount fields are converted into the calculation format (1234.56, i.e. no triad separators, decimal point).

If data for several structures is stored in **one file** the field **Identifying field value** has to be maintained.

Please maintain only one identifying field value per structure !

For fields of top hierarchy level structures, ID "selection parameter" can be set during *Read/Convert data*. If you set this indicator, the corresponding field is made available as selection parameter when reading or converting data. As a rule, this is used for testing.

5.3.2 Maintain Source Fields in Table Form

Make sure that you are in change mode and the cursor is positioned on a source structure or an existing source field. Clicking on *Table Maintenance* displays the following screen:

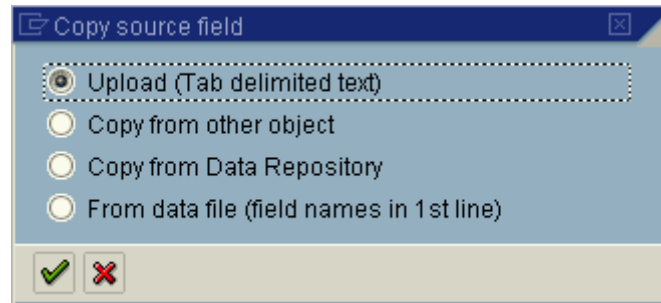
[illegible]

Maintain Source Fields in Table Form

When you enter a field name and press *Enter*, the following values are proposed:

- Field type 'C'
- Field length 10
- Field text = field name

5.3.3 Copy Source Fields from Other Sources



Make sure that you are in change mode and the cursor is positioned on a source structure or an existing source field. Selecting *Copy Source Fields* displays the following popup:

Copy Source Fields: Selecting the Source

Upload (text separated by tabs):

It is assumed that the source field description is stored in a text file the columns of which are separated by tabs, e.g.:

| | A | B | C | D | E | F | G |
|----|----------------|-----------------|---|----|---|---|---|
| 1 | CUSTOMERNUMBER | Customer number | N | 4 | | | |
| 2 | NAME | Customer name | C | 60 | | | |
| 3 | COUNTRY | Country code | C | 2 | | | |
| 4 | ZIPCODE | Zip code | C | 10 | | | |
| 5 | CITY | City | C | 20 | | | |
| 6 | STREET | Street | C | 60 | | | |
| 7 | PHONE | Phone | C | 30 | | | |
| 8 | FAX | Fax | C | 30 | | | |
| 9 | VATNR | VAT number | C | 20 | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |

Field Description

Copy from another object:

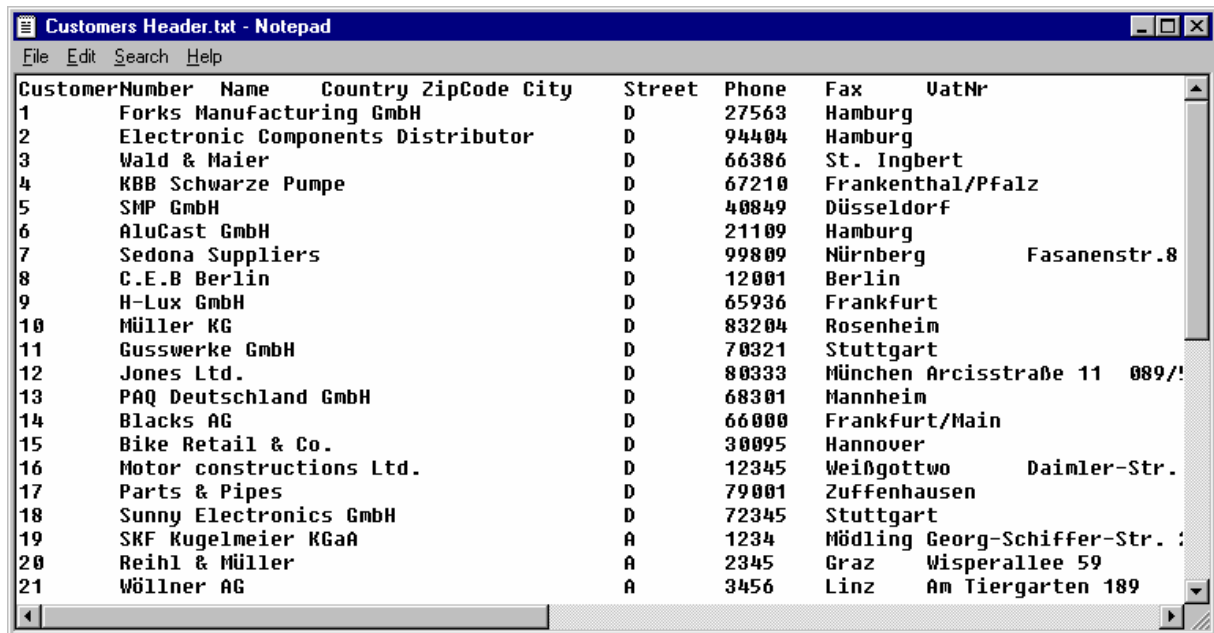
Source fields may be copied from the source structure of another object.

Copy from data repository:

Source fields may be copied from a structure of the SAP Data Dictionary.

From data file (field names in 1st line)

Source fields may be copied from a data file. This file must be stored on the PC in the form of "text separated by tabs" and contain the field names in the first line.



| CustomerNumber | Name | Country | ZipCode | City | Street | Phone | Fax | VatNr |
|----------------|-----------------------------------|---------|---------|------------------------|-----------------------|-------|-----|-------|
| 1 | Forks Manufacturing GmbH | D | 27563 | Hamburg | | | | |
| 2 | Electronic Components Distributor | D | 94404 | Hamburg | | | | |
| 3 | Wald & Maier | D | 66386 | St. Ingbert | | | | |
| 4 | KBB Schwarze Pumpe | D | 67210 | Frankenthal/Pfalz | | | | |
| 5 | SMP GmbH | D | 40849 | Düsseldorf | | | | |
| 6 | AluCast GmbH | D | 21109 | Hamburg | | | | |
| 7 | Sedona Suppliers | D | 99809 | Nürnberg | Fasanenstr.8 | | | |
| 8 | C.E.B Berlin | D | 12001 | Berlin | | | | |
| 9 | H-Lux GmbH | D | 65936 | Frankfurt | | | | |
| 10 | Müller KG | D | 83204 | Rosenheim | | | | |
| 11 | Gusswerke GmbH | D | 70321 | Stuttgart | | | | |
| 12 | Jones Ltd. | D | 80333 | München Arcisstraße 11 | 089/! | | | |
| 13 | PAQ Deutschland GmbH | D | 68301 | Mannheim | | | | |
| 14 | Blacks AG | D | 66000 | Frankfurt/Main | | | | |
| 15 | Bike Retail & Co. | D | 30095 | Hannover | | | | |
| 16 | Motor constructions Ltd. | D | 12345 | Weißgottwo | Daimler-Str. | | | |
| 17 | Parts & Pipes | D | 79001 | Zuffenhausen | | | | |
| 18 | Sunny Electronics GmbH | D | 72345 | Stuttgart | | | | |
| 19 | SKF Kugelmeier KGaA | A | 1234 | Mödling | Georg-Schiffer-Str. : | | | |
| 20 | Reihl & Müller | A | 2345 | Graz | Wisperallee 59 | | | |
| 21 | Wöllner AG | A | 3456 | Linz | Am Tiergarten 189 | | | |

Example of a data file from which the source fields are to be copied from

5.4 Maintain Structural Relationships

The structural relationships define the relationships between source and target structures.

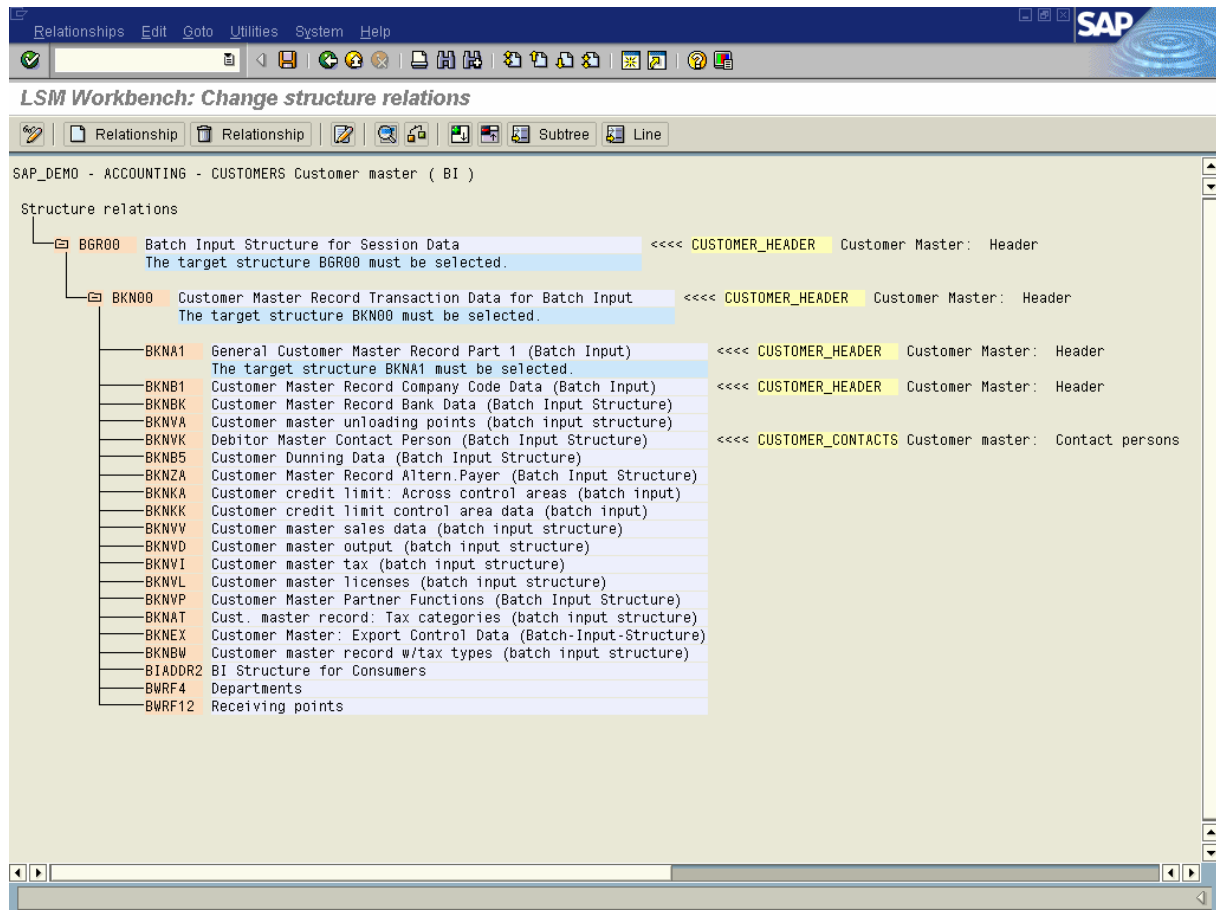
The possible target structures are defined during the selection of the object type and the import technique.

In general, there are target structures that must be selected (required segments). In this case the following note is displayed: "This structure must be selected".

To define structural relationships, position the cursor on a field of the SAP structures / target structures. Clicking *Relationship* opens a window that displays the existing source structures for selection.

If you want to change the relation, remove the existing relation first. To do this, a pushbutton is available as well.

In addition, you can use *Check* to check the structural relationships for errors. The status bar then displays an error message or message: "The structural relationships do not contain any errors".



Maintain Structural Relationships

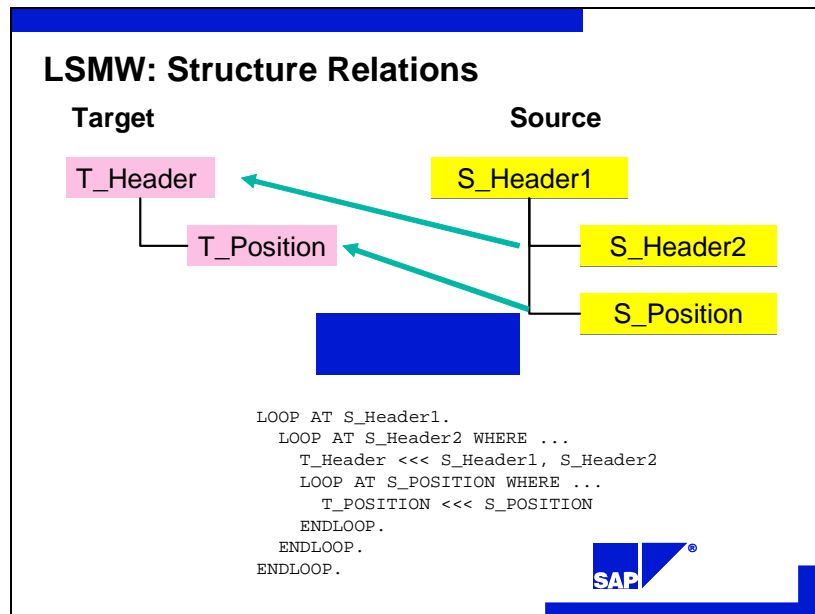
In the above example, the fields of SAP structures BGR00, BKN00, BKNA1, and BKNB1 are filled by the fields from CUSTOMER_HEADER, the fields of SAP structure BKNVK are filled by the fields from CUSTOMER_CONTACTS.



Note 1: Many Batch Input and Direct Input programs use a control record named BGR00 or BI000. You should always assign the top level source structure („header structure“) to this control record.



Note 2: It might be necessary to assign two or more source structures to one target structure. In this case you should proceed as follows: Create the source structures in the usual way. Then assign the subordinate source structure to the target structure. Thus, the fields of both source structures will be available for the fields of the target structure.



Structure relations: Example

5.5 Maintain Field Mapping and Conversion Rules

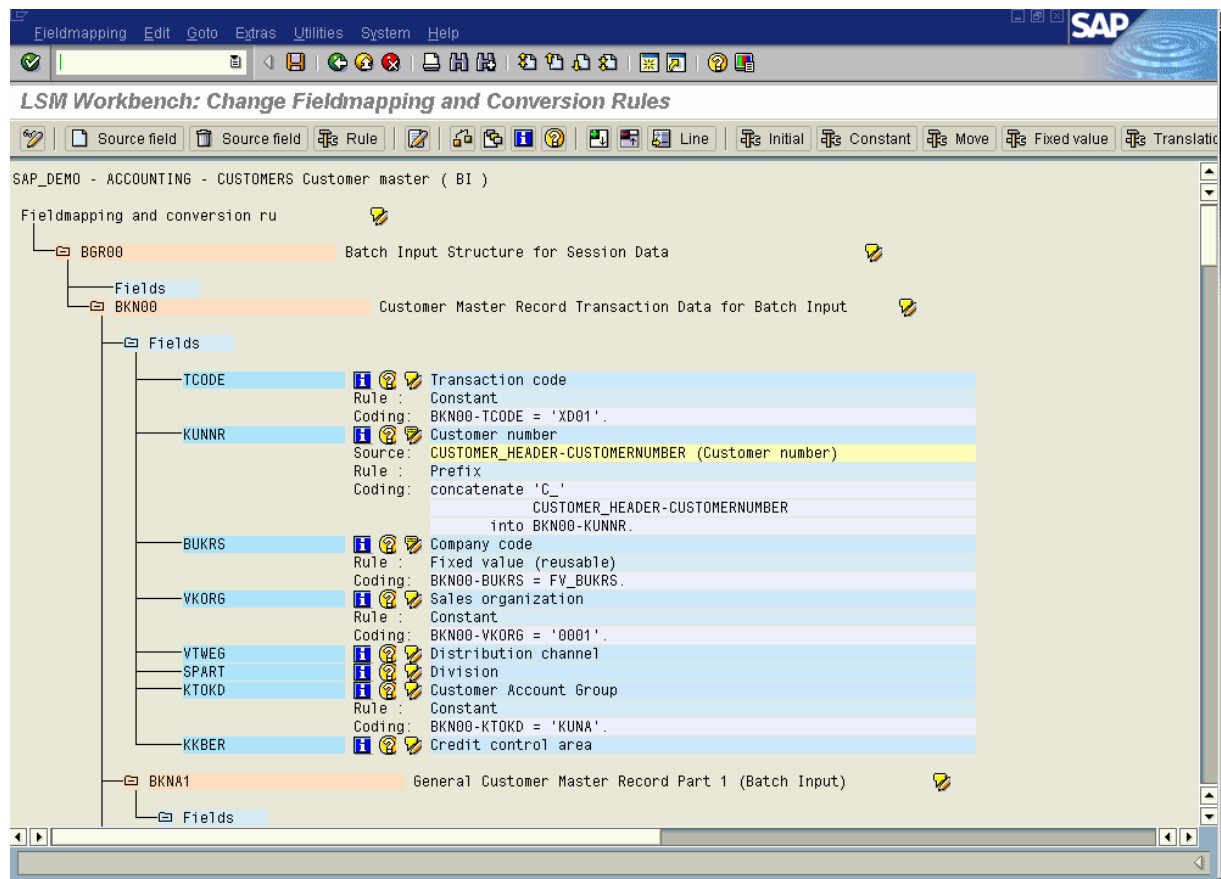
In this step, you assign source fields to target fields and define how the field contents will be converted.

All fields of all target structures, which you selected in the previous step, will be displayed. For each target field the following information is displayed:

- Field description
- Assigned source fields (if any)
- Rule type (fixed value, translation etc.)
- Coding.





Note: Some fields are preset by the system. These fields are called „technical fields“ are marked with „Default setting“. The coding for these fields is not displayed when first entering the fieldmapping; it can be displayed via the display variant (see 5.5.1). Changing the default setting may seriously affect the flow of the data conversion. If you erroneously changed the default setting, you can restore it by choosing *Extras* → *Restore default*.



Field Mapping: Tree of Target Fields for the Target Structures Selected

The following functions are available:

Field documentation : Displays a short documentation for the target field the cursor is positioned on. The documentation may branch off to further information.

Possible values : Displays a selection list of all values possible for this target field.

Longtext / Documentation : Maintenance of the documentation for a field etc.

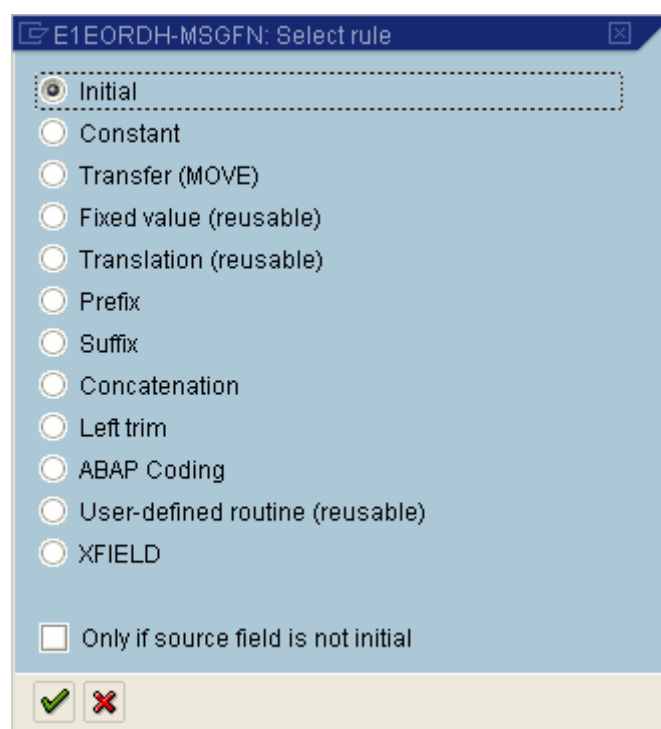
Assign a source field: To assign a source field, position the cursor on a target field in the tree structure and select *Assign source field*. This displays a list of all available source fields for selection. You can assign the fields by double-clicking on them.



Note: If you choose *Extras* → *Auto-Fieldmapping*, LSMW will give suggestions for assigning source fields to target fields.

Remove the assignment of a source field: To remove a source field assigned before, position the cursor on a target field in the tree structure and select *Remove source field*. If one source field has been assigned only, this field is removed. If several source fields have been assigned, a list of all source fields assigned is displayed for selection. The corresponding source field can then be selected by double-clicking on it.

After assigning the source fields, you define the **conversion rules**. The default rule is "Move". However, you can select various standard techniques via pushbutton.



Conversion Rules: Select Rule

Assign rules:

Initial: This deletes the coding assigned to the target field. In addition, source fields assigned to the target fields are removed as well. Depending on the object type, the target field is assigned the following value:

- For standard batch input/standard direct input: *Nodata* characters (determined e.g. in session header BGR00, BI000)
- For batch input recording: '/' as nodata character
- For BAPIs, IDocs: Clear field (i.e.: character field → blank; numeric field → '00...0')

Move: The data is transferred using ABAP command "Move". For source fields that are not of type 'C' or 'N', this means:

| | | |
|---------------------|--|---------------------------------|
| Packed field | Unpack to target field | WRITE...TO... |
| Date field | Popup to select - internal format - user format - ... | e.g. 01.10.1998 YYYYMMDD |
| Amount field | Batch input/direct input: The amount value is edited according to the format settings in the user master. BAPIs, IDocs: The amount value keeps the internal calculation format. | |

Constant: The target field is assigned a fixed value.

Fixed value (reusable): A "fixed value object" (variable) named FV_<fixedvalue> is assigned to the target field. This fixed value object is filled with an actual value in step "Maintain fixed values, translations and user-written routines".

Translation (reusable): The target field is assigned coding carrying out field contents conversion using a translation table. The values of this translation table can be entered in step "Maintain fixed values, translations and user-written routines" see 5.6.

User-written routine (reusable): The system creates the frame of a form routine (ABAP subroutine) with name prefix "ur_". This routine can be reused, i.e. it can also be used in other objects of the project.

With all kinds of reusable rules, the LSM Workbench proposes one to three possible names. One name is recommended by the system. SAP recommends you to use the proposed name. For details regarding naming conventions, see 5.5.4.

When creating user written routines please keep in mind that:

- the correct amount of source fields has been linked (regarding the amount of input parameters)
- the source fields are related in correct sequence (i.e. the sequence of the parameters).

Prefix: Specify any prefix to precede the contents of the source field.

Suffix: Specify any suffix follow the contents of the source field.

Concatenation: You can concatenate two or more source fields.

Transfer left-justified: Transfers the contents of the source field in left-justified form.

ABAP coding: Double-clicking on a target field branches off to the ABAP editor. There you can edit generated ABAP coding or write and save your own coding. A large part of the usual standard SAP editor functions, such as *Check* (syntax check), *Pretty Printer*, etc., are available there.

Under *Insert* you can add the following to your coding:

- **source fields:** all source fields available are displayed for selection
- **global variable:** see 5.7
- **global functions:** see 5.5.3

XFIELD: This is a special function for processing of IDocs. In some cases an 'X-structure' exists in addition to the data transfer structure (where the values for the import can be found); the fields of this 'X-structure' have to be filled with 'X' or blank to decide if the corresponding field in the data transfer structure should be transferred or not.

The following coding is generated automatically:

If not <field in the data transfer structure> is initial.

<field in X-structure> = 'X'.

else.

<field in X-structure> = ' '.

Endif.

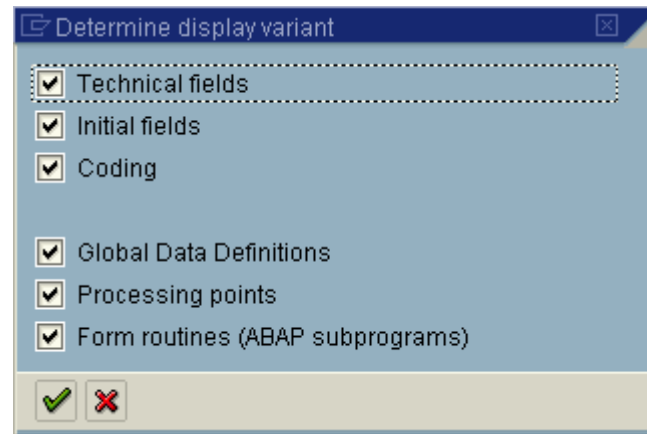


Note: Via *Extras* -> *fill X-structures* the coding for whole structures can be added

5.5.1 For the Advanced User: Display Variant, Processing Times

Define display variant: In work step "Maintain field mapping and conversion rules", select '* display variant'. This displays popup *Define display variant*. This function is useful mainly to advanced users who want to modify their field mapping.

You can specify the information to be displayed.



Define Display Variant

t

Global data definitions: Displays label __GLOBAL_DATA for global data definitions and declarations. There, you can define a variable, structures, tables, etc., to be used in the field mapping of your own coding.

Processing times: Here you can insert your own coding at specific processing times.

The following processing times are available:

| Processing time | Meaning | Default setting |
|--------------------------|---|--|
| __BEGIN_OF_PROCESSING__ | Before the beginning of data processing | (blank) |
| __BEGIN_OF_TRANSACTION__ | Before the beginning of transaction data processing | (blank) |
| __BEGIN_OF_RECORD__ | Before applying the conversion rules for a source structure | Initialize the structure <segment> (Name of target structure) Batch Input, Direct Input: <segment> = init_<segment>. BAPI, IDoc: g_edidd_segnam = '...'. g_edidd_segnum = '....'. g_edidd_psgnum = '.....'. g_edidd_hlevel = '...'. Clear <segment>. |
| __END_OF_RECORD__ | After applying the conversion rules for a source structure | Transfer_record. |
| __END_OF_TRANSACTION__ | After finishing transaction processing | Transfer_transaction. |
| __END_OF_PROCESSING__ | After finishing data processing | (blank) |

Form routines: Displays label __FORM_ROUTINES__ for form routines (ABAP subroutines). There, you can define ABAP subroutines to be used in your own coding for field mapping.

Technical fields: Displays the so-called technical fields. These are target fields for which LSMW proposes a conversion rule (e.g. constant). As a rule, modifications need not be made.

Initial fields: Displays initial fields.

Coding: Displays the stored coding.



Note: Under menu item *Extras* → *Source fields not assigned* you can display the source fields not yet assigned, i.e. you can see whether there is data which has not yet been adequately dealt with.

5.5.2 For the Advanced User: Global Variables

The LSM Workbench internally uses a number of global variables.

1. From the list of work steps, select *Field mapping and conversion rule*.
2. Branch off to the coding by double-clicking on a target field
3. Select *Insert* → *Global variable*.

This variable can be used in your ABAP coding.

| Global variable | Description |
|---|---|
| <code>g_project</code> | Current project |
| <code>g_subproj</code> | Current subproject |
| <code>g_object</code> | Current object |
| <code>g_record</code> | Current target structure |
| <code>g_cnt_records_read</code> | Number of records read |
| <code>g_cnt_records_skipped</code> | Number of records skipped |
| <code>g_cnt_records_transferred</code> | Number of records transferred to a file |
| <code>g_cnt_transactions_read</code> | Number of transactions read |
| <code>g_cnt_transactions_skipped</code> | Number of transactions skipped |
| <code>g_cnt_transactions_transferred</code> | Number of transactions transferred to a file |
| <code>g_cnt_transactions_group</code> | Number of transactions in the current batch input session |
| <code>g_userid</code> | User ID |
| <code>g_groupname</code> | Name of the batch input session |
| <code>g_groupnrx</code> | Current number of the current batch input session |

5.5.3 For the Advanced User: Global Functions

The LSM Workbench provides a series of functions that can be used in any position of the ABAP coding.



Note: These functions allow to partially considerably influence the flow of the data conversion program. Please do apply these functions with care.

1. From the list of work steps, select *Field mapping and conversion rule*.
2. Branch off to the coding by double-clicking on a target field
3. Select *Insert → Global functions*.

The following functions are available:

| Global function | Description |
|---|---|
| <code>transfer_record.</code> | Transfers the current record (i.e. for the current target structure) to the output buffer. |
| <code>transfer_this_record '...'</code> | Transfers a record of another target structure to the output buffer. The name of the target structure has to be specified as argument in single quotes. |
| <code>at_first_transfer_record.</code> | Transfers the current record to the output buffer, if it is the first transaction. |
| <code>on_change_transfer_record.</code> | Transfers the current record to the output buffer, if it has changed compared to the last record. |
| <code>transfer_transaction.</code> | Writes the current transaction to an output file. All records of |

| | |
|-------------------|---|
| | the output buffer are transferred to the output file. |
| skip_record. | The current record is not transferred to the output buffer. |
| skip_transaction. | The current transaction is not written to the output file. |

5.5.4 For the Advanced User: Reusable Rules — Naming Conventions

Reusable rules are rules that are available across the project. They can be used in all objects of a project. Reusable rules are: fixed values, translations, and user-written routines.

If you assign a reusable rule to a target field, the system proposes one to three different names. To understand the naming conventions, we should look at the definition of data objects in the SAP system.

Data object definition in the SAP system is performed on three levels:

Domain: On the "lowest" level, technical attributes are defined, e.g. field type, field length, value table or fixed values.

Data element: On the "second" level, "semantic" characteristics are defined on the basis of a domain and its characteristics, e.g. language-dependent texts, documentation.

Field: On top level, attributes of the field in the context of a structure or table are defined on the basis of a data element, e.g. foreign key relations, search helps.

This means in particular: For a domain, there normally are several data elements which refer to the domain.

(A count in the R/3 system, Release 4.5A produces the following figures: Domains: about 22,000, data elements: about 117,000, fields: about 1,028,000)

SAP recommends to accept the names defaulted by the system as a rule. An exception is given, if the domain is very general such as "CHAR1" (about 5,200 data elements) or "XFELD" (about 13,500 data elements). If you used the name of the domain in this case, the reusable rule might not be usable for another field, since this field may have a completely different meaning.

This naming procedure keeps the number of conversion rules small and maintains the consistency in data conversion.

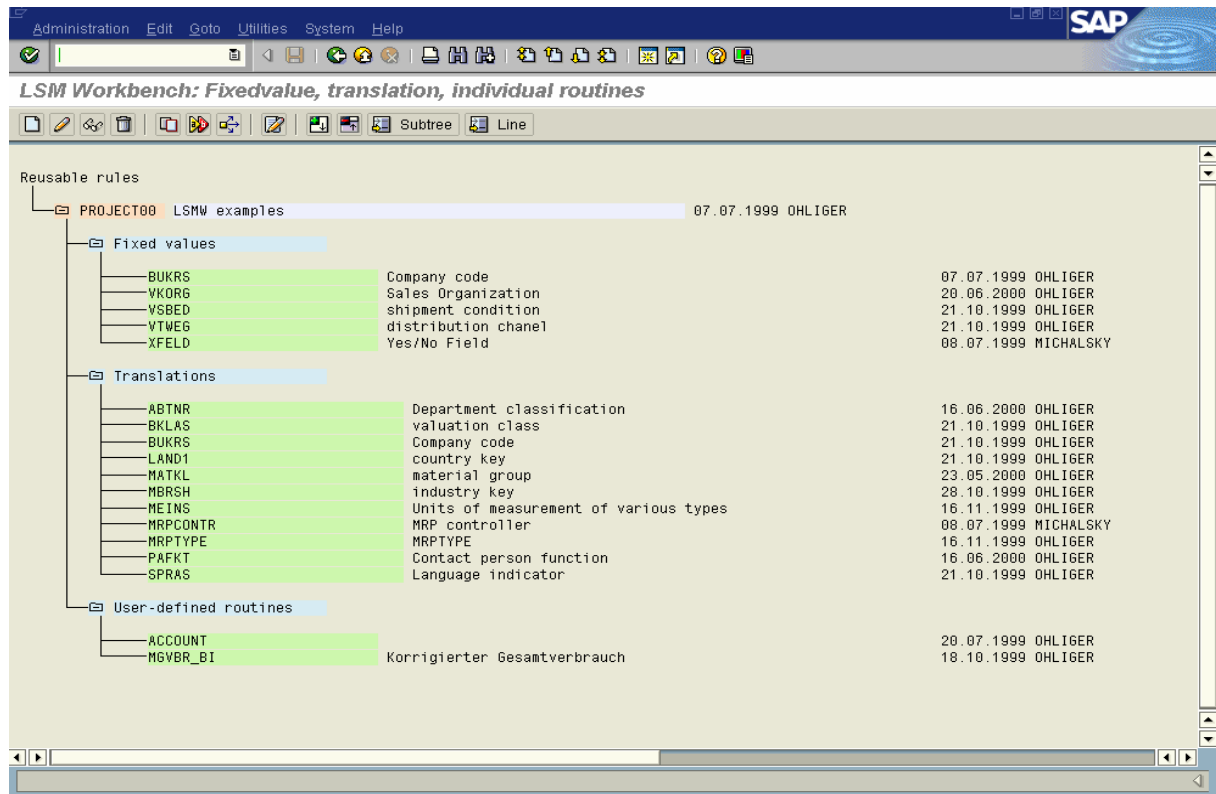
Example:

| No. | Field | Data element | Domain | Name |
|-----|---------|--------------|--------|--------------|
| 1 | BUKRS | BUKRS | BUKRS | Company code |
| 2 | CO_CODE | CO_CODE | BUKRS | Company code |

Both fields are named "Company code". The field names are different, the domain is the same. Thus both fields should be filled with the same fixed value or the same translation or user-written routine.

5.6 Maintain Fixed Values, Translations and User-written Routines

In this step you can process the reusable rules of a project:



Process Reusable Rules

Fixed value: Here you can specify length, type, flag for lowercase/uppercase and value in addition to the description.

| Fixed value | |
|--------------------|--------------------------|
| Project | PROJECT00 |
| Name | BUKRS |
| Description | Company code |
| Attributes | |
| Output length | 4 |
| Dictionary type | CHAR |
| Lower case letters | <input type="checkbox"/> |
| Value | 0001 |

Change Fixed Value

Translation: Here you can enter information on the source field and the target field:

The screenshot shows the 'Change translation' dialog box in SAP. The 'Translation' tab is selected, displaying the following information:

- Project:** PROJECT00
- Description:** country key

Below this, there are four tabs: 'Source field, target fields', 'Control data', '1:1 translation values', and 'Interval tra...'. The 'Source field, target fields' tab is active, showing two sections:

- Source field:**
 - Length: 2
 - Type: C
 - Lower case letters: ☒
- Target field:**
 - Output length: 3
 - Dictionary type: CHAR
 - Lower case letters: ☐

Change translation / Source field, target fields

If you are creating a new translation you have to save data before you can change to **Control information**.

Control information: Here you can define the translation type. You can specify which of the two translation tables will be searched for a value first and which alternative will be selected, if no suitable entry is found:

The screenshot shows the 'Change translation' dialog box in SAP. The 'Control data' tab is selected. The 'Translation' section shows 'Project' as 'PROJECT00' and 'Description' as 'country key'. The 'Variants' section has four columns: 'Variant', 'First Alternative', 'Second Alternative', and 'New value'. The '1:1 translation' variant is selected. The 'Add 1:1 translation values automatically' checkbox is checked.

| Variant | First Alternative | Second Alternative | New value |
|----------------------|----------------------------------|----------------------------------|-----------|
| 1:1 translation | <input checked="" type="radio"/> | <input type="radio"/> | |
| Interval translation | <input type="radio"/> | <input type="radio"/> | |
| Initial value | <input type="radio"/> | <input type="radio"/> | |
| Constant | <input type="radio"/> | <input type="radio"/> | |
| Transfer (MOVE) | <input checked="" type="radio"/> | <input type="radio"/> | |
| User-defined routine | <input type="radio"/> | <input type="radio"/> | |
| No action | <input type="radio"/> | <input checked="" type="radio"/> | |

Other
☒ Add 1:1 translation values automatically

1:1 translation values: Here you specify the value table to be used during translation. You may also upload the values from a PC file (text separated by tabs). In addition, F4 help is available in column "New value".

Important

During translation, only values for which the OK flag was set are included.

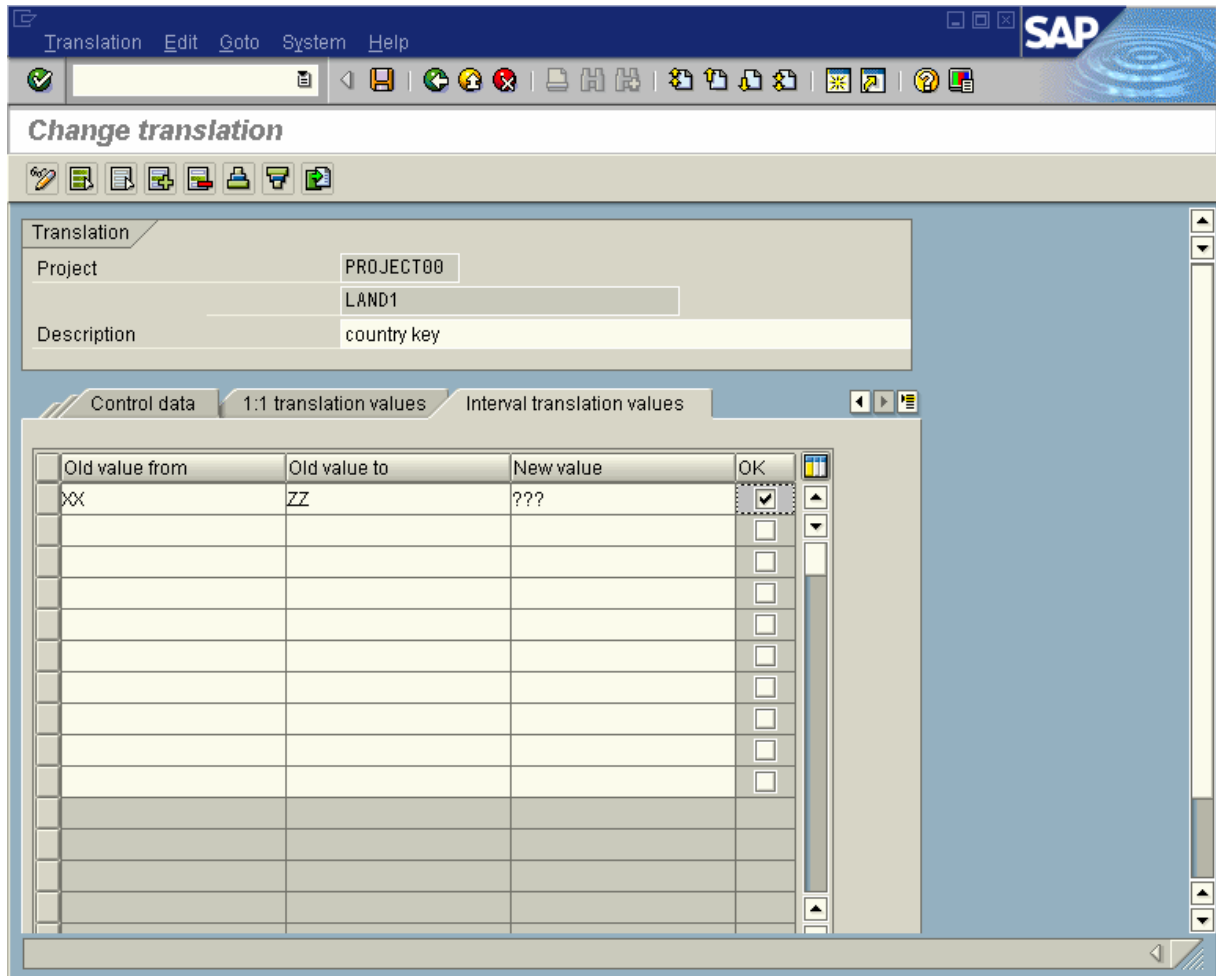
The screenshot shows the 'Change translation' dialog box in SAP, with the '1:1 translation values' tab selected. It displays a table of translation values with columns for 'Old value', 'New value', and 'OK'.

| Old value | New value | OK |
|-----------|-----------|-------------------------------------|
| B | BE | <input checked="" type="checkbox"/> |
| BR | BR | <input type="checkbox"/> |
| CA | CA | <input checked="" type="checkbox"/> |
| CH | CH | <input checked="" type="checkbox"/> |
| D | DE | <input checked="" type="checkbox"/> |
| DE | DE | <input checked="" type="checkbox"/> |
| DK | DK | <input type="checkbox"/> |
| E | ES | <input checked="" type="checkbox"/> |
| F | FR | <input checked="" type="checkbox"/> |
| FI | FI | <input type="checkbox"/> |
| FR | FR | <input checked="" type="checkbox"/> |
| GB | GB | <input checked="" type="checkbox"/> |
| I | IT | <input checked="" type="checkbox"/> |
| IR | IR | <input type="checkbox"/> |

Interval translation values: Here you specify the value table to be used during translation by intervals. You may also upload the values from a PC file (text separated by tabs). In addition, F4 help is available in column "New value".

Important

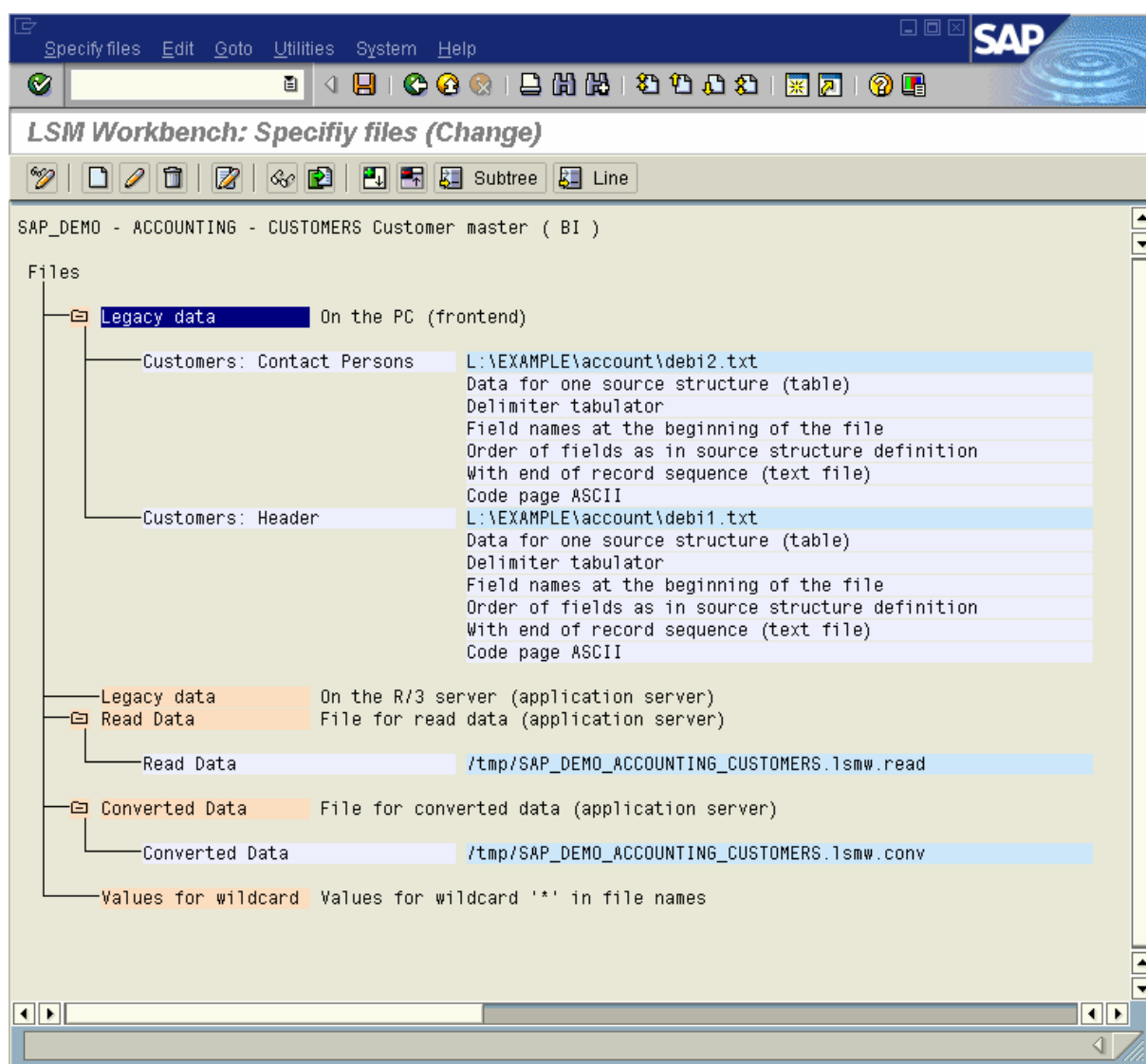
During translation, only values for which the OK flag was set are included.



5.7 Specify Files

In this step you describe all files to be used in the following steps:

- Your legacy data on the PC and/or SAP server
- The file for the read data
- The file for the converted data



Specify Files

If your legacy data is on the PC:

1. In change mode, position the cursor on the line "Legacy data — on PC (frontend)".
2. Select *Add entry*.
A popup is displayed.
3. Specify file path (F4 help), file name and description and other properties.

File on frontend: Edit properties

File: L:\EXAMPLE\account\debi2.txt

Description: Customers: Contact Persons

File contents

☒ Data for one source structure (table)

☐ Data for several source structures (seq. file)

Delimiter

☐ No delimiter ☐ Comma

☒ Tabulator ☐ Space

☐ Semicolon ☐ Other

File structure

☒ Field names at the beginning of the file

☒ Order of fields as in source structure definition

File type

☒ Record end indicator (text file)

☐ Fixed record length

☐ Hexadecimal length field (4 B.) at the beginning

Code page

☒ ASCII ☐ IBM DOS

OK Cancel

File on Frontend (PC): Properties

If your legacy data is on the SAP server:

1. In change mode, position the cursor on the line "Legacy data on SAP server (application server)".
2. Select *Add entry*.
A popup is displayed.
3. Specify file path, file name and description.
4. Under "Codepage ID", specify the indicator of the legacy system's character set.
5. Determine the technical record description and the separators.



Note: Please note that the SAP system uses user ID <sid>adm with regard to the operating system. Therefore, make sure that you have read/write authorization for the selected directory.

File on application server: Edit properties

File: tmp/debi1.txt

Description: Customer master data

File contents

☒ Data for one source structure (table)

☐ Data for several source structures (seq. file)

Delimiter

☐ No delimiter ☐ Comma

☒ Tabulator ☐ Space

☐ Semicolon ☐ Other

File structure

☒ Field names at the beginning of the file

☒ Order of fields as in source structure definition

File type

☒ Record end indicator (text file)

☐ Fixed record length

☐ Hexadecimal length field (4 B.) at the beginning

Code page

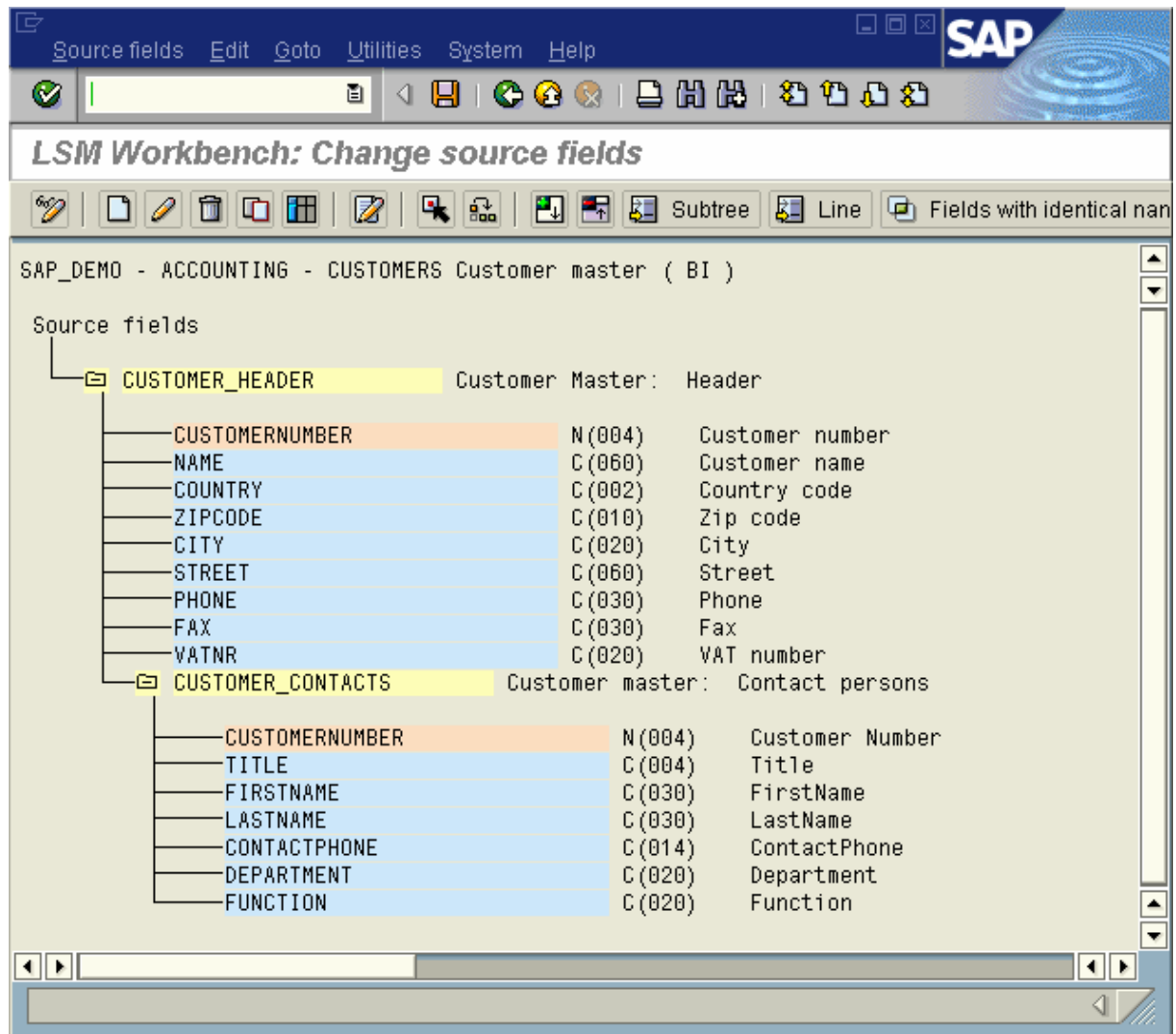
Code page: 1100 SAP-intern, wie ISO 8859-1 (00697/00819)

OK Cancel

File on SAP Server: Maintain Properties

Please consider the following notes:

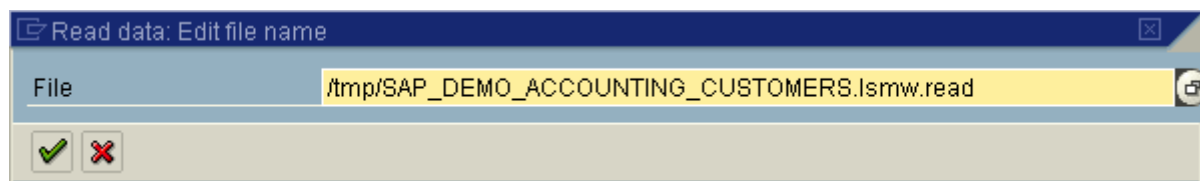
- If a file contains data for several source structures, the field sequence has to correspond to the source structure definition.
- If a file contains data for a single source structure, either the field sequence has to correspond to the source structure definition or field names have to be specified at the beginning of the file which can be used for assigning the columns to the fields.
- If a file contains end-of-line indicators (text file), packed fields are not allowed.
- If a file contains separators, packed fields are not allowed.
- PC files and server files may be mixed at will.
- In the following step, a file containing data for several source structures can be assigned to several source structures.
- In the following step, a file containing data for a single source structure can be assigned to one single source structure only.
- If several files are used in an object, the corresponding source structures have to contain fields of the same name. In our example, this is field CUSTOMER_NUMBER:



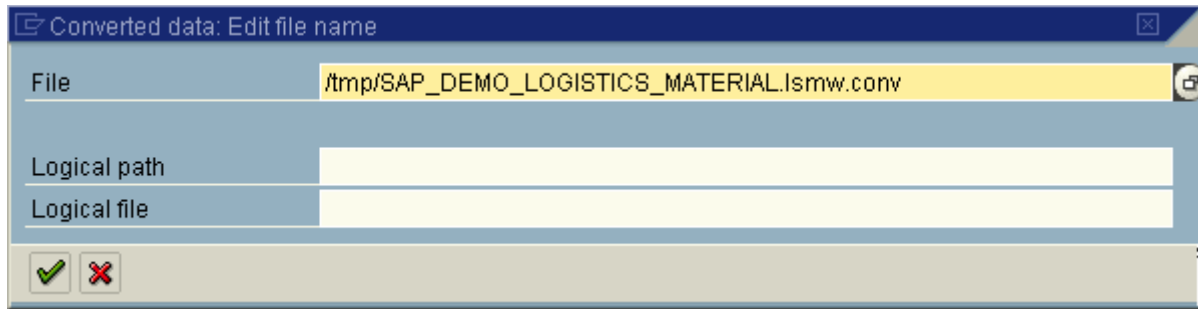
Display Merge Fields

File of read data:

Here, the file name is entered. We recommend you to use file extension ".lsmw.read" to differentiate the read data from the converted data.

**File of converted data:**

Specify the file name. We recommend you to use file extension ".lsmw.conv". Fields "Logical path" and "Logical file name" should be filled only if this is required for the subsequently called batch input or direct input program (fields only are shown in this case) For both fields, F4 help is available.



Converted data: Edit file name

| | |
|--------------|--|
| File | /tmp/SAP_DEMO_LOGISTICS_MATERIAL.lsmw.conv |
| Logical path | |
| Logical file | |

OK Cancel



Note 1: Names for paths and files can be freely assigned according to the operating system's naming conventions.



Note 2: If your files are stored in several sets of files, you can add a wildcard (*) to the name of your file. The possible values for '*' can be specified under "Values for wildcard".

5.8 Use Wildcards in File Names

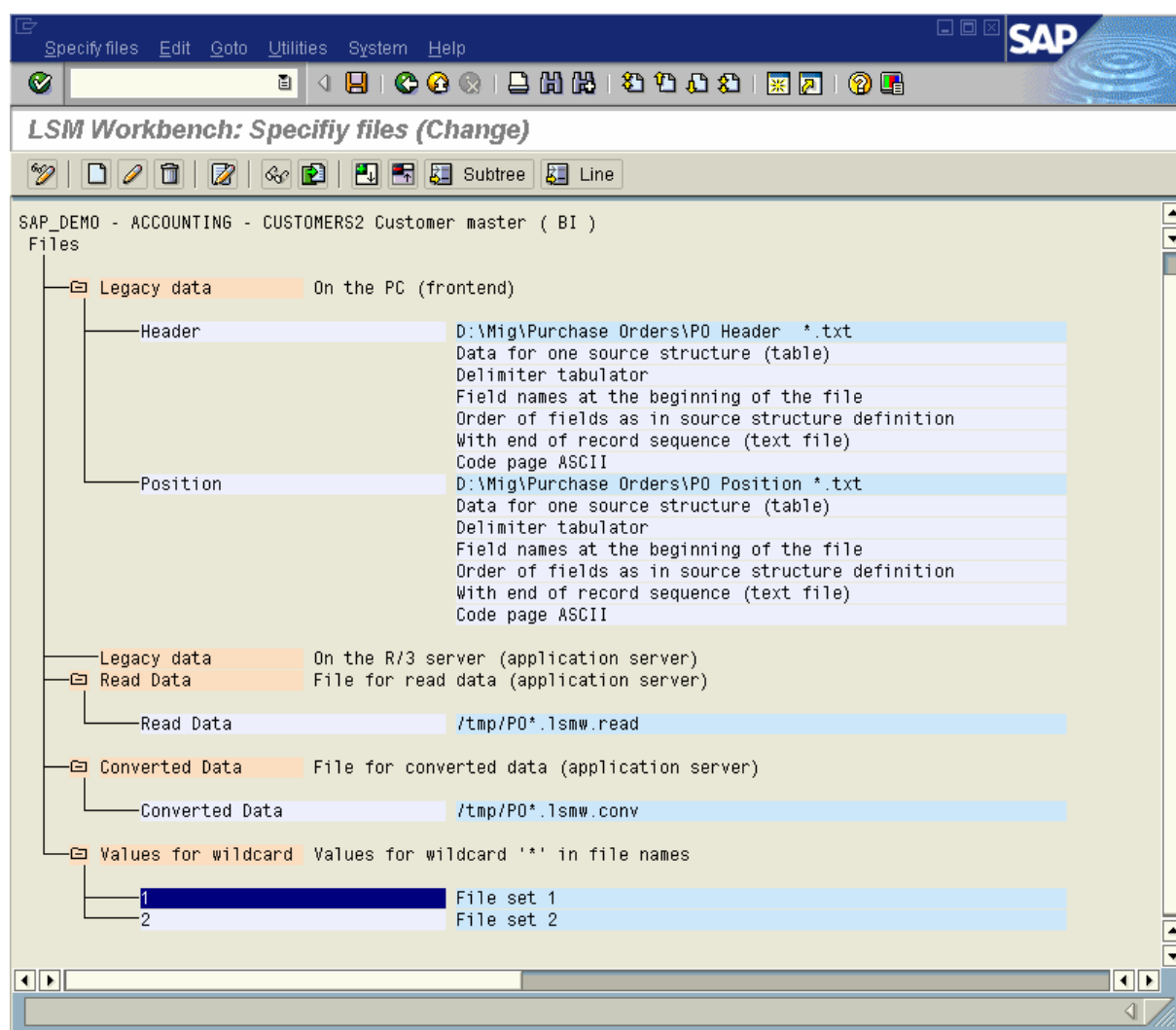
Example for the usage of wildcards in file names: Let's assume that the legacy data is stored in the following four files:

- File 1: D:\Mig\Purchase Orders\PO Header 1.txt
- File 2: D:\Mig\Purchase Orders\PO Position 1.txt
- File 3: D:\Mig\Purchase Orders\PO Header 2.txt
- File 4: D:\Mig\Purchase Orders\PO Position 2.txt

Two files each (*1.txt and *2.txt) form a "set", i.e. file 2 contains the position data for the header records in file 1, file 4 contains the position data for the header records in file 3.

When reading the data, files 1 and 2 shall be processed before files 3 and 4.

This is achieved by means of the following settings:



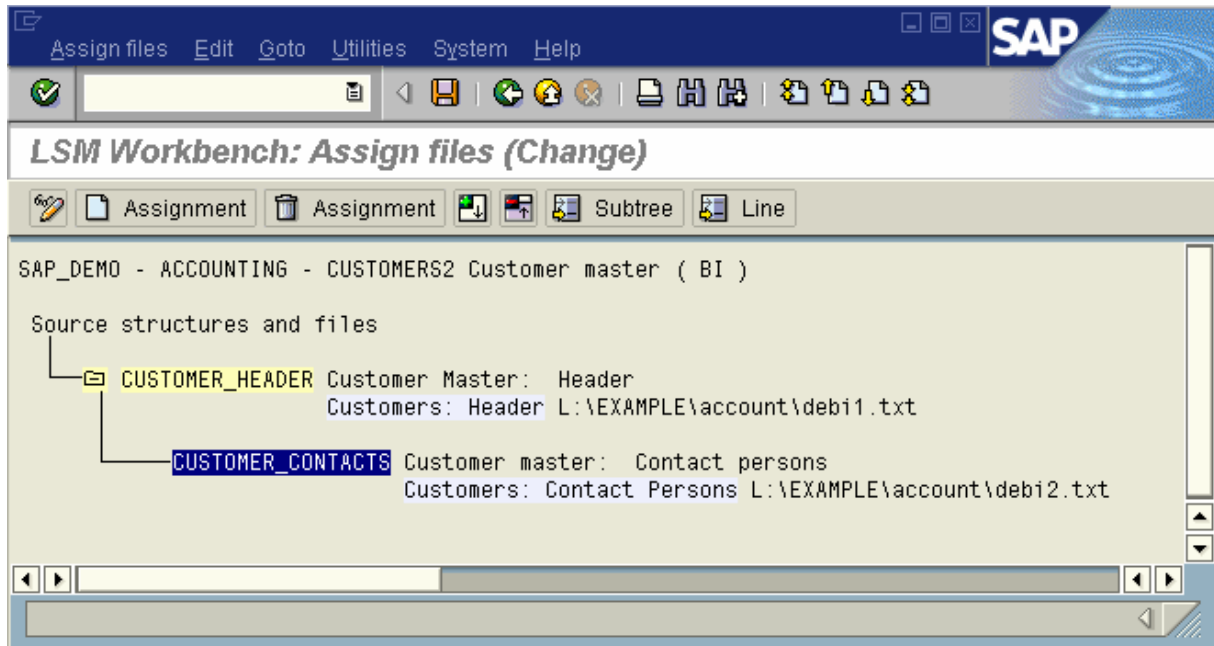
Specify Files: Using Wildcards



Note: You can also use a wildcard in the names of the files of read data and converted data.

5.9 Assign Files

In this step, you assign defined files to the source structures:



Assign Files



Note: If you change file names or properties subsequently, the file assignment is kept.

5.10 Read Data

Proceeding:

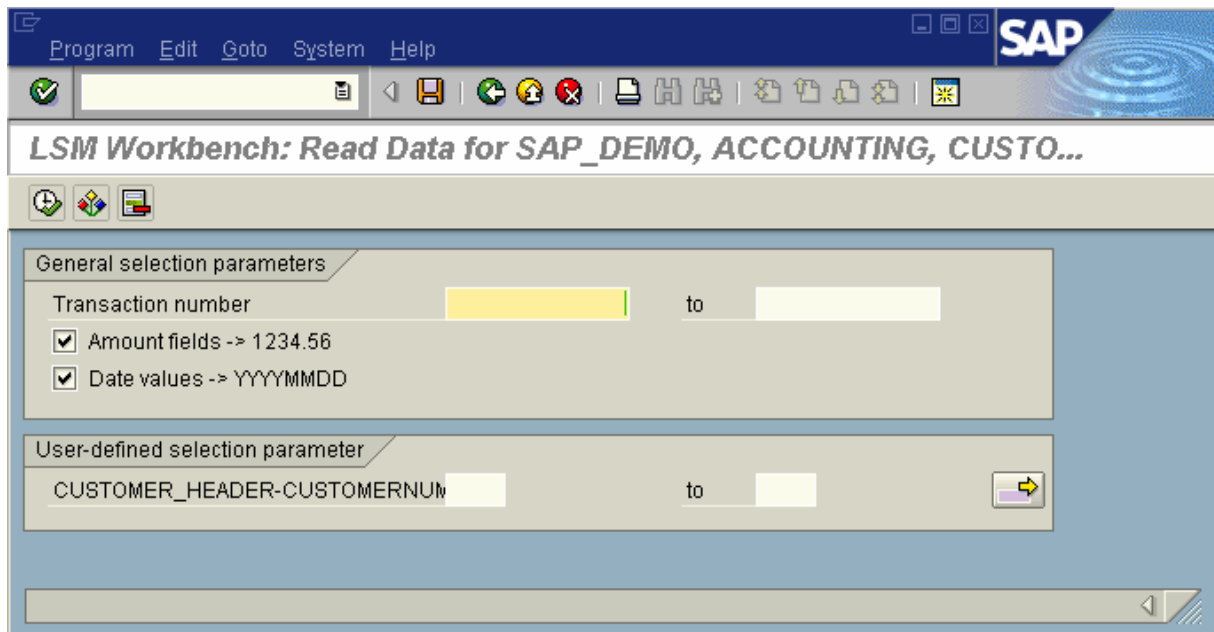
- If you want to process all data belonging to an object, click on *Execute*. The process is started.
- If you want to migrate a part of the data only, you can limit the number of data to be migrated in field "General selection parameters". Make your selection in field "Transaction number" from "... to ...". Multiple selection is possible.

If you marked one or several source fields as selection parameters when defining the source fields, these fields are also offered as selection parameters.

In addition, two check boxes are offered:

- Amount field: Amount fields are converted into calculation format (with decimal point).
- Date field: Date fields are converted into internal format (YYYYMMDD).

If you use a wildcard in the file names for the input files, and at least one value has been defined for the wildcard, a selection parameter for the wildcard is offered as well. If you do not make any entry here, all wildcard values defined are processed.



Data Read Program: With User-defined Selection Parameter



Note: First, the system checks whether the data read program is still up-to-date. If this is not the case, it is regenerated automatically.

5.10.1 Display Read Data

In this step, you can display all or a part of the read data in table form. Clicking on a line displays all information for this line in a clear way. The same happens when you click on *Field contents*.

Change display allows to select either a one-line or multi-line view.

Display color palette displays the colors for the individual hierarchy levels.

5.11 Convert Data

5.11.1 General Remarks

With regard to operation, this work step essentially corresponds to work step "Read Data" (see 5.8).

If you do not make any data selection, confirm the process by clicking on *Execute*. Otherwise, make your selection in field "Transaction number" from "...to...". Here, multiple selection of transaction numbers is possible as well.

If you marked one or several source fields as selection parameters when defining the source fields, these fields are also offered as selection parameters.

If you use a wildcard in the file names for the input files, and at least one value has been defined for the wildcard, a selection parameter for the wildcard is offered as well. If you do not make any entry here, all wildcard values defined are processed.

Data Conversion Program: With User-defined Selection Parameter



Note: First, the system checks whether the data conversion program is still up-to-date. If this is not the case, it is regenerated automatically.

5.11.2 Additional Function for BAPI/IDoc

If the LSMW object is based on a BAPI or an IDoc, further selection parameters are displayed on the data conversion program selection screen:

Convert Data: Further Selection Parameters for BAPI/IDocs

If you select *Create file*, a file is created during data conversion.

If you select "Create IDocs directly", IDocs are collected during data conversion and submitted for IDoc creation in packages. The package size can be determined using parameter "Number of IDocs per package". The default value is 50.

5.12 Display Converted Data

See section 5.10.1.

5.13 Import Data

The steps displayed by the program depend on the selected object type:

- Standard batch input or recording:
 - Generate batch input session
 - Run batch input session
- Standard direct input:
 - Start direct input session
- BAPI or IDoc:
 - Start IDoc creation
 - Start IDoc processing
 - Create IDoc overview
 - Start IDoc postprocessing

5.17.1. Import Data with Batch Input

5.13.1.1 Generate Batch Input Session

In this step, the standard batch input program belonging to the object is directly called. The name of the file with the converted data is already proposed.

The batch input sessions to be generated are named after the LSMW object.

5.13.1.2 Run Batch Input Session

The program goes to SAP standard transaction SM35. However, only the batch input sessions for the selected object are displayed.



Note: If you used the name of the object in other projects or subprojects as well, batch input sessions from these objects may also be displayed.

5.13.2 Import Data with Direct Input

5.13.2.1 Start Direct Input Session

Depending on the object type, either the standard direct input program belonging to the object is called or you can select a direct input program or a direct input transaction.

5.13.3 Import Data with BAPI or IDoc Technique

Data stored in a file by means of the IDoc technique is generally imported in two steps. You can call these steps in LSM Workbench:

- **Start IDoc creation.** First, the file of the converted data is read. The "information packages" contained are stored in the SAP database in IDoc format. It is, however, not stored in the database of the corresponding application. The system assigns a number to every IDoc. Then the file of the converted data is deleted.
- **Start IDoc processing.** The IDocs created in the first step are submitted to the corresponding application program. This application program checks the data and posts it in the application's database, if applicable.



Note: Step "**Start IDoc creation**" is not performed, if you selected option "Create IDocs directly" during data conversion.

Whether the second step is automatically initiated depends on the settings of the ALE-EDI customizing.

One essential setting is made in the so-called partner agreement (for a partner and a message type, see section 5.2). This agreement specifies whether the IDocs are to be processed immediately or by means of a background program.



Note #1: Partner agreements automatically created by the LSM Workbench are set as follows: "Initiation by background program". (You can manually change this setting at any time.)



Note #2: During the processing of inbound IDocs, so-called work items are created in the standard program. These are elements of the SAP workflow that are usually not required during data migration. For information on how – and with which consequences – the creation of work items can be suppressed see R/3 Note no. 149368.



Note #3: CD-ROM "Interface Adviser" provided by SAP contains useful information that helps to increase performance in connection with IDoc processing. Follow the path → *Technology* → *Interfaces* → *Background processing* → *Import* → *ALE/IDoc* → *Performance*.

You can do the following in addition to these two processing steps:

Create IDoc overview. This displays a „status overview which allows to display individual IDocs with the "drill-down" technique.

6 Recordings

Perform a transaction „trial run“;



Caution

This is no simulation mode! Your input is posted in the system!

Postprocess recording: Assign field names, field texts and default values

Save recording: This generates the above structure in the Data Repository.

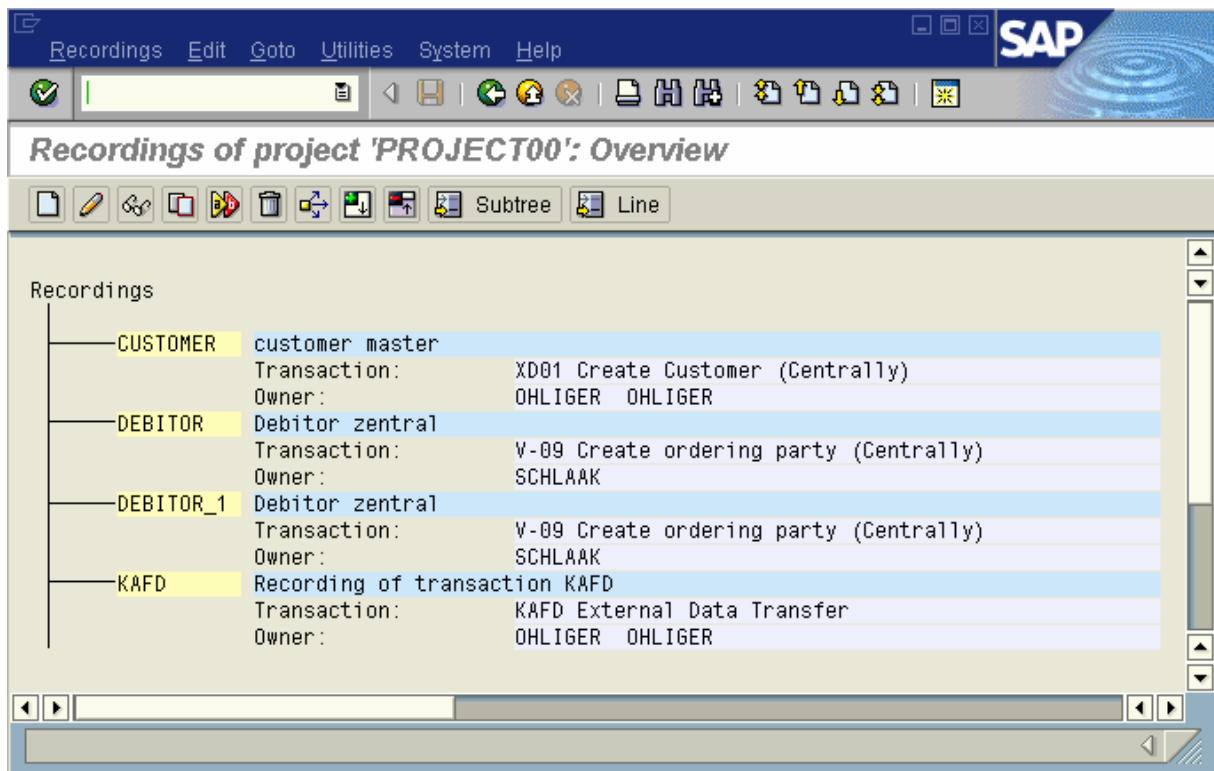


Caution

In *Attributes for an object* you can assign any number of recordings to an object. This way you can run various transactions in succession for one data record.

6.1 Detailed Description of the Process

In the initial screen you select function *Recordings* under *Goto*.

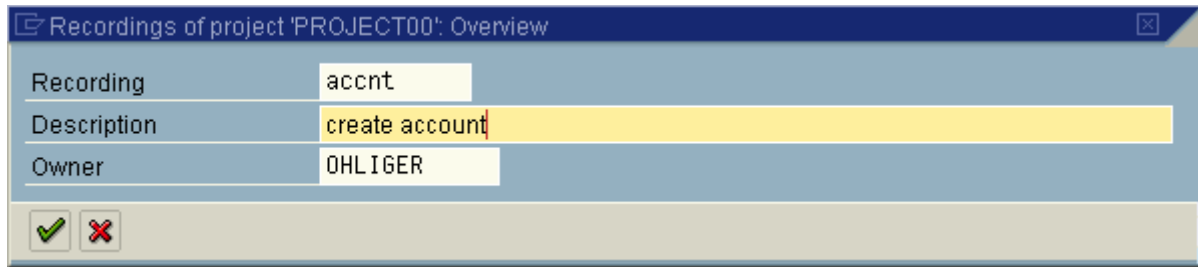


Recordings: Overview



Note: Recordings are assigned to exactly one project.

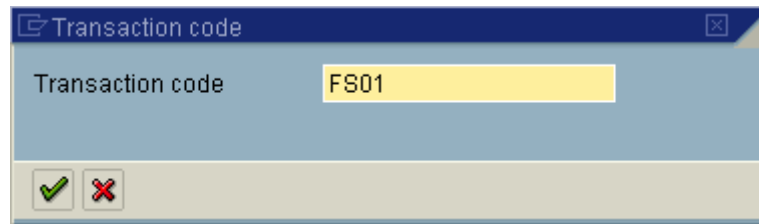
Select *Recordings* → *Create recording*. Fill the displayed fields.



| Recording | acctnt |
|-------------|----------------|
| Description | create account |
| Owner | OHLIGER |

Create Recording

After pressing *Continue* you can start to record the transaction whose transaction code you have to enter first.



| Transaction code |
|------------------|
| FS01 |

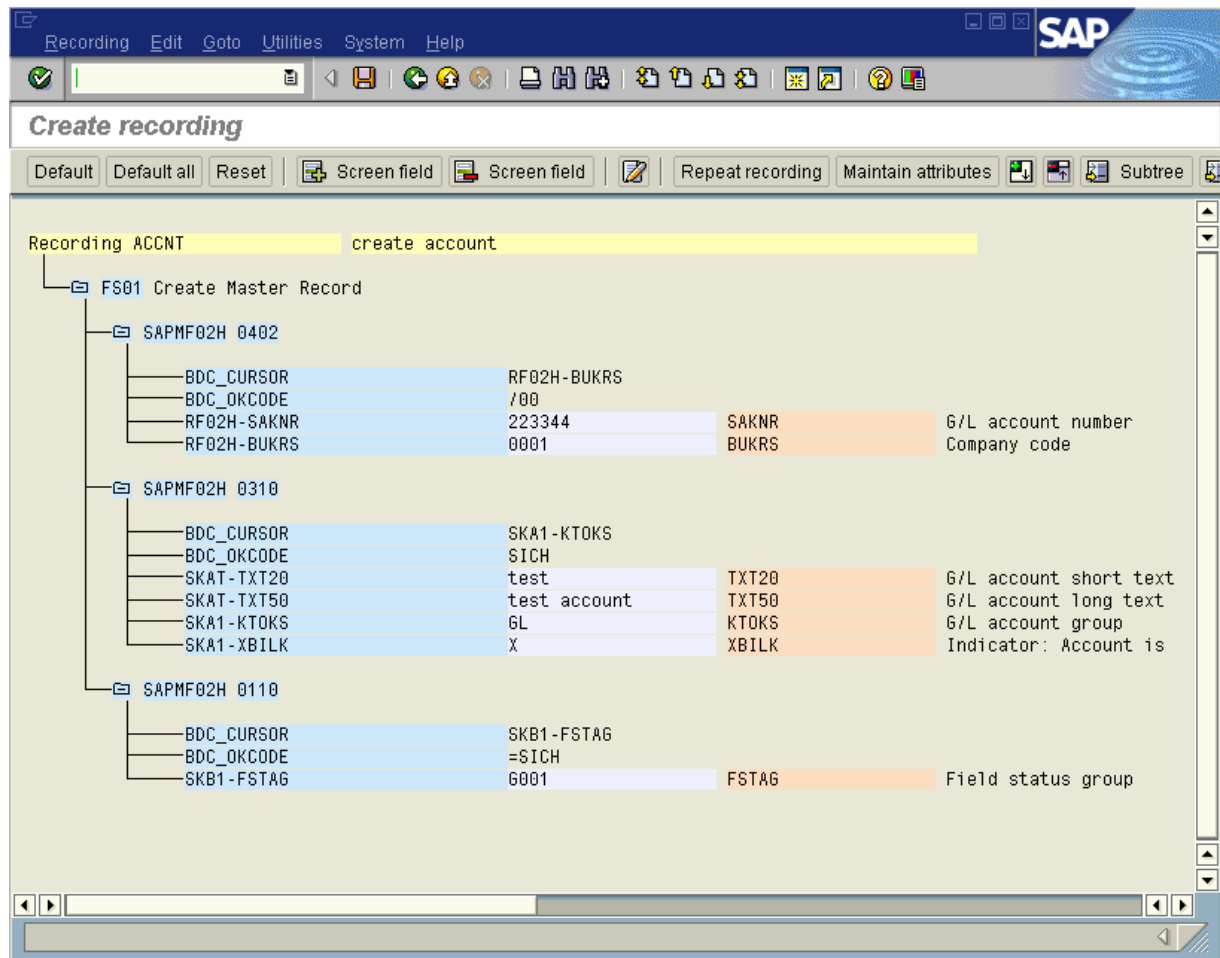
Create Recording: Enter Transaction Code



Note: If you do not know the transaction code of the transaction you selected: Select *System* → *Create session*. This displays the initial screen of the SAP system. Then select the relevant application component. This displays the relevant dialog screen. Select the transaction you want to record and then *System* → *Status*. The repository data includes the transaction code.

Now you can execute the selected transaction. Here you should input values in all the fields you intend to fill with the values from your legacy data later.

After the recording has finished, you can process it. You can delete or add fields.



Process Recording

You can assign field names freely. During the generation of the batch input session, the contents of these fields are assigned to the target fields displayed in the left column.

The following functions are available:

Default: Assigns the field name of the relevant target field and its field description.

Reset: Deletes field names and field descriptions.

Double-click: Edits fields names, field descriptions and default values.

Important

You may use field names repeatedly. However, in field mapping a field name can only be used once.

For all fields in which you did not specify a field name the specified default value is used for the batch input session generation. Thus these default values can be considered as constants. This is useful in particular with check boxes (e.g. MM01, view selection).

After you saved, the status line displays the following message: "Data saved successfully". The recording is now available among the attributes for the object.

7 Transport LSMW Projects

The LSM Workbench provides data transport for a project via both the SAP transport system and down-/upload. (Excluded are the presettings for IDoc inbound processing. These presettings should be manually created in every SAP system and every client.)

7.1 Generate Change Request

Choosing this function creates an SAP change request containing all information about an LSMW project. This SAP change request can be exported / imported with the usual means of SAP correction and transporting. You can find this function in the initial screen under *Extras -> Create change request*.

When transporting LSMW data this way, you can trace the transports any time in SAP correction and transporting.



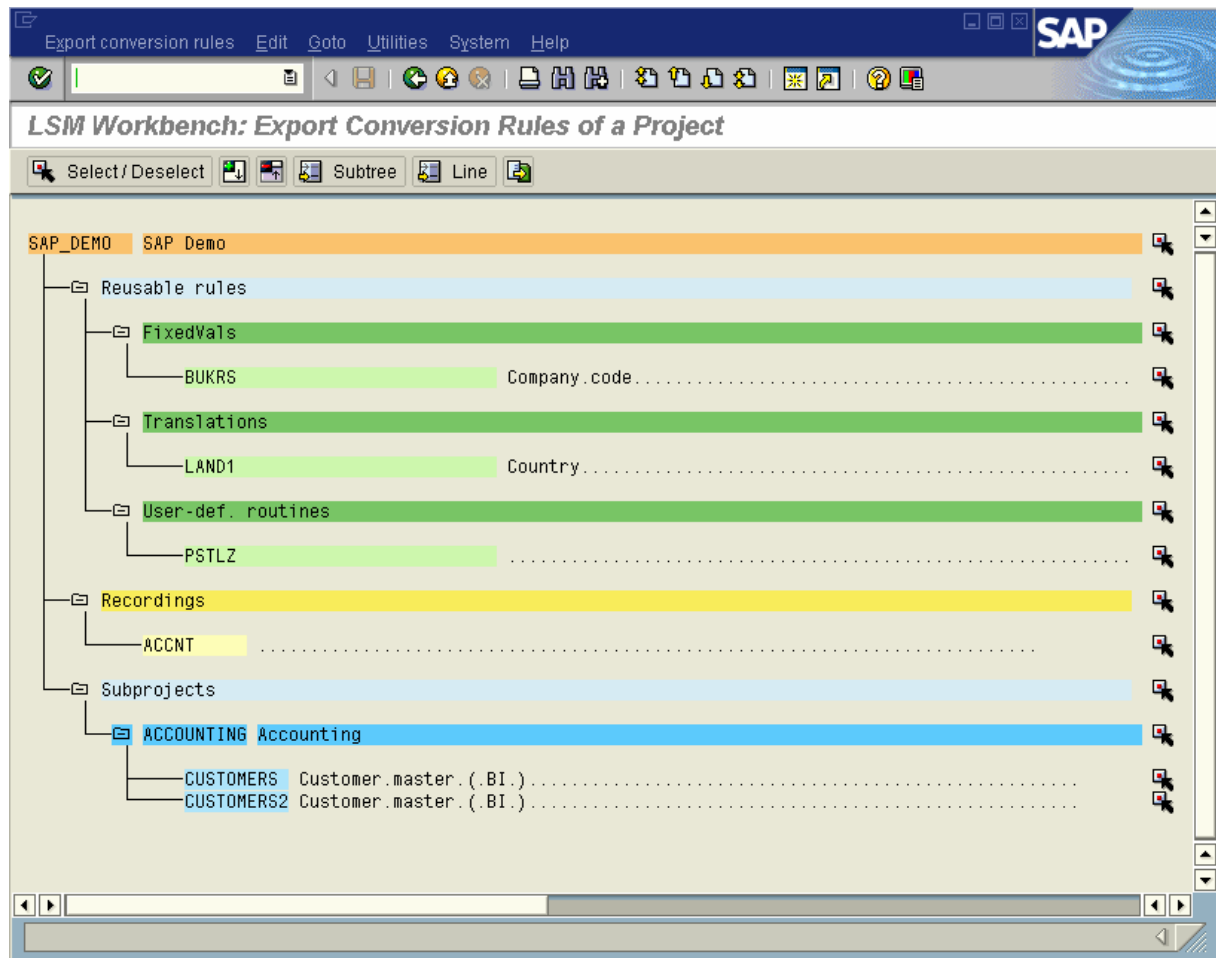
Note 1: When importing such a request, the complete project is deleted from the target system first. It is then created again.



Note 2: When exporting the transport request, all changes to the selected project made until the time of export are entered (not only until the time of creation of the transport request.)

7.2 Export Project

In the initial screen, select *Extras -> Export project*. This first displays the structure tree of the selected project. Via *Select / Deselect* you can select whether the entire project or parts of the project are exported. Then select *Export*. The program then creates an ASCII file.



Export Project: Project Table of Contents



Note: The selected elements are exported together with their documentation.

7.3 Import Project

The exported mapping and rules can be imported into another SAP system.

On the selection screen, select *Extras -> Import*. The program then prompts you to enter the name of the PC file. The file is imported and the contents are analyzed. After the analysis, a list of the subprojects and objects found is displayed.

You can now mark the objects to be imported. Project data existing already are check-marked. They are overwritten by the import.

You can prevent a project already existing in the target system from being overwritten by using function "Import under different name".



Note: The selected elements are imported together with their documentation.

8 Periodic Data Transfer

To a limited extent, the LSM Workbench also supports periodic data transfer. Preconditions are:

- The LSMW object has been created and tested completely.
- The "source application" periodically makes available one or several files on the SAP application server.
- The LSMW object does not access files on the frontend. (Files on the frontend cannot be read in batch mode.)

If all these conditions are met, you can set select button "Periodic" in step "Maintain object attributes". Then, step "Control program for periodic data transfer" is displayed in the overview of work steps. This program carries out the following steps in sequence:

- Read data
- Convert data
- Import data

This program (name: `/SAPDMC/SAP_LSMW_INTERFACE`) can be used according to your requirements.

Frame program for periodic data transfer

Project, subproject, object

Project: SAP_DEMO

Subproject: ACCOUNTING

Object: CUSTOMERS

General parameters

Flag file (path and name):

Values for wildcard (*): to

Handling of the input files (legacy data)

Leave after processing: ☒

Delete after processing: ☐

Rename after processing: ☐

Handling of the files with read and converted data

Leave after processing: ☒

Delete after processing: ☐

Rename after processing: ☐

Read data

☒ Amount fields -> 1234.56

☒ Date values -> YYYYMMDD

Variant (read program):

Convert data

Create file: ☒

Pass IDocs directly: ☐

Number of IDocs per package:

Variant (conversion program):

Import data

☒ Batch input: Submit session

☒ BAPI/IDoc: Process IDocs

Frame Program for Periodic Data Transfer: Selection Screen



Note 1: Specification of a flag file is optional.



Note 2: A flag file serves for creating a "handshake" with the application providing the input file(s):

- The control program for periodic data transfer is only executed if the specified flag file exists.
- After finishing data transfer, the control program for periodic data transfer deletes the flag file.
- The "providing" application should behave in a complementary way: Before new files are created, a check is carried out as to whether the flag file exists. If this is the case, the program stops. Otherwise, the files are generated, and the flag file is created.



Note 3: Some of the standard batch input and direct input programs use additional parameters. Some of these parameters are also used in other programs. For information about which parameters are used in which program, please refer to the coding of program /SAPDMC/SAP_LSMW_INTERFACE.

| Program | Parameter used | | | | | |
|--|-------------------------|----------------------------|--------------------------------|-----------|--------|------------|
| | Test run without update | Create batch input session | BI, DI, Call Transaction, Test | Lock mode | Action | User group |
| RAALTD01 RAALTD11 | X | | | | | |
| RCCLBI01 RCCLBI02 RCCLBI03 RCCTBI01 | | X | | | | |
| RCSBI010 RCSBI020 RCSBI030 RCSBI040 | | X | | | | |
| RCVBI010 | | X | | | | |
| RFBIBL00 | | | X | | | |
| RHALTD00 | | | X | | | |
| RLBEST00 RLPLAT00 | | X | | | | |
| RMDATIND | | | | X | | |
| RPUSTD00 | | | | | X | X |



Note 4: You can specify variants for the read program, the conversion program and (in case Batch/Direct Input) the Batch or Direct Input program. These variants have to be defined before.

9 Long Texts

To transfer long texts, there are two possibilities:

- Direct input program /SAPDMC/SAP_LSMW_IMPORT_TEXTS (object 0001, method 0001); this object is not available in the standard program. To make it available, run the following program : /SAPDMC/SAP_LSMW_SXDA_TEXTS
- Direct input program RSTXLTF (object 2000, method 0000); to be able to use this object, you have to download the transport from SAPNET (<http://service.sap.com/LSMW>) and import it into your system

9.1 Long Texts in the SAP System

Long texts (texts covering more than one line) are stored in a text pool in the SAP system. The key of a long text is composed of four parts:

| Key field | Meaning | Example | Length | Check table |
|-----------|--------------------|---|--------|--------------|
| OBJECT | Application object | AUFK = Order texts | 10 | TTXOB, TTXOT |
| ID | Text ID | Object AUFK <ul style="list-style-type: none"> • Id KOPF = Order header text • Id POSN = Order item text • Id RMEL = Order confirmation text | 4 | TTXID, TTXIT |
| NAME | Actual text key | Order number | 70 | (none) |
| SPRAS | Language | Text language | 1-2 | T002 |

9.2 Determine Text Key Structure

There is no uniform rule for the structure of the actual text key *NAME*. To determine the values for *OBJECT* and *ID* for a specific text type and the structure of *NAME*, proceed as follows:

- Display a text of the required text type (e.g. order header text) and branch off to the editor.
- There you can display the required information via *Goto* → *Header*:

| | | |
|---------------|-----------------------|-----------------------|
| Text name | TESTMATERIAL01 000101 | |
| Language | EN | |
| Text ID | 0001 | Sales text |
| Text object | MVKE | Material texts, sales |
| Short title | | |
| Short title 1 | MM01SAPMMG01 | |
| Short title 2 | | |
| Style | | |
| Form | SYSTEM | |
| Created by | OHLIGER | Changed by OHLIGER |
| Date created | 22.06.2000 | Changed on 22.06.2000 |
| Time | 17:22:25 | Time 17:22:25 |
| Release | 46B | Release 46B |

The following applies in the above example of a material sales text:

OBJECT = MVKE

ID = 0001

NAME

- Material number (18 characters) +
- Sales organization (4 characters) +
- Distribution channel (2 characters)

9.3 Develop Objects for Long Texts via Object 0001

The following target structures are available:

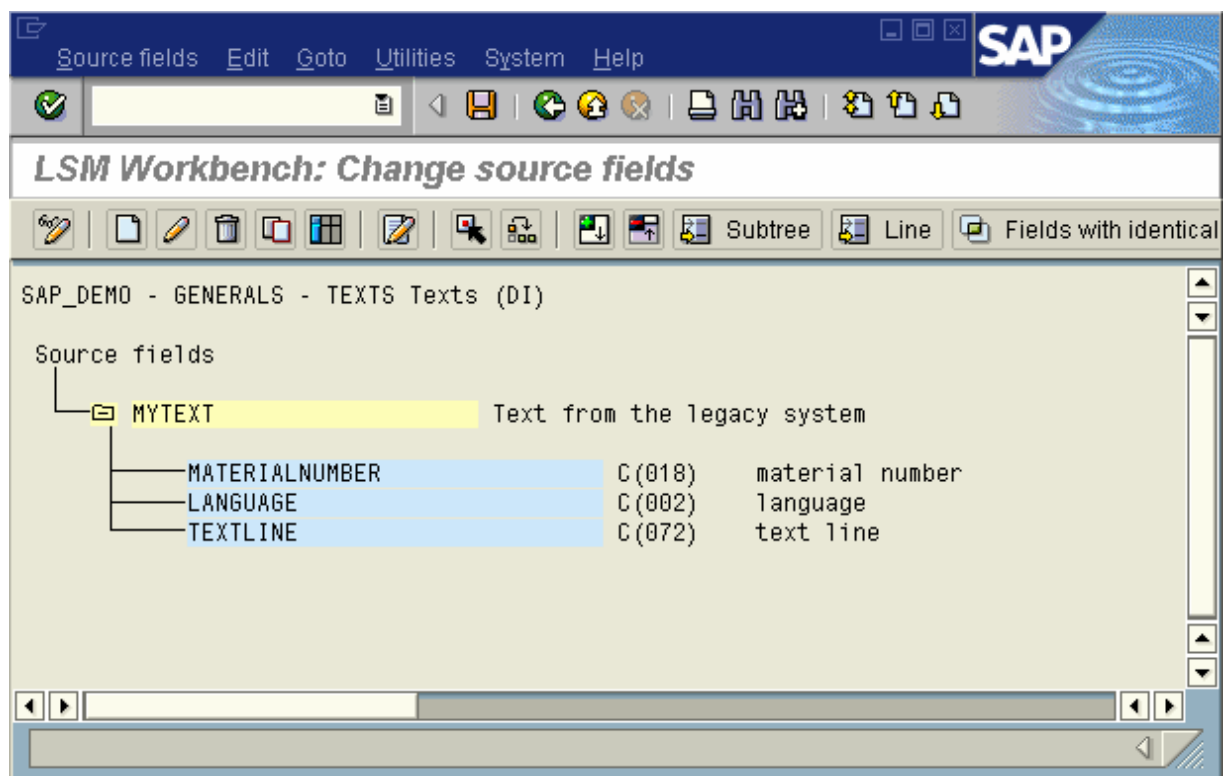
/SAPDMC/LTXTH: Long text header

- STYPE Record type (technical field, value = '1')
- OBJECT Application object
- NAME Text name
- ID Text ID
- SPRAS Language

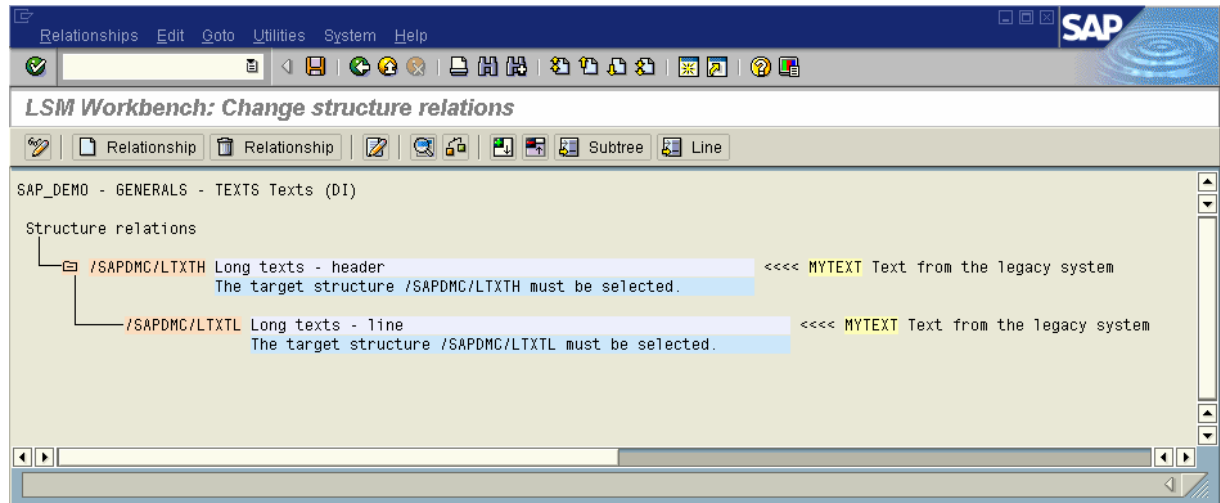
/SAPDMC/LTXTL: Long text text line

- STYPE Record type (technical field, value = '2')
- TEXTFORMAT Format field (2 characters)
- TEXTLINE Application object

Field TEXTFORMAT contains text formatting information. To simply map the field 1:1, enter character '*'.
 For the material sales text in the example, a migration object could look as follows:



Long Texts: Source Fields



Long Text: Structural Relationships

The screenshot shows the SAP LSM Workbench interface for 'SAP_DEMO - GENERALS - TEXTS Texts (DI)'. The main window displays the field mapping for the target field '/SAPDMC/LTXTH' (Long texts - header). The mapping is structured as follows:

- BEGIN_OF_RECORD**: Before applying conversion rules. Rule: Default settings. Coding: /SAPDMC/LTXTH = INIT_/SAPDMC/LTXTH.
- OBJECT**: Texts: application object. Rule: Constant. Coding: /SAPDMC/LTXTH-OBJECT = 'MVKE'.
- NAME**: Text name. Source: MYTEXT-MATERIALNUMBER (material number). Rule: Transfer (MOVE). Coding: /SAPDMC/LTXTH-NAME = MYTEXT-MATERIALNUMBER. /SAPDMC/LTXTH-NAME+18 = '0001'. /SAPDMC/LTXTH-NAME+22 = '01'.
- ID**: Text ID. Rule: Constant. Coding: /SAPDMC/LTXTH-ID = '0001'.
- SPRAS**: Language Key. Source: MYTEXT-LANGUAGE (language). Rule: Transfer (MOVE). Coding: /SAPDMC/LTXTH-SPRAS = MYTEXT-LANGUAGE. * Attention: Source field is longer than target field!
- END_OF_RECORD**: After applying conversion rules. Rule: Default settings modified! Coding: on_change_transfer_record.

The screenshot shows the SAP LSM Workbench interface for 'SAP_DEMO - GENERALS - TEXTS Texts (DI)'. The main window displays the field mapping for the target field '/SAPDMC/LTXTL' (Long texts - line). The mapping is structured as follows:

- BEGIN_OF_RECORD**: Before applying conversion rules. Rule: Default settings. Coding: /SAPDMC/LTXTL = INIT_/SAPDMC/LTXTL.
- TEXTFORMAT**: Tag column. Rule: Constant. Coding: /SAPDMC/LTXTL-TEXTFORMAT = '*'.
- TEXTLINE**: Text line. Source: MYTEXT-TEXTLINE (text line). Rule: Transfer (MOVE). Coding: /SAPDMC/LTXTL-TEXTLINE = MYTEXT-TEXTLINE.
- END_OF_RECORD**: After applying conversion rules. Rule: Default settings. Coding: transfer_record.
- END_OF_TRANSACTION**: After processing the data of a transaction. Rule: Default settings. Coding: transfer_transaction.
- END_OF_PROCESSING**: After processing the data.

Long Text: Field Mapping



Note: Statement on_change_transfer_record. has the effect that the text header is transferred only, if it has changed compared to the previous record (see 5.5.3).

9.4 Develop Objects for Long Texts via Object 2000

Please have a look at the documentation for program RSTXLITF first. There you will find very useful information concerning the file format the import program needs.

The following target structures are available (defined during the import of object 2000):

/SAPDMC/LSMW_TEXTHTEXT

/SAPDMC/LSMW_TEXTOBJEKT

/SAPDMC/LSMW_TEXTNAME

/SAPDMC/LSMW_TEXTID

/SAPDMC/LSMW_TEXTLANGUAGE

/SAPDMC/LSMW_TEXTFORM

/SAPDMC/LSMW_TEXTSTYLE

/SAPDMC/LSMW_TEXTFIRSTUSER

/SAPDMC/LSMW_TEXTFIRSTDATE

/SAPDMC/LSMW_TEXTFIRSTTIME

/SAPDMC/LSMW_TEXTLASTUSER

/SAPDMC/LSMW_TEXTLASTDATE

/SAPDMC/LSMW_TEXTLASTTIME

/SAPDMC/LSMW_TEXTTITLE

/SAPDMC/LSMW_TEXTTITLE1

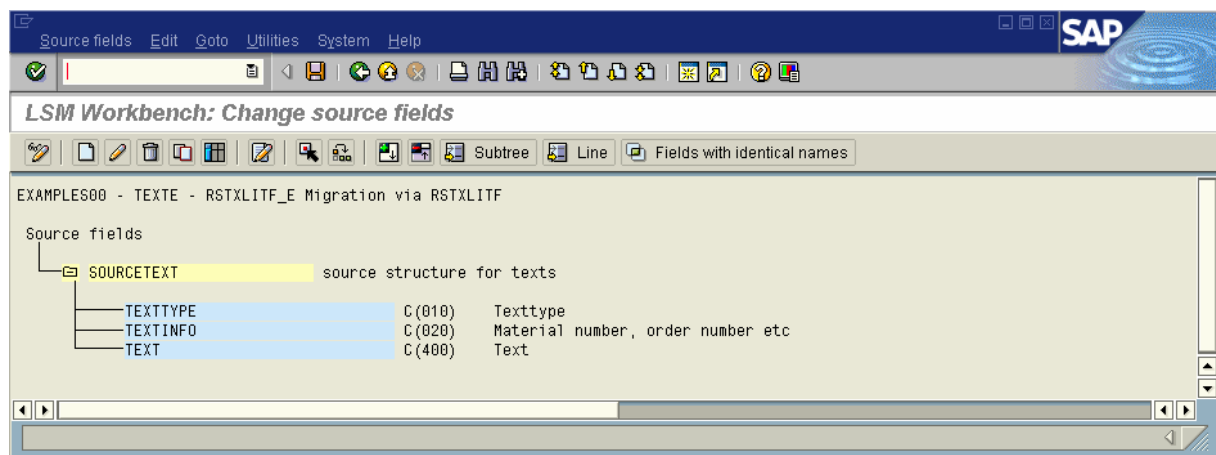
/SAPDMC/LSMW_TEXTTITLE2

/SAPDMC/LSMW_TEXTMAIN

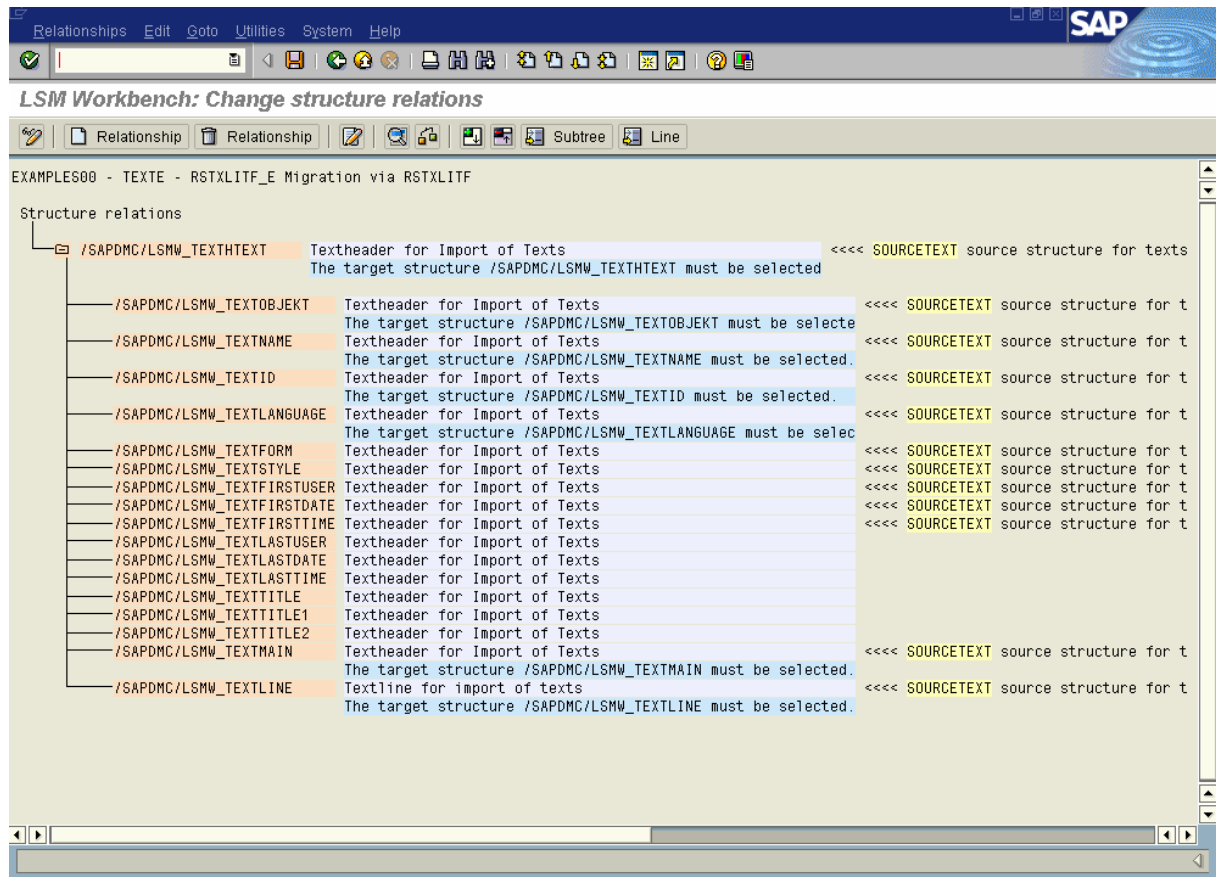
/SAPDMC/LSMW_TEXTLINE

Most of the fields for these structures are technical fields and are filled by default.

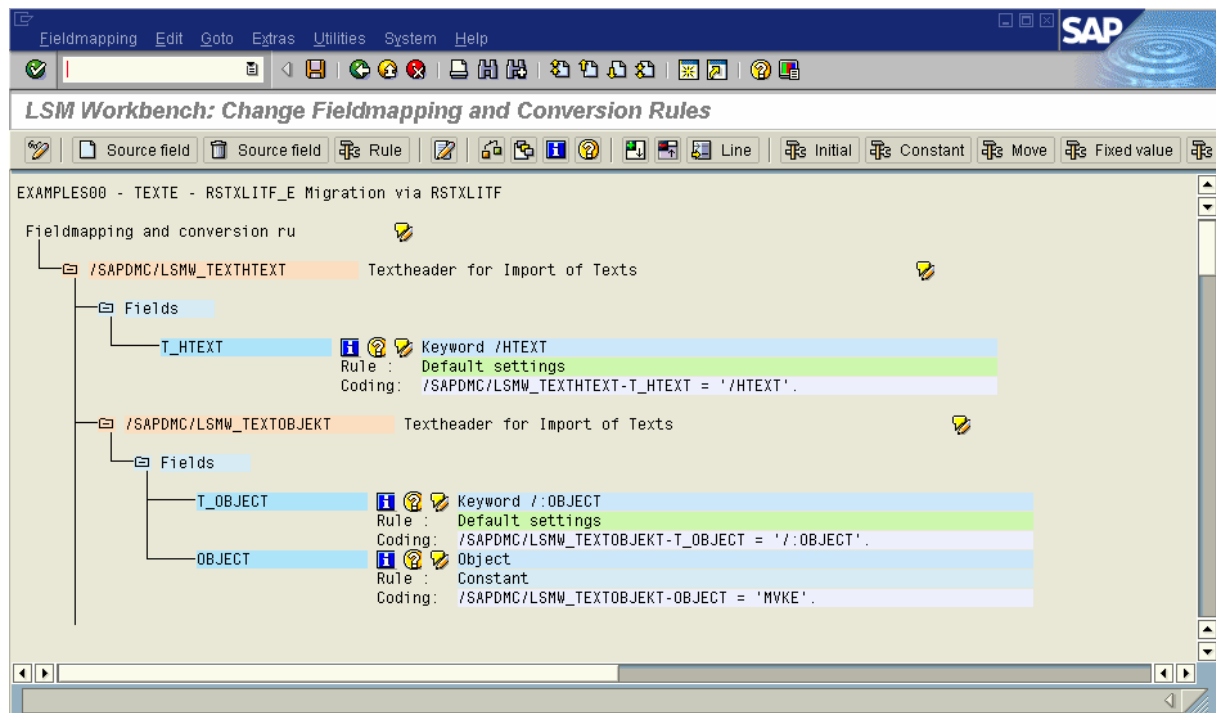
A migration object could look as follows:

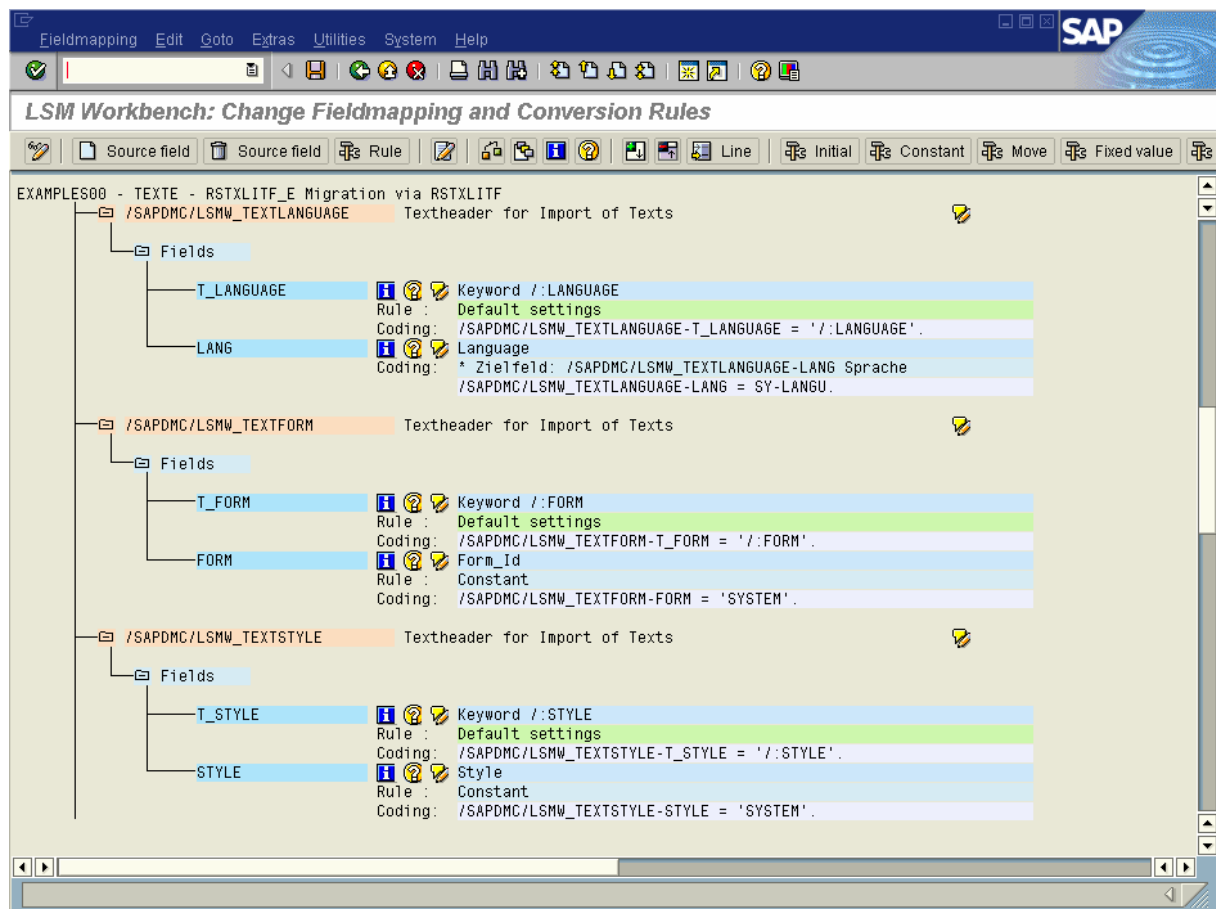
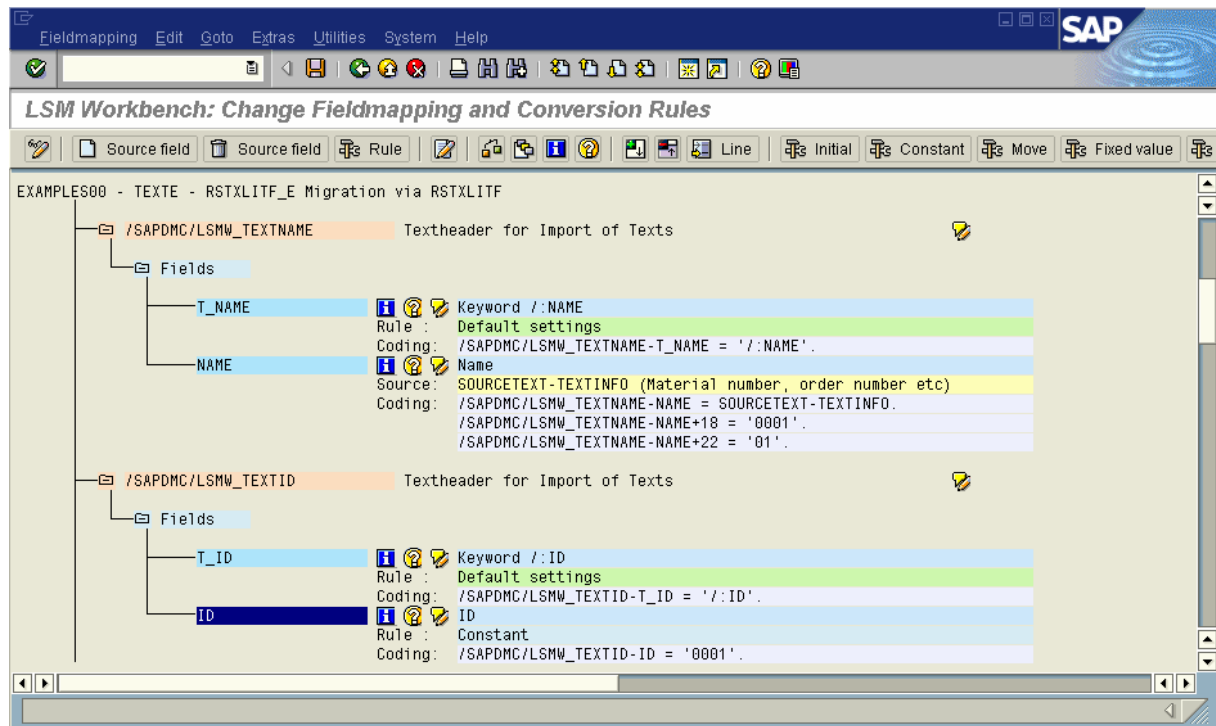


Source Fields



Structural Relationships





Fieldmapping Edit Goto Extras Utilities System Help

LSM Workbench: Change Fieldmapping and Conversion Rules

Source field Source field Rule Line Initial Constant Move Fixed value

EXAMPLES00 - TEXTE - RSTXLITF_E Migration via RSTXLITF

/SAPDMC/LSMW_TEXTFIRSTUSER Textheader for Import of Texts

Fields

T_FIRST_USER Keyword /:FIRST-USER
Rule: Default settings
Coding: /SAPDMC/LSMW_TEXTFIRSTUSER-T_FIRST_USER = '/:FIRST-USER'.

FIRST_USER Identifier
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTUSER-FIRST_USER Identifier
/SAPDMC/LSMW_TEXTFIRSTUSER-FIRST_USER = SY-UNAME.

/SAPDMC/LSMW_TEXTFIRSTDATE Textheader for Import of Texts

Fields

T_FIRST_DATE keyword /:FIRST-DATE
Rule: Default settings
Coding: /SAPDMC/LSMW_TEXTFIRSTDATE-T_FIRST_DATE = '/:FIRST-DATE'.

FIRST_DATE_DAY Creation date - day
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_DAY Anlegedatu
/SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_DAY = SY-DATUM+6(2).

DUMMY1 Space
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_MONTH Anlegeda
/SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_MONTH = SY-DATUM+4(2).

FIRST_DATE_MONTH Creation date - month
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_YEAR Anlegedat
/SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_YEAR = SY-DATUM(4).

DUMMY2 Space
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_YEAR Anlegedat
/SAPDMC/LSMW_TEXTFIRSTDATE-FIRST_DATE_YEAR = SY-DATUM(4).

Fieldmapping Edit Goto Extras Utilities System Help

LSM Workbench: Change Fieldmapping and Conversion Rules

Source field Source field Rule Line Initial Constant Move Fixed value

EXAMPLES00 - TEXTE - RSTXLITF_E Migration via RSTXLITF

/SAPDMC/LSMW_TEXTFIRSTTIME Textheader for Import of Texts

Fields

T_FIRST_TIME Keyword /:FIRST-TIME
Rule: Default settings
Coding: /SAPDMC/LSMW_TEXTFIRSTTIME-T_FIRST_TIME = '/:FIRST-TIME'.

FIRST_TIME_HOUR Creation time - hour
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_HOUR Anlegezei
/SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_HOUR = SY-UZEIT(2).

DUMMY1 Space
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_MINUTE Anlegeze
/SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_MINUTE = SY-UZEIT+2(2).

FIRST_TIME_MINUTE Creation time - minutes
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_SECONDS Anlege
/SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_SECONDS = SY-UZEIT+4(2).

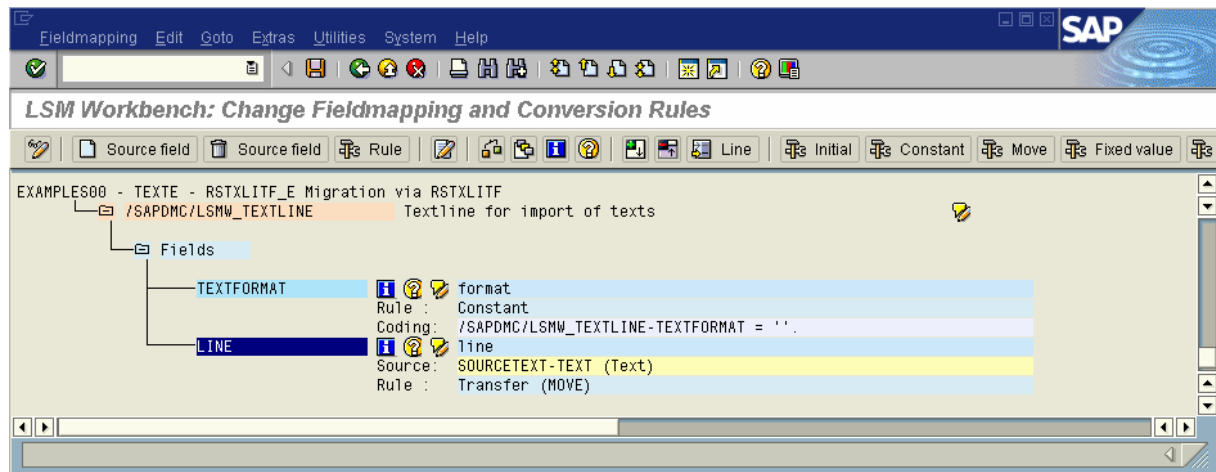
DUMMY2 Space
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_SECONDS Anlege
/SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_SECONDS = SY-UZEIT+4(2).

FIRST_TIME_SECONDS Creation time - seconds
Coding: * Zielfeld: /SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_SECONDS Anlege
/SAPDMC/LSMW_TEXTFIRSTTIME-FIRST_TIME_SECONDS = SY-UZEIT+4(2).

/SAPDMC/LSMW_TEXTMAIN Textheader for Import of Texts

Fields

T_MAIN Keyword /M
Rule: Default settings
Coding: /SAPDMC/LSMW_TEXTMAIN-T_MAIN = '/M'.



9.5 Import Texts

Texts are imported into the SAP system by means of direct input. The relevant direct input program can be easily called from the LSM Workbench via *Start Direct Input Session*.

Important

After the import of long texts, sometimes these cannot be read within the corresponding application. Via function module 'READ_TEXT', the texts are found, thus, they are stored correctly in the database. Some applications have a field in the master data which shows whether a long text exists or not. This field is not filled by the direct input programs (since these programs apply to all applications, and at runtime it is not known which application a text belongs to).

There are 2 possible solutions:

1. The flag is supplied by a user-defined report after the import
2. In the field assignments / mapping an update is coded on the respective table; however, this that the flag is set already during the conversion → if the text is not imported later, the flag is set, however, no long text is available.

10 Tips and Tricks

10.1 Determine the Transaction Code at Runtime

Situation: You want to transfer data a part of which has already been created in the system. You want to decide at runtime whether the data is created or changed.

Example: Customer master

Solution: Insert under "Global Data":

```
TABLES: KNA1.
```

Add the following coding for field BKN00-TCODE:

```
Select count(*) from kna1 where kunnr = <alte_kundennummer>.
if sy-dbcnt = 0.
    bkn00-tcode = 'XD01'.
else.
    bkn00-tcode = 'XD02'.
endif.
```

10.2 Skip a Record

Situation: You want to "skip" a record depending on a certain condition, i.e. this record shall not be converted and transferred to the output file

Solution:

```
if <condition>.
    skip_record.
endif.
```

10.3 Skip All Records of a Transaction

Situation: You want to "skip" all records of a transaction depending on a certain condition.

Solution:

```
if <condition>.
    skip_transaction.
endif.
```

10.4 Duplicate a Record

Situation: You want to create two (or more) target records from a source record.

Example: Your customer master of legacy files consists of one record containing among other things the fields "First name", "Name", "Phone number" for two contact persons. In the SAP system, a BKNVK record has to be filled for each contact person.

Solution: Your legacy structure is assumed to look as follows:

```
CUST Customer master
...
VORNAME1 First name of first contact person
NACHNAME1 Name of first contact person
TELEFON1 Phone number of first contact person
VORNAME2 First name of second contact person
NACHNAME2 Name of second contact person
TELEFON2 Phone number of second contact person
...
```

Create the following rules:

```
...  
BKNVK-NAME1  
    ← CUST-NACHNAME1 (Move)  
BKNVK-TELF1  
    ← CUST-TELEFON1 (Move)  
BKNVK-NAMEV  
    ← CUST-VORNAME1 (Move)
```

and add

```
End_of_Record  
transfer_record.  
...  
BKNVK-NAME1 = CUST-NACHNAME2.  
BKNVK-TELF1 = CUST-TELEFON2.  
BKNVK-NAMEV = CUST-VORNAME2.  
transfer_record.  
  
...
```

at processing time. This creates two BKNVK records.

10.5 Error messages in the conversion log

How can user-defined error messages be added to the conversion log?

1. Use of `WRITE` command in the Coding
 - a. Example: `WRITE: ,Error during conversion of field xxx'`
2. In the editor use 'Insert -> Message' to use a message already existing in the SAP system.
 - a. Example (Coding after using the menu function and filling the fields):

```
WA_ERRORTAB-ID = '/SAPDMC/LSMW'.  
WA_ERRORTAB-MSGNO = 012.  
WA_ERRORTAB-PAR1 = 'A'.  
WA_ERRORTAB-PAR2 = 'B'.  
WA_ERRORTAB-PAR3 = 'C'.  
WA_ERRORTAB-PAR4 = 'D'.  
INSERT WA_ERRORTAB INTO TABLE G_ERROR_TAB.
```

The message will be added to the log.

10.6 Extra Handling for “POS-IDOCs”

What is a “POS-IDOC”? – These are IDocs where the control record (EDI_DC40) has to be filled with source data (example message type WPUWBW, where the store number is transferred via the control record). This means control record EDI_DC40 needs to be available as target structure in step “maintain structure relations”; in the fieldmapping rules and processing points have to be identical to the other target structures

How does the processing look like for “POS-IDOCs”?

Step “maintain object attributes”

Please set flag “Enable structure relation for EDIDC40”

Attributes

Object: STOCKS **stock level**

Owner: OHLIGER OHLIGER

Data transfer: ☒ once ☐ periodic

File names: ☐ system dependent

Object type and import technique

☐ Standard Batch/Direct Input

Object:

Method:

Program name:

Program type:

☐ Batch Input Recording

Recording:

☐ Business Object Method (BAPI)

Business object:

Method:

Message type:

Basic type:

☒ IDoc (Intermediate Document)

Message type: WPUWBW POS interface: Upload goods ...

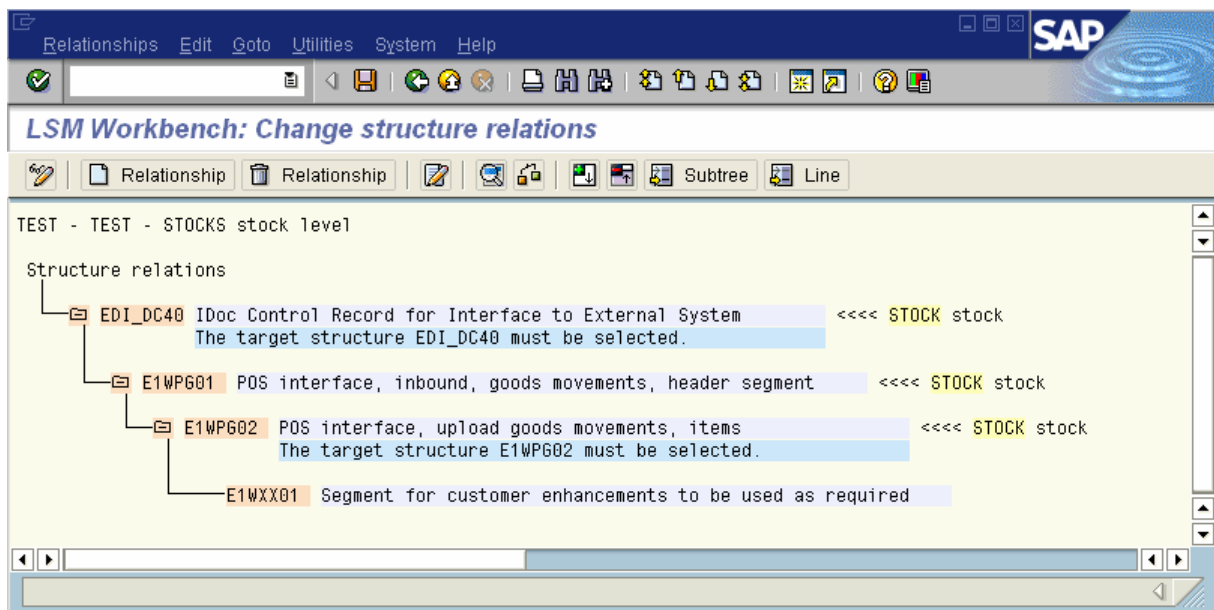
Basic type: WPUWBW01 POS interface: Upload goods ...

Enhancement:

☒ Enable structure relation for EDIDC40

Step “maintain structure relations”

Please assign a source structure to target structure EDI_DC40:



Step “maintain fieldmapping and conversion rules”

→ EDI_DC40 is available as target structure inclusive processing points

In all other steps there are no changes.

11 Upgrade from LSMW 1.0 to LSMW 1.8

11.1 Differences Between Version 1.0 and Version 1.8 of the LSM Workbench

The following lists the main differences between version 1.0 and 1.8 of the LSM Workbench:

- **Transaction code:** The transaction code is "LSMW" (in version 1.0: "DLSM"). Version 1.0 is still available after installing version 1.8.
- **Key field names:** New: "Project" instead of "Legacy system", "Subproject" instead of "Legacy system release", "Object" instead of "Migration object". All these key fields now have a length of 10 places (in version 1.0, they had four places).
- **Naming convention:** The names for project, subproject and object can be assigned freely without any restrictions.
- **Changed description:** New: "Recording" instead of "User-defined migration object class". The name of a recording may have up to 10 characters and should meet the ABAP naming conventions (e.g. first character = letter). Every recording is assigned to exactly one project.
- **Changed description:** New: "Reusable rules" instead of "Central rules".
- **Object attributes:** In the object attributes, the object and import types are defined. (In version 1.0, this definition was already made via the selected object name.) You can now assign any number of recordings to an object.
- **Import techniques:** In addition to standard batch input/standard direct input and recording, BAPI and IDocs are available as additional import techniques.
- **Source structures:** The name may have a length of up to 25 characters. Information on the identification of a record is now found with the source fields. Now, you can also define several structures at the top hierarchy level.
- **Source fields:** Now, additional field types for amount fields AMT1, AMT2, AMT3, AMT4 are available. The record description of a structure can be comfortably transferred via the *Copy* symbol from various sources: Upload from file, copy of another object, copy from Data Repository, from data file (with field names in the first line). Fields for structures of the top hierarchy level can be marked as selection parameters (for *Import/convert data*).
- **Structural relations:** The program displays the information whether a structure is a required segment.
- **Field mapping and rules:**
 - **Display variant:** You can select the elements to be displayed: Global data, processing times, technical fields, initial fields, coding.
 - **Processing times:** You can complete the ABAP coding at various times of data conversion. In version 1.0, you used additional includes with user-defined routines.
 - **Field documentation, F4 help:** Symbols per target field
 - **Source fields not assigned:** *Menu Extras* → *Source fields not assigned* displays the source fields that are not assigned.
 - **Additional conversion techniques via pushbutton:** Prefix, suffix, transfer left-justified, user-written routine
 - **Global functions:** In addition to the functions `transfer_record` and `skip_record` already available in version 1.0, other functions are available: `transfer_this_record`, `at_first_transfer_record`, `on_change_transfer_record`, `transfer_transaction`, `skip_transaction`.
 - **Editor:** Check the coding for a target field; insert source fields, global variables, global functions; Pretty Printer.

- **Comparison with Data Repository:** No longer required.
- **WHERE relationships:** No longer required.
- **Files:** All definitions in connection with files are merged in the two processing steps *Specify files* and *Assign files*. Here, '*' may be used as wildcard.
- **Import data:** Replaces and enhances functions *Spreadsheet Interface* and *Host Interface* of version 1.0. Now, you can use any combination of PC and server files. Date fields and amount fields are converted into an internal or calculation format by default. If required, the file import program generates itself again.
- **Convert data:** The data to be converted is processed target-oriented.
- **Action log:** All actions for an object are stored in an action log.
- **Download/upload rules:** Can now be carried out simultaneously for all parts of a project. The dataset produced in this process is considerably smaller than in version 1.0.
- **Display/change:** In many functions you can switch between *Display* and *Change*.
- **Documentation:** You can store your documentation at a total of 25 levels. The total documentation for a project can be further processed in hierarchical form.
- **Recordings:** Version 1.8 does not generate a data repository structure for a recording. The data of a recording are stored in LSMW internal tables. There is no change from the user's point of view.

12 Transfer of LSMW Data from Version 1.0 to Version 1.8

When transferring data from other systems, export them under version 1.0 and import them under version 1.0 as well. To transfer data to 1.8, use *LSM Workbench → Transfer LSMW data from version 1* on the initial screen. The same applies to transferring data from the same system.

Normally all LSM Workbench data transferred using the standard procedure can easily be transferred into version 1.8. Special features to be taken into account mainly concern advanced users who already made complex modifications to the LSM Workbench during data migration with LSM Workbench version 1.0.

Please take into account:

- Version 1.8 no longer has a generation lock.
- User-written ABAP coding should be checked via syntax check.
- User-defined translation variants will be lost.
- Includes created in version 1.0 and included in the object attributes via flag should be transferred manually.
- If you used `perform skip_record` or `perform transfer_record` so far, you can assign a corresponding processing time to the coding now.
- The specifications for identifying source structures in files should be maintained subsequently → this means: if there is data for more than one structure in a source file the specifications had to be done via offset and value during the definition of the source structures; in version 1.8 the specification is done via field 'Identifying field value' in the definition of the source fields
- After transferring the data from version 1.0, field mapping should be checked in any case.
- If you used recordings, you should switch to change mode for each recording and save the changes there.

13 Upgrade from LSMW 1.5 to LSMW 1.8

13.1 Notes on the Upgrade to LSMW 1.8

- All objects created under LSMW 1.5 are kept in LSMW 1.8.
- Transferring data from LSMW 1.0 to LSMW 1.8 is still possible.
- Transporting rules from LSMW 1.5 to LSMW 1.8 is possible. Vice versa, the specifications for the files may have to be corrected.
- The file settings may have to be corrected. First, run program /SAPDMC/SAP_LSMW_REPAIR_15 once, and check the settings ("Specify files").
- Check the fieldmapping for EVERY OBJECT. If you have created coding for the processing points `__BEGIN_OF_RECORD__`, save this. Then restore the default. Finally you can insert your own coding again.
- All programs generated in LSMW 1.5 have to be regenerated manually.

13.2 Corrections

- **Step numbering and welcome popup:** Numbering On/Off --> Welcome popup is displayed again: Error corrected.
- **Interfaces of function modules compatible with 4.0:** In some function modules, tables with reference type are used. The entered type is a structure, however. This is tolerated in 4.0B. When testing the individual function module, a syntax error is reported, however.
- **Create change request:** No authorization check: was added.
- **Object attributes:** When pressing F4 for standard BI/DI objects, objects with numbers < 8000 are displayed only: Error corrected.
- **Field mapping:** Descriptions and documentation for target fields of IDoc segments were partly not found: Error corrected.
- **User-written routines:** During creation, 1 input parameter and 1 output parameter were assumed automatically. Now, these values can be entered in a popup.
- **Display data read program, data conversion program:** When one of these functions were called before the corresponding program had been created, a termination occurred: Error corrected.
- **Display read data:** On the detail screen, packed fields were displayed without decimal places: Error corrected.
- **Generate the data conversion program:** Under certain conditions, the rules were confused: Error corrected.
- **Copy an object:** F4 - Help for subproject, object, source structure did not work: Error corrected. When the object contained wildcards, the copy process abended: Error corrected.
- ➔ please see note 168644 for further changes

13.3 Developments

- **Display <-> change:** In the steps screen you can specify whether the program switches to display or change mode when you double-click.
- **Source fields:** Field definition is now possible with table control as well.
- **Field mapping, display attributes:** Item `__FORM_ROUTINES__` is displayed separately for selection. Technical fields are marked with rule type „Default setting“. The default setting can be restored via *Extras* ➔ *Restore default*. The processing point `__BEGIN_OF_RECORD__` is preset.

- **Global functions:** Additional global function `transfer_this_record 'XXXXX'`, to transfer the specified segment.
- **Specify file:** Standardization: File on the Frontend and files on the application server. All files can (independent of the location)
 - contain data for one or more source structures,
 - have separators or not,
 - field names at the start of the file (one record per source structure) or not,
 - be a text or binary file,
 - have a different Codepage.
 - From application server: all Codepage
 - From Frontend: ASCII or IBM-DOS
- **Default file names for read and converted data:** Blanks within a word are replaced by underscores.
- **Read data:** Usage of sorted internal tables yielding a massive performance gain when "merging" several files.
- **IDoc inbound processing:** You can now additionally specify a tRFC port. During data conversion you can determine whether a file is to be created or the data is directly transferred in packages to function module `IDOC_INBOUND_ASYNCHRONOUS`. (This function module creates IDocs in the database.)
- **Control program for periodic data transfer:** Now also works together with wildcards (*) in file names. Some important parameters of BI/DI programs are forwarded to the outside.

14 Final Remarks

There only is to wish you success in your data migration project.

Yours sincerely

Your SAP LSMW team