# The Legacy System Migration Workbench (LSMW) — A Guide to Data Migration with BAPIs

Arthur Wirthensohn

*Arthur Wirthensohn is a senior consultant at EDS Switzerland and a member of EDS's international Technical Leadership Network. Currently, he is the technical lead of the EAI department, which delivers SAP-related services like ALE/EDI, SAP Business Workflow, Web Application Server, Data Migration Services, and ABAP Programming.*

*(complete bio appears on page 86)*

When you need to import data from an external application into an SAP system, you have several alternatives, some of them easier, faster, and more cost-effective than others. Planning a data migration from an external application to an SAP system involves, first of all, looking at the list of SAP business objects that will be needed. For each business object, you must choose the appropriate technology for migrating data, keeping in mind that most of the source data must be converted to a suitable data structure before it can be uploaded into the R/3 system.

If a standard SAP report exists for importing the data of an SAP business object, you can be fairly confident that the report will work — though you probably will need to do some additional ABAP programming, which can be time-consuming and cumbersome, not to mention its potential for complicating change management in your migration project. If no standard report exists, the data migration can be programmed from scratch, but that might involve considerable effort and some degree of doubt regarding its feasibility, since you won't be able to test the concept until a good deal of time and effort has already been invested.

Another alternative is to use the Data Transfer Workbench (or DX Workbench, transaction SXDA), an integrated SAP tool that supports data analysis and transfer. However, this tool will not convert the structure of the source data to satisfy the requirements of the target business object in the SAP system, and it will not map the source data to the target data.

Fortunately, there is yet another alternative, and it is the one I recommend you make your first choice for performing data migrations (with a few exceptions, which I'll get to in a moment). SAP's Legacy System Migration Workbench (LSMW) is a free add-on tool that supports data migration for numerous business objects with a user-friendly mapping feature. It usually requires little or no programming for data conversion and mapping. With the LSMW, you simply search for an appropriate object type from the selected import technology (for example, BAPIs or IDocs) and then use the LSMW to map the source data structure to the target data structure of the object. The LSMW does the rest. If there is no satisfactory object type for the data you need to import, the LSMW provides a feature to create your own object type — for example, its integrated batch input recording function allows you to create a batch data communication (BDC) processing report by walking through the posting transaction.

There are only two situations in which the LSMW does not make sense for a data migration:

- When you are already using a third-party tool for data mapping and data conversion in combination with an SAP data exchange tool (e.g., the DX Workbench). If you are satisfied with the third-party tool, and the SAP data exchange tool supports the appropriate object type, you can just keep on using it.

- When a data conversion tool is not required (e.g., conversion is done in the legacy system). In this case, you will use the DX Workbench (transaction SXDA).

This article shows project managers and team members, consultants, and programmers how to work with the LSMW to perform easier, faster, and more cost-effective data migrations. Step by step, I will take you through the process of importing data as part of a data migration project, using the LSMW and the BAPI import technology (the LSMW supports several technologies for importing data, including BAPIs, standard batch input and direct input, and ALE IDocs). Seeing how it is done using one import technology should make it fairly easy to understand how to utilize the other import technologies supported by the LSMW.

We'll start by taking a closer look at the LSMW — what it is, and what you can do with it. Then, after walking you through an example migration with the LSMW, I'll provide some insight on how to use the LSMW and the DX Workbench to analyze data for import into SAP systems, and how to test a "migration concept" — that is, how to test whether a selected object type and import technology will yield satisfactory results.

Because I will be using many similar-sounding terms throughout the article, I have provided a brief glossary on the next page that you may find useful.

## LSMW Basics: What It Is and How to Install It

The LSMW is an add-on tool for converting data and importing it as business objects into SAP systems, including R/3, CRM, and APO. Note that I said the LSMW tool *converts* and *imports* data. It does not *extract* legacy data. Specifically, the LSMW will:

- Transfer data — once or periodically — from a server or client (frontend) file system to an SAP system. (Periodic data transfers may only be made from a server file system.)

- Map source data to target data and perform data conversions (converting the source data structure to the target data structure).

- Utilize the following import technologies: batch input, direct input, BAPIs, and IDocs.

- Use a recording function (batch input recording) to generate an object type. This is a report that creates data in a batch data communication

## Glossary of Terms

**Business object:** In SAP systems, business objects are the representations of real-world objects, like material masters, G/L accounts, or sales orders. You can view examples of business objects in the Business Object Repository (BOR) via transaction SWO2.

**Instance of a business object:** An instance of a business object is the data of a single business object, which can be identified by a unique key (e.g., sales order number 1020).

**LSMW object:** An LSMW object is the implementation in the LSMW for importing data using a certain import technology (such as BAPIs or IDocs) and specified rules. There may be more than one LSMW object for a single business object — for example, for the business object material master there may be an LSMW object for importing sales materials and also one for importing raw materials. An LSMW object is organized as a subelement in the LSMW project hierarchy.

**Object type:** An object type is a program or method for importing data that is specific to a particular business object and import technology — for example, when using the BAPI import technology, the BAPI CREATE is the object type for the master data characteristics.

*Figure 1 LSMW Version and SAP Release Compatibility*

| SAP System Release | Example | LSMW Version | Comment | Available Since |
|---|---|---|---|---|
| R/3 3.x | R/3 3.1I | 1.0 | No further development planned. | 1998 |
| R/3 4.x | R/3 4.6C | 1.5 to 1.72 | Versioning for 4.x releases started with 1.5. | 1999 (for 1.5) |
| Web AS 6.x | CRM 3.0 | 3.0 | Functions are equal to those of version 1.72. | 2001 |

(BDC) structure, similar to the R/3 transaction recorder (transaction SHDB).

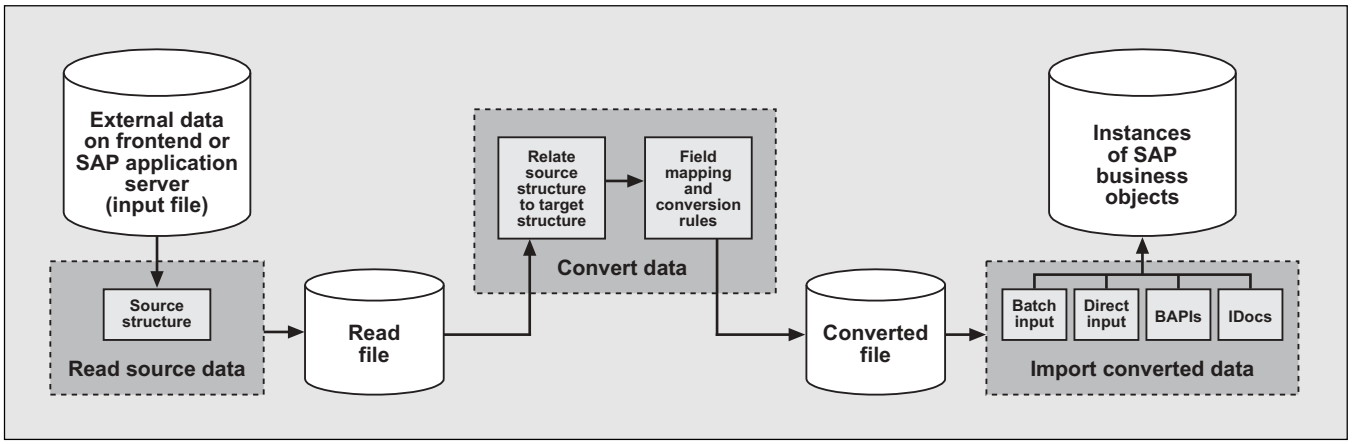
The LSMW is supported by SAP via the Online Service System (OSS) component BC-SRV-DX-LSM, and it can be downloaded from the SAP Service Marketplace at **http://service.sap.com/LSMW**/. For information on which LSMW versions are compatible with which SAP releases, see **Figure 1**.

### *Installation*

Because the LSMW is not an integrated part of the SAP system, it must be installed separately as follows:

1. Check to see if the LSMW has been installed already by calling transaction LSMW. If it has, the LSMW will start. Otherwise, you will get the

*Figure 2 The LSMW Information Flow*




message "transaction LSMW does not exist." (If the LSMW is there, but you do not have the authority to access it, proceed to step 3.)

2. Ask your system administrator to download the appropriate LSMW release from the SAP Service Marketplace (**http://service.sap.com/LSMW/**.) Also request the "Installation Guide," which is located in the literature section of the LSMW media center.

3. Ask your system administrator to check your authorizations. The LSMW documentation[1] lists the necessary authorization objects.

## How the LSMW Migrates Your Data

**Figure 2** shows the main information flow in the LSMW. The LSMW carries out three major activities in this process:

- It reads the external input data (the data to be migrated) and creates a read file.

- The LSMW applies mapping and conversion rules to the data in the read file to create a converted file. The data in this file conforms to the required data structure of the target application.

- The LSMW then imports the converted data into the target application using one of the following import technologies: standard batch input, direct input, BAPIs, or IDocs.

Let's look at each of these activities in detail.

### *Reading the Source Data*

The LSMW is very flexible regarding the structure of the external data (i.e., input) file. It supports various delimiters, field sequences, and the like. The data may be structured in one or more data files (for example, in File 1 for header data, File 2 for item data, and so forth). External data may be located on any SAP application server or on the frontend PC.

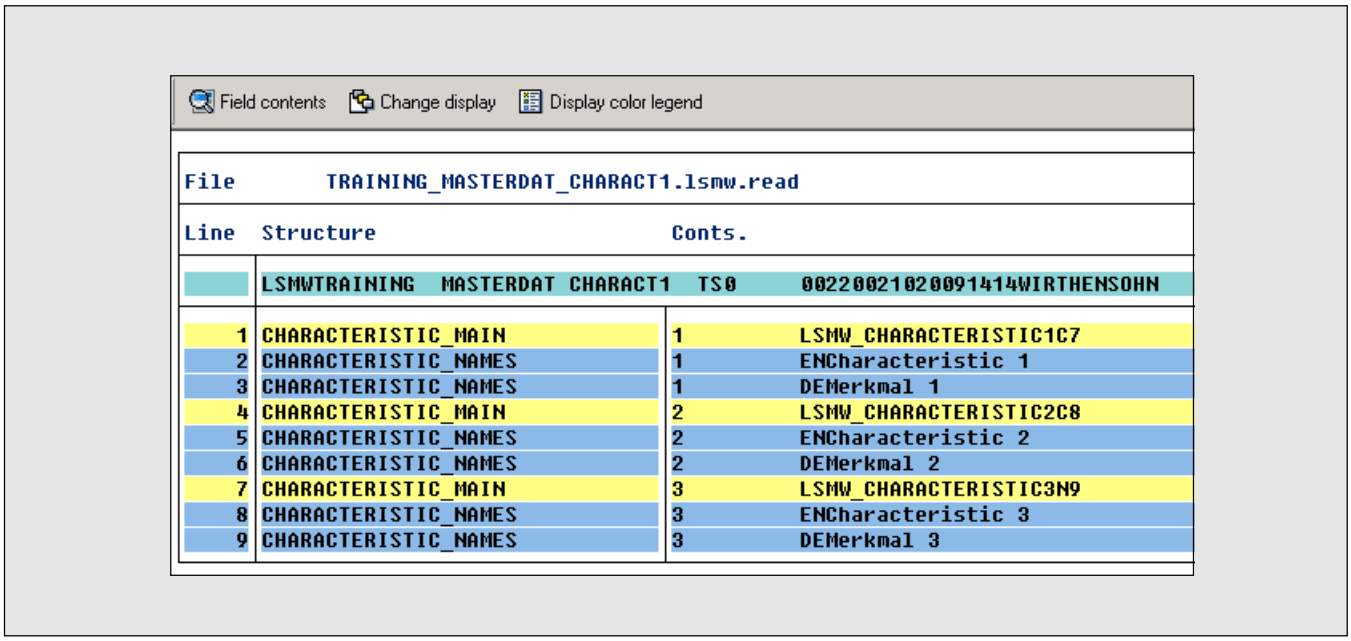
The LSMW reads the external data into a single sequential file — the read file. If there is more than one input file, it sorts the data according to instances of the business objects in the files (more on this later).

By default, the naming convention of the read file is *<project>_<subproject>_<object>.lsmw.read*. The file is saved in the SAP directory DIR_HOME.

[1] For information on obtaining the LSMW's "Quick Reference Guide," see the "Additional Resources" sidebar on page 85.

*Figure 3* *An Example of a Read File*




The LSMW supplies a viewer program so that you can look at the read file. **Figure 3** is an example of a read file called *TRAINING_MASTERDAT_CHARACT1.lsmw.read* that is displayed in the viewer, where "TRAINING" is the project, "MASTERDAT" is the subproject, and "CHARACT1" is the object (more on projects, subprojects, and objects in the upcoming migration walkthrough).

### *Converting Data*

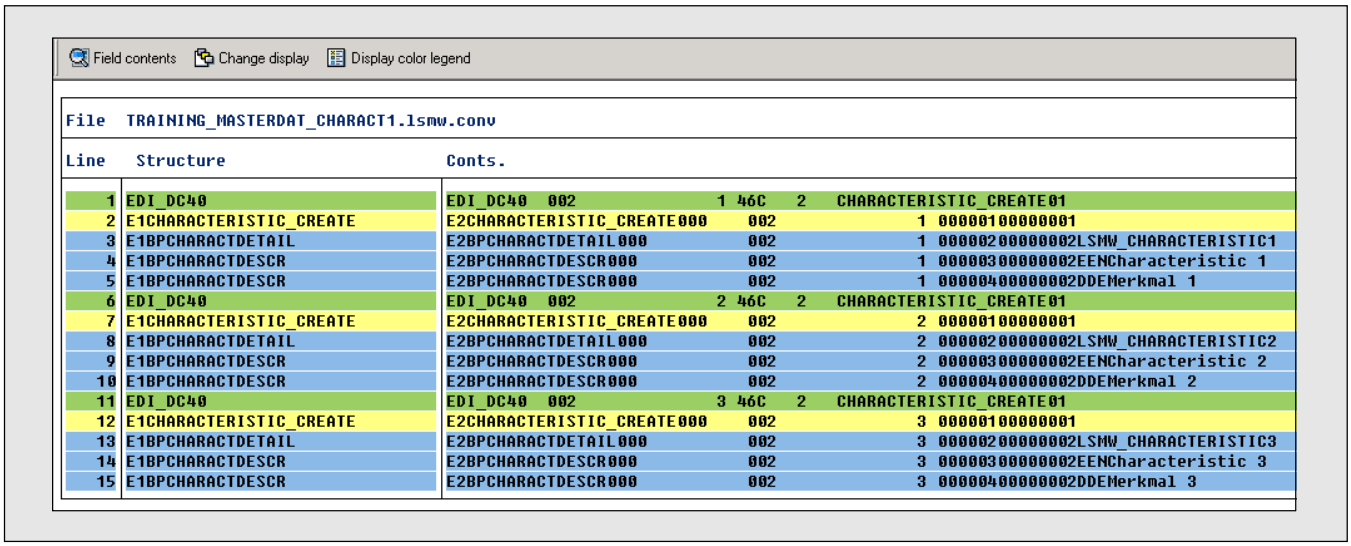
Before the LSMW can convert data to the structure of the target application, you or someone on the data migration team must relate the source data structure to the target data structure by making assignments in the LSMW (see "Step 6: Apply Mapping and Conversion Rules to the Data" in the upcoming walkthrough). The target structure is an SAP format predefined by the import technology and object type you have chosen:

- The *batch input* and *direct input* import technologies use the batch data communication (BDC) structure of the business object.
- Both the *IDoc* and *BAPI* import technologies use the IDoc structure of the business object. BAPIs are integrated into SAP's Application Link Enabling (ALE) technology, which is based on using IDocs for transmission of data into an SAP system. The IDoc import technology uses the same function modules for inbound processing that ALE uses.

> ✓ ***Note!***
>
> *You can create your own data transfer BAPIs by generating their corresponding IDoc types and message types, including the corresponding control tables. If you want to learn more about this, see the "Additional Resources" sidebar on page 85, under "BAPIs for Mass Data Transfer."*

Once you have made assignments that relate the source structure to the target structure, the LSMW maps the source data to the target data and applies conversions. The LSMW supplies numerous features for data mapping and conversion, including:

*Figure 4 An Example of a Converted File*




- Moving data from source fields to target fields
- Setting constants
- Predefining and applying reusable rules for data translations, fixed values, and routines
- Inserting ABAP coding for custom data manipulation
- Initializing target fields

The LSMW writes the results into the converted file, which is structured in the appropriate SAP format. By default, the converted file is named *<project>_<subproject>_<object>.lsmw.conv*. You can check the converted file using the viewer program (**Figure 4** shows a converted file called *TRAINING_MASTERDAT_CHARACT1.lsmw.conv*).

### *Importing Converted Data*

Using the chosen import technology, the LSMW tries to post the converted data as instances of the business objects.

To import the converted file into the SAP system, the LSMW calls the SAP standard or custom-made application program associated with the import technology and the business object. This means that:

- All features of the import programs depend on the import technology and the corresponding program or method.
- All errors or strange behaviors should be supported in the OSS component of the import technology and/or of the import program.

## *An Example Data Migration in Eight Steps*

Let's now walk through an example data migration project. A typical data migration project consists of the following eight steps:

1. Call transaction LSMW.
2. Create the LSMW project hierarchy.
3. Define the IDoc inbound processing.
4. Define the object attributes.
5. Define the source data.
6. Apply mapping and conversion rules to the data.

## Before You Use the LSMW…

The LSMW is a nice, easy-to-use tool. However, to successfully migrate data to an SAP system, you first need all of the following:

- ☑ **A data migration concept:** You must define the business objects to be imported electronically or manually, which import technology to use for each business object, the migration schedule, any dependent business objects, and the like.

  Identifying the import technology to use is a process that requires some analysis, since you can only use certain import technologies to import data into certain object types. In a later section, I provide some insight on how to perform this analysis (see "Analyzing Input Data for an Object Type").

- ☑ **Knowledge of data migration principles:** You have to know how to set up a data migration project, how to analyze data, and so forth.

- ☑ **Proper customizing of the associated business objects in the SAP system:** Without proper customizing, you won't be able to get data into the system, or data will not be consistent with the associated business processes.

- ☑ **Proper sizing of the system landscape:** The system has to perform well enough to meet the schedule of the implementation project. For example, if you are migrating production data before going live, you may have strict time limits to which you must adhere. Ask your system administrator how the system will perform with the planned amount of data, parallel data migrations, and the time schedule by showing the results of test runs (see "Analyzing Input Data for an Object Type").

7. Specify the input files.

8. Migrate the data.

For our example project, let's assume that we've already identified the SAP business object we require, that we have chosen the BAPI import technology, and that we have analyzed the data structure required by the BAPI that corresponds to the object type. (See the sidebar above for some of the advance tasks you should also keep in mind.) We will create an LSMW object for migrating business objects of type "Characteristics" from two sample files shown in **Figure 5** (*CharacteristicsMain.txt* and *CharacteristicsDescr.tx*t). These files will be migrated from a frontend PC to an SAP system, using the standard BAPI Characteristic.Create.

*Figure 5 Sample Files to Be Migrated*

CharacteristicsMain.txt

| | A | B | C | D |
|---|---|---|---|---|
| 1 | INDEX | CHARACT_NAME | DATA_TYPE | LENGTH |
| 2 | 1 | LSMW_CHARACTERISTIC1 | C | 7 |
| 3 | 2 | LSMW_CHARACTERISTIC2 | C | 8 |
| 4 | 3 | LSMW_CHARACTERISTIC3 | N | 9 |
| 5 | | | | |

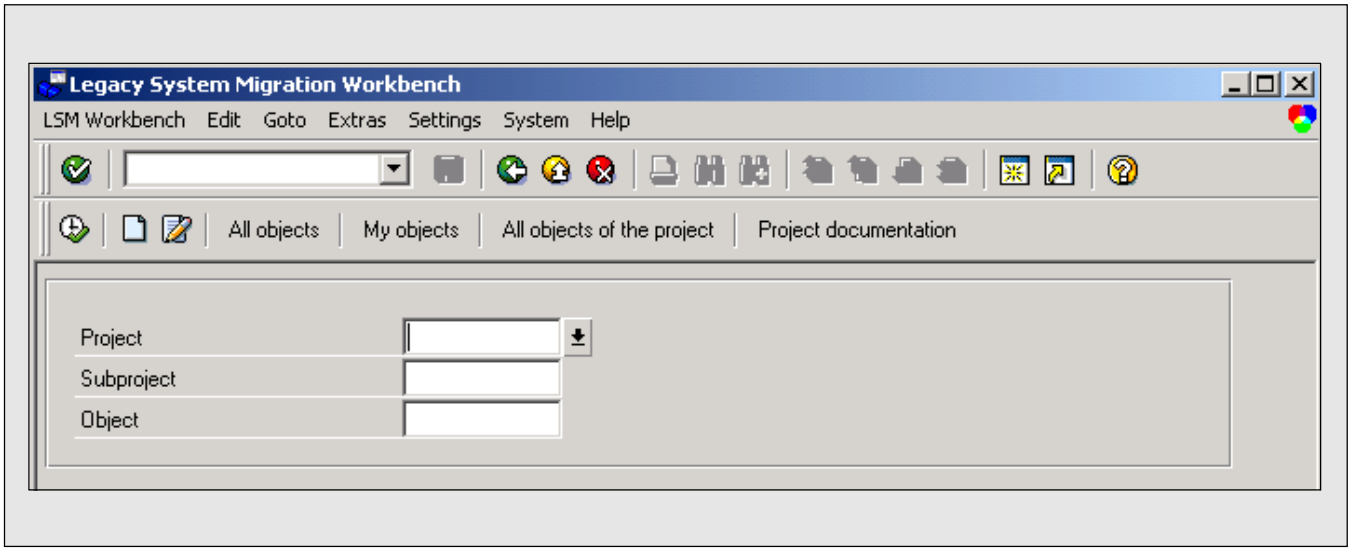
CharacteristicsMain

CharacteristicsDescr.txt

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | INDEX | LANGUAGE_ISO | DESCRIPTION | | |
| 2 | 1 | EN | Characteristic 1 | | |
| 3 | 1 | DE | Merkmal 1 | | |
| 4 | 2 | EN | Characteristic 2 | | |
| 5 | 2 | DE | Merkmal 2 | | |
| 6 | 3 | EN | Characteristic 3 | | |
| 7 | 3 | DE | Merkmal 3 | | |
| 8 | | | | | |

CharacteristicsDescr

*Figure 6* *The LSMW Initial Screen*




The Characteristic.Create BAPI has been available since R/3 Release 4.6A, so this example can be used for all R/3 releases from 4.6A on with LSMW version 1.72. It should also work on R/3 Enterprise and CRM 3.0 with LSMW version 3.x.

Before beginning this exercise, create the two sample files with Microsoft Excel: Start a new spreadsheet and create content similar to what you see in Figure 5 for the first source file, which is *CharacteristicsMain.txt*. Apply changes to the content if desired and then save the file to the temporary directory of your frontend PC as a tab-delimited text file. The file and path name should be something like *C:\temp\CharacteristicsMain.txt*. Follow the same steps for the second file, *CharacteristicsDescr.txt*.

> ✓ ***Note!***
>
> *If you have more than one source file, you must tell the LSMW which rows belong to the same instance of a business object. The LSMW recognizes the first field of a data row as the identifying field. Thus, in our example (Figure 5), the identifying field is in column "INDEX."*

We're now ready to walk through the steps of the migration, so let's get started.

### *Step 1: Call Transaction LSMW*

The first step is simply to call transaction LSMW. You will see a screen like the one shown in **Figure 6**.

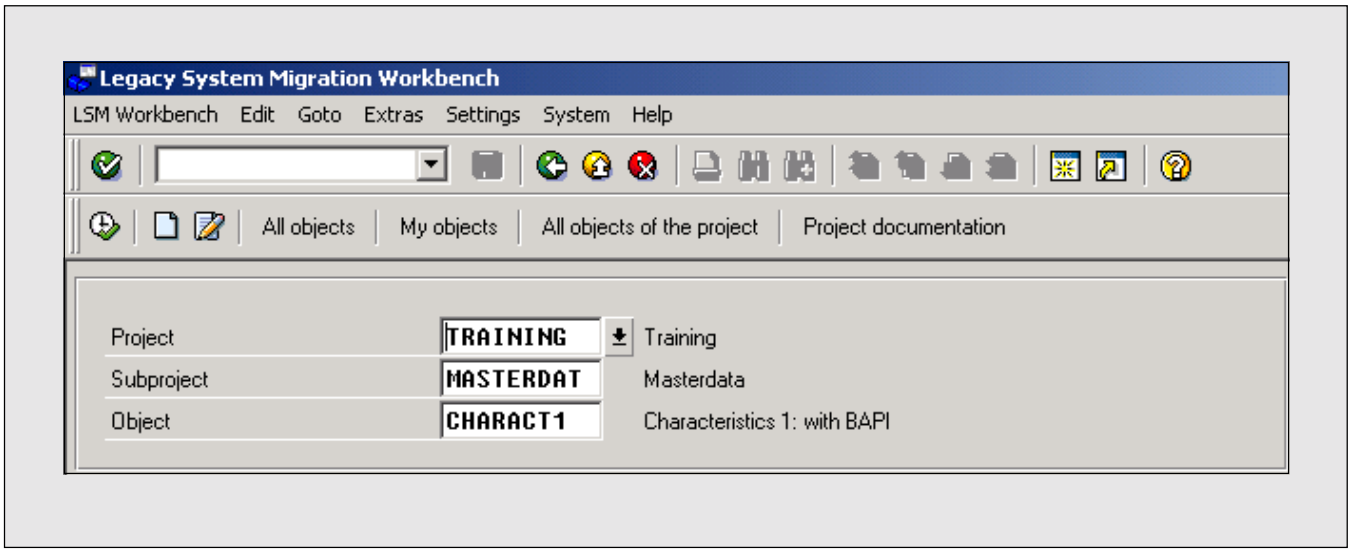
### *Step 2: Create the LSMW Hierarchy*

Within the LSMW, data migration projects are organized in a hierarchy with *project* as the top level, *subproject* as the middle level, and *object* as the bottom level. For example, you may have projects like "Control Data," "Document Data," "Master Data," and the like. These projects are the top level of the hierarchy. Expanding "Master Data," the hierarchy might look like this:

```
Master Data
    FI Master Data
    Logistics Master Data
            Material Master 1
            Material Master 2
            Info Records
            …
```

*Figure 7* *Create a Project Hierarchy*




where "Master Data" contains the subprojects "FI Master Data" and "Logistics Master Data." The subproject "Logistics Master Data" is the parent of the objects "Material Master 1," "Material Master 2," and so on.

For our example, name the project "TRAINING," the subproject "MASTERDAT," and the object "CHARACT1" (or use any other names you want). Place the cursor in the field labeled "Project" (to make the field active) and click the "Create" button (📄) or press Shift+F1. Describe each hierarchy element with any text you like. The result will be similar to what you see in **Figure 7**.

The LSMW object CHARACT1 is now created. It is possible, and useful, to enter some documentation for each hierarchy element. Click the "Project documentation" button in the application toolbar and all elements of the project will be displayed. The best way to document the hierarchy elements is to double-click on each one in turn and add a description to the text area.

### *Step 3: Define IDoc Inbound Processing*

Because data migration with BAPIs utilizes the ALE interface, and because IDoc inbound processing is part of the ALE interface, we have to set up IDoc inbound processing for our chosen BAPI.

> ✓ ***Note!***
>
> *Definitions for IDoc inbound processing are dependent on the particular LSMW project, so the ALE customizing for different LSMW projects can differ.*

Follow these steps:

1. Back on the LSMW initial screen, go to *Settings → IDoc inbound processing* (or press Ctrl+Shift+F1). Verify that you are defining the settings for the right project. Usually there is more than one LSMW project on a system, and you have to choose the right one before customizing the IDoc inbound process.

2. Click "Maintain ports," which will take you to transaction WE21 (Define Port).

3. Create a file port by placing your cursor on the

"File" folder and clicking "Create" (). Name the port "LSMW" and enter a description.

4. Choose the "IDoc record types SAP Release 4.x" radio button under "Version."

5. Now you have to specify dummy outbound and inbound file descriptions. Practically speaking, you'll never use the file descriptions specified here because the LSMW uses its own file-handling mechanism, but the WE21 customizing transaction requires a file description for "normal" ALE file handling, such as file communications with EDI subsystems. So, to specify the dummy file descriptions:

   - On the tab "Outbound file," leave the "Physical directory" selection as is (this is usually the default directory *\usr\sap\<system>\SYS\global*) and name the outbound file "LSMWfile" (if you have more than one application server, talk to your system administrator about using logical directories).
   - Test the file system by clicking the Access test button.
   - Do the same on the "Inbound file" tab.

     If the default directory is different from the one listed above and can't be accessed, try *\usr\sap\<system>\<SYS>\global* (where *<SYS>* is the three-digit system name). If this doesn't help, ask your system administrator for a transfer directory.

6. Go back to the "IDoc inbound processing" screen and enter your newly created file port in the "File port" field.

7. For this example you can leave field "tRFC port" empty. This port is used for IDocs that are transferred directly to the function module for ALE/tRFC inbound processing, meaning that no conversion file will be created.

8. Before you can proceed, you must open a new mode and maintain the view V_TBDLS in transaction SM30. Create a new entry for logical system LSMW, including a description, and save it.

9. Go back to the "IDoc inbound processing" definitions. Enter "LS" (logical system) in the field "Partn.type" and then click "Maintain Partner Numbers." This will take you to transaction WE20 (Partner Profiles).

   Generally, the partner profile describes the inbound and outbound messages for a given partner. In ALE, this partner is a logical system (e.g., a client on another R/3 system), which is the communication partner to exchange IDocs with. In the LSMW, the partner profile is used to store the inbound messages. The LSMW will create the message-specific entries automatically when you define your LSMW object attributes with the BAPI or IDoc import technology (more on this in "Step 4: Define the Object Attributes").

10. Place your cursor on the "Partner type LS" folder, click "Create," and enter "LSMW" in field "Partn.number." On the tab "Post processing: permitted agents," type "US" in field "Typ" and your user name in field "Agent." Save the folder, navigate back to the "IDoc inbound processing" screen, and enter "LSMW" as the partner number.

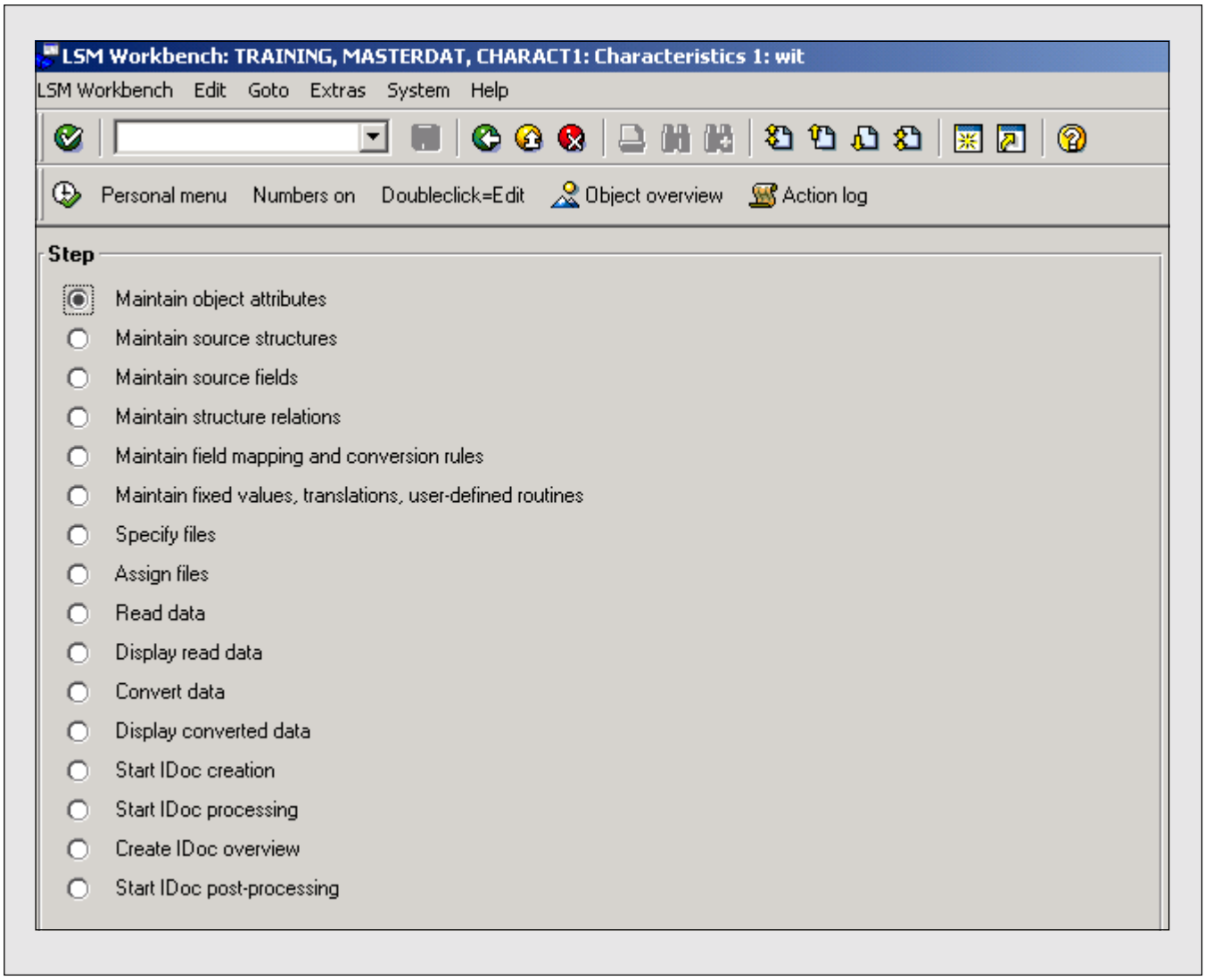
11. To be sure that your system is prepared for ALE, perform the following two tasks[2]:

    - Activate the IDoc inbound processing. This task automatically customizes the workflow event linkage for ALE inbound processes.
    - Perform the workflow customizing. It's okay if the customizing of the workflow runtime system and development environment is incomplete and the traffic-light indicators are red. For our purposes, it's only necessary that the following items have green checkmarks next to them: "Workflow administrator maintained," "Workflow RFC destination configured completely,"

[2] If you are not familiar with ALE or SAP Business Workflow, have someone who is familiar with them perform these tasks.

*Figure 8* *The LSMW Project Steps*




"Generic decision task classified completely," and "Sending to objects and HR objects activated."

If one of the listed items is *not* checked, and/or if one of them is named differently (e.g., "Workflow RFC destination does not exist" instead of "Workflow RFC destination configured completely"), then run automatic customizing by clicking the icon that corresponds to the item. This action usually corrects the problem. If you are not sure whether the action was successful, contact someone in your organization who is familiar with ALE or SAP Business Workflow.

12. Return to the main screen where you maintained the LSMW project hierarchy.

### *Step 4: Define the Object Attributes*

Click (or press F8) to access the LSMW's object definition and execution environment. A screen will appear with a list of project "steps" (see **Figure 8**).

### ✓ *Working in the LSMW's Object Definition and Execution Environment*

- *You can customize the list of LSMW project steps shown in Figure 8 via the menu path Extras → Personal Menu (or by pressing Shift+F5).*
- *The LSMW defaults to display mode when you double-click on the project step that you want to work with. This is awkward when you are defining a new object because you have to manually switch from display mode to edit mode for each project step. To make edit mode the default for all project steps, click the "Doubleclick=Display" button (the button text will then change to "Doubleclick=Edit").*
- *When navigating through the screens of the LSMW project steps, such as the "Change object attributes" screen, you are mainly in edit mode, if you set edit mode as the default (see above). If you find at any point that you are in display mode, to switch back click the "Display <-> Change" button, which is available on most screens and is usually also Ctrl+F1.*

Follow these steps to define the LSMW object:

1. On the main project step screen (shown in Figure 8), double-click on project step "Maintain object attributes." On the resulting screen, leave the radio button "Data transfer" set to "once" and do not set the flag for system-dependent file names.

2. Choose "Business Object Method (BAPI)" as the import technology, press F4 (Possible Entries Help) in the field "Business object," and choose the business object for "Characteristics" (which is "BUS1088") from the pop-up list that is displayed.

3. Specify the import method by pressing F4 in the "Method" field. As there is only one method, "CREATE," it will be chosen automatically. Usually there is only one method available. If more than one method is available, however, you must be sure to choose the right one. To find out which is the right method, read the documentation for the BAPI you are using.

   The ALE interface requires a message type and a "basic" IDoc type. When the method has been defined, the system will automatically suggest the corresponding message and basic IDoc type.

4. Save your settings. You will get an information message telling you that a partner agreement for message type "CHARACTERISTIC_CREATE" has been created.

   The partner agreement is an ALE customizing setting that defines data exchange partners with their associated messages. When you use the LSMW, this customizing setting is created for you automatically. If desired, you can check the settings in transaction WE20 in the inbound parameters of logical system LSMW.

5. Return to the list of project steps (shown in Figure 8).

### *Step 5: Define the Source Data*

Source data consists of a source structure and source fields that will be mapped against the target structure and target fields. The source *structure* shouldn't be confused with the external input *file* — the source *structure* is the logical representation of the data in the source *file*.

In a later step ("Step 7: Specify the Input Files"), we will assign the external source file(s) to the source structure(s).

To define the source data structure:

*Figure 9* *Create the Source Structure*

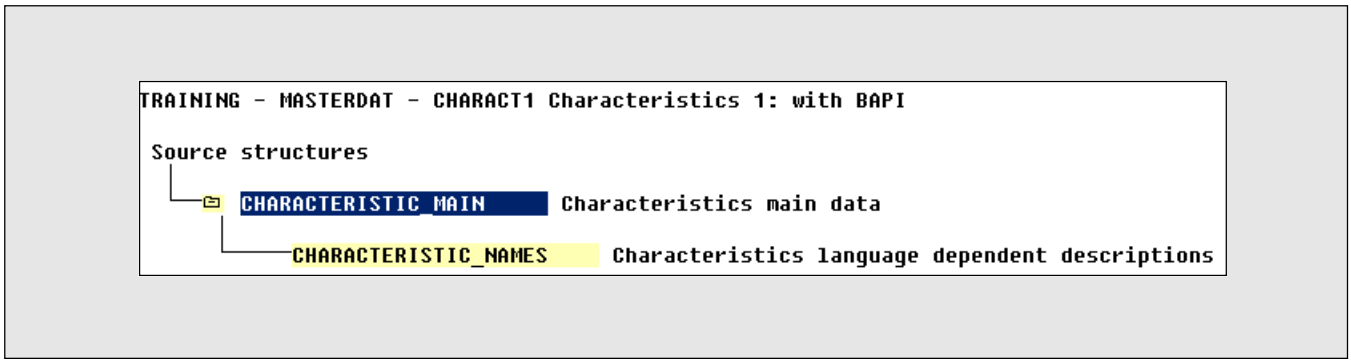
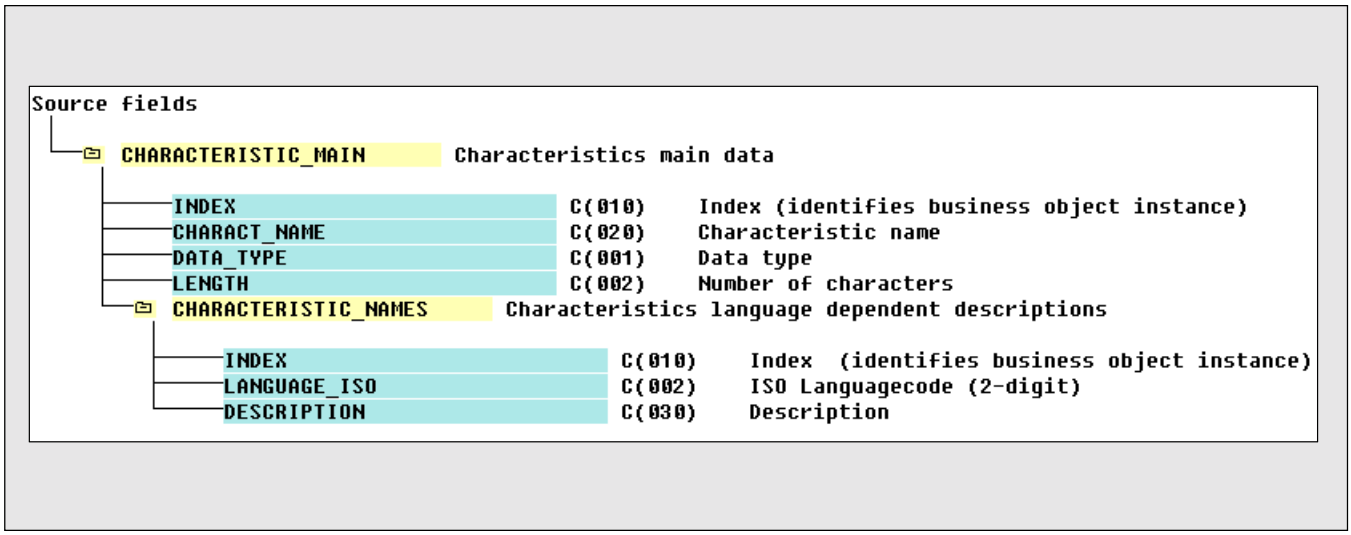



*Figure 10* *Define the Source Fields*




1. Double-click on project step “Maintain source structures.” On the resulting screen, click the new structure icon ( ). Enter a name for the source structure (“CHARACTERISTIC_MAIN”) and add a description (e.g., “Characteristics main data”).

2. Place the cursor on your newly defined source structure and create another source structure (“CHARACTERISTIC_NAMES”) that is sub-ordinate to the previous one. Add a description (e.g., “Characteristics language dependent descriptions”). The final source structure should look like the one shown in **Figure 9**.

3. Save what you have done so far and return to the list of project steps (Figure 8).

4. Now define the fields for the source structure. Double-click on “Maintain source fields.” On the resulting screen, place your cursor on the first source structure and click “Table maintenance” ( ) or press Ctrl+F9. **Figure 10** shows how the field definitions in our example should look.

There are additional functions — e.g., "Create single field" ( ) or "Copy fields" ( ) — for specifying source fields, but the "Table maintenance" function used in this example is the most convenient way to manually enter the fields.

> ✓ ***Note!***
>
> *To simplify the mapping definitions (more on this in the next step), use the same field names in the source structure and the target structure.*

> ***Warning!***
>
> *Do not change the field names of the source structures after you've applied mapping and conversion rules. The LSMW remembers the "old" field names, so changing them later may result in strange error messages.*

### *Step 6: Apply Mapping and Conversion Rules to the Data*

It's time to apply mapping and conversion rules to the data. There are three tasks involved with this step:

- Define the relationship between the source structure and the target structure.
- Map the source fields to the target fields.
- Assign the conversion rules.

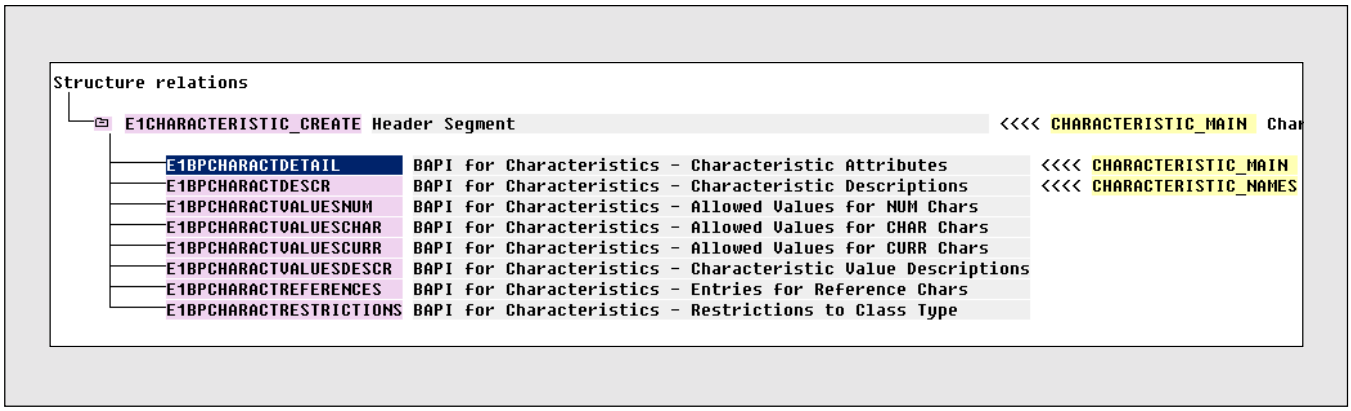
Let's take a closer look at the individual steps involved in each of these three tasks.

To *define the relationship between the source structure and the target structure* (i.e., the IDoc structure), follow these steps:

1. Return to the list of project steps and double-click on "Maintain structure relations." The resulting screen will display the target structure.
2. Place your cursor on the first structure, E1CHARACTERISTIC_CREATE, click the Relationship button (or press Ctrl+F4), and select the source structure CHARACTERISTIC_MAIN from the pop-up list that appears.

Perform these steps for target structures E1BPCHARACTDETAIL and E1BPCHARACTDESCR. The source structure for the former is CHARACTERISTIC_MAIN and the source structure for the latter is CHARACTERISTIC_NAMES. Your result should look like **Figure 11**.

*Figure 11 Define the Relationship Between the Source and Target Structures*



> ✓ ***Note!***
>
> *Once assigned, all source fields of a structure will be available to succeeding target structures. For example, the field CHARACTERISTIC_MAIN-DATA_TYPE can be used for data mapping to target structure E1BPCHARACTDESCR.*

Once you've defined the structural relationship, the source and target data must then be mapped. In the example here, we will map the fields of the source structure to the corresponding target fields of the BAPI's IDoc structure. Remember that in the previous step ("Step 5: Define the Source Data") we defined the fields of the source structures with the same names as the ones in the target structures, which allows us to use the LSMW mapping feature "Auto-Fieldmapping."

So, follow these steps to *map the source fields to the target fields*:

1. Double-click on project step "Maintain field mapping and conversion rules." On the resulting screen, select *Extras* → *Auto-Fieldmapping* (or press Ctrl+Shift+F3).

2. Confirm the defaults of the resulting pop-up.

3. Accept the automatic mapping proposals.

4. Almost all default fields of the target structure are now filled. If you have many 1:1 relationships of source to target fields, you can label the source fields with the same names to increase the success rate of the Auto-Fieldmapping feature.

Now that the automatic mapping of the source fields to the target fields is complete, you must next *assign the conversion rules* to convert data between the source and target structures. Conversion rules comprise constants, translation tables, user-defined routines, and so forth. To explore all available types of conversion rules, double-click on project step "Maintain field mapping and conversion rules" and then click the "Rule" button, which pops up a list of available conversion rules. In our example, we will use the *Constant*, *Translation*, and *ABAP Coding* conversion rules.

Some target fields have to be filled with fixed values, which we will do by assigning a *constant* to the relevant fields:

1. Place the cursor on the target field and click the "Constant" button (or press Ctrl+F9).

2. Assign a constant value of "1" to E1BPCHARACTDETAIL-STATUS and a constant value of "S" to E1BPCHARACTDETAIL-VALUE_ASSIGNMENT.

In our example source structure CHARACTERISTIC_MAIN, we defined field DATA_TYPE as a one-digit representation for data types (e.g., "C" for character). The target field E1BPCHARACTDETAIL-DATA_TYPE is a four-digit representation (e.g., "CHAR"). For *translation* of the value from the source field to the target field, we must first define a translation table as follows:

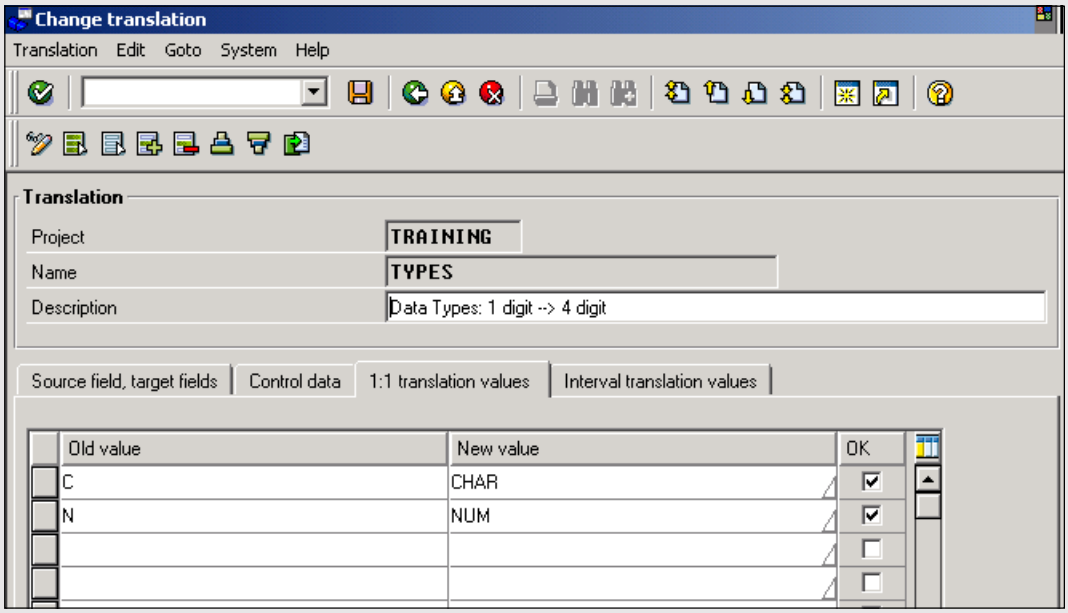
1. Double-click on project step "Maintain fixed values, translations, user-defined routines." On the resulting screen, position the cursor on "Translations" and click the "Create" button (🗋) or press Ctrl+F1.

2. Name the new translation "TYPES," give it a short description, and confirm your translation definition.

3. Expand the "Translations" node and double-click on the newly created translation ("TYPES" in the example here).

4. Define the source field with length "1" and type "C," and the target field with output length "4" and dictionary type "CHAR."

*Figure 12 Enter Values in the Translation Table*




5. Save your settings and go to the "1:1 translation values" tab, shown in **Figure 12**.

6. Enter values in the translation table as shown in Figure 12. The "OK" flag tells the LSMW that the rule is valid, so be sure to select it.

Finally, to apply the translation table as a conversion rule:

7. Return to project step "Maintain field mapping and conversion rules" and place the cursor on field E1BPCHARACTDETAIL-DATA_TYPE (which, as you'll recall, already has a relation to source field CHARACTERISTIC_MAIN-DATA_TYPE due to the Auto-Fieldmapping feature we used).

8. Click the "Translation" button in the application toolbar, which launches a pop-up with radio buttons for different translation options. Press F4 on the upper field of the pop-up for a list of values for translation rules.

9. Choose the rule you've just created ("TYPES" in the example). Make sure the selected translation option is your rule and confirm the settings.

10. To call the translation conversion rule, the LSMW will now insert ABAP code into the target data field that looks like what you see in **Figure 13**.

Now we want to fill the target field E2BPCHARACTDESCR000-LANGUAGE_INT (the one-digit SAP internal language character field) with data from the source field CHARACTERISTIC_NAMES_LANGUAGE_ISO (the two-digit ISO language code) by converting the source field data via custom *ABAP coding*.[3]

[3] Perhaps you have noticed that SAP systems (from Release 4.0 on) display all language codes as a two-digit language key (similar to the ISO representation of the language code) despite storing the information (usually) as a one-digit character. SAP systems accomplish this trick by applying a conversion routine, which is associated with the underlying domain of the field.

*Figure 13 ABAP Code That Calls the Translation Conversion Rule "TYPES"*

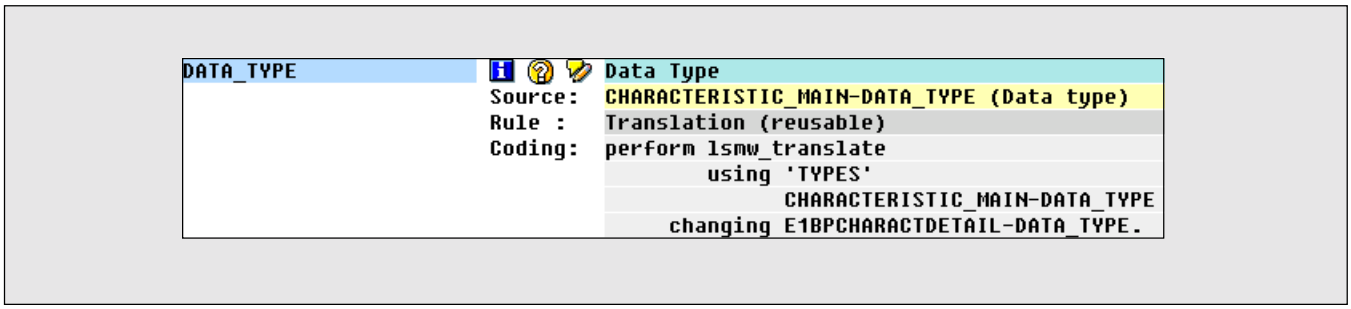
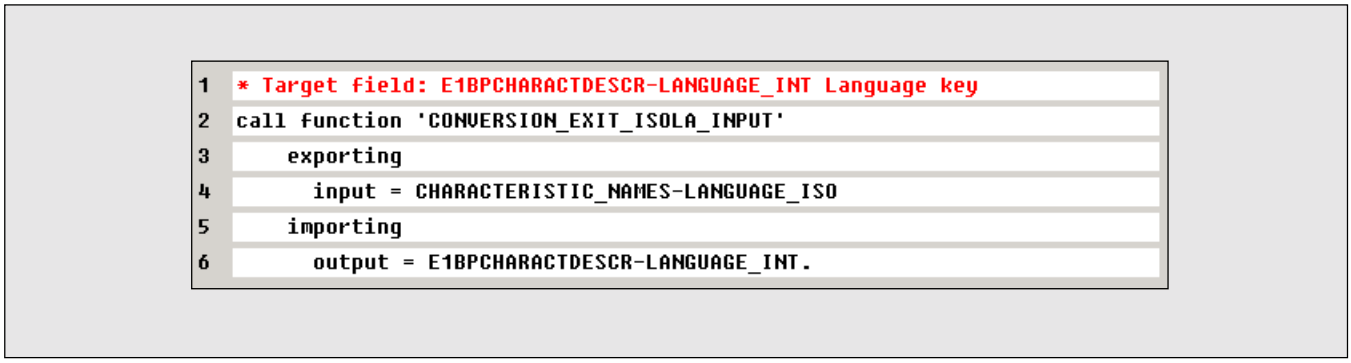



*Figure 14 Insert Custom ABAP Coding into the Conversion Rule*




> ✓ ***Note!***
>
> *Applying the "ABAP Coding" conversion rule is not actually necessary for this example because we already delivered the language information to the BAPI via an ISO language key when we used the Auto-Fieldmapping feature. Therefore, if you don't want to do any ABAP coding, you can skip this step. I am including it here because it is a useful example of how to insert ABAP coding into a conversion rule and how to utilize SAP standard conversion routines for input fields.*

Each ABAP conversion routine is represented by an input and an output function module. The output function module converts the database field to a display field, while the input function module does the opposite (converts the display field to a database field). The nomenclature for the function modules is *CONVERSION_EXIT_< CONVERSION ROUTINE>_OUTPUT* and *CONVERSION_EXIT_ <CONVERSION ROUTINE>_INPUT*, respectively.

To use the input function module to convert the source file's two-digit ISO language code into the single-digit SAP internal language code:

1. In project step "Maintain field mapping and conversion rules," double-click on field LANGUAGE_INT; you will get one line of code in the ABAP Editor.

2. Insert five more lines after the first line, which contains a comment, and copy the code shown in **Figure 14** into the empty lines.

3. Click the "Check" button ( ) to verify that there are no errors (you should get the message "No syntax error found") and save your coding.

### ✓ *Tip*

*Check the ABAP coding each time you create or change any coding. The check routine will be applied to the actual coding context, which involves user-defined routines as well. If a check of the coding associated with a target field reveals an error in a user-defined routine, the check will pop up a message, but it won't tell you that the error lies in the user-defined routine. If an error is displayed that you can't locate, go to your user-defined routines and check their coding.*

*Figure 15 Defining the Specifications for the CharacteristicsMain.txt Input File*

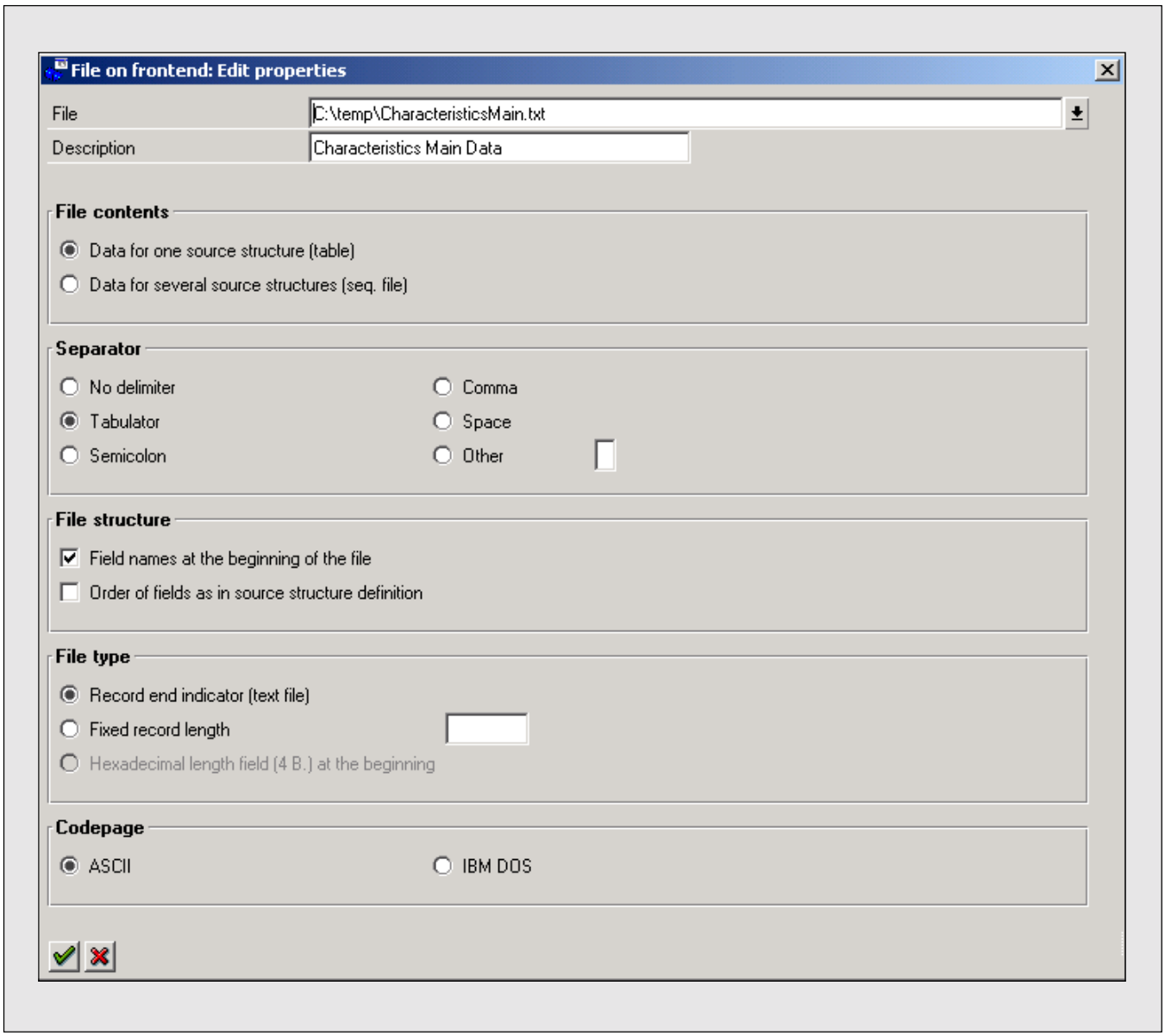
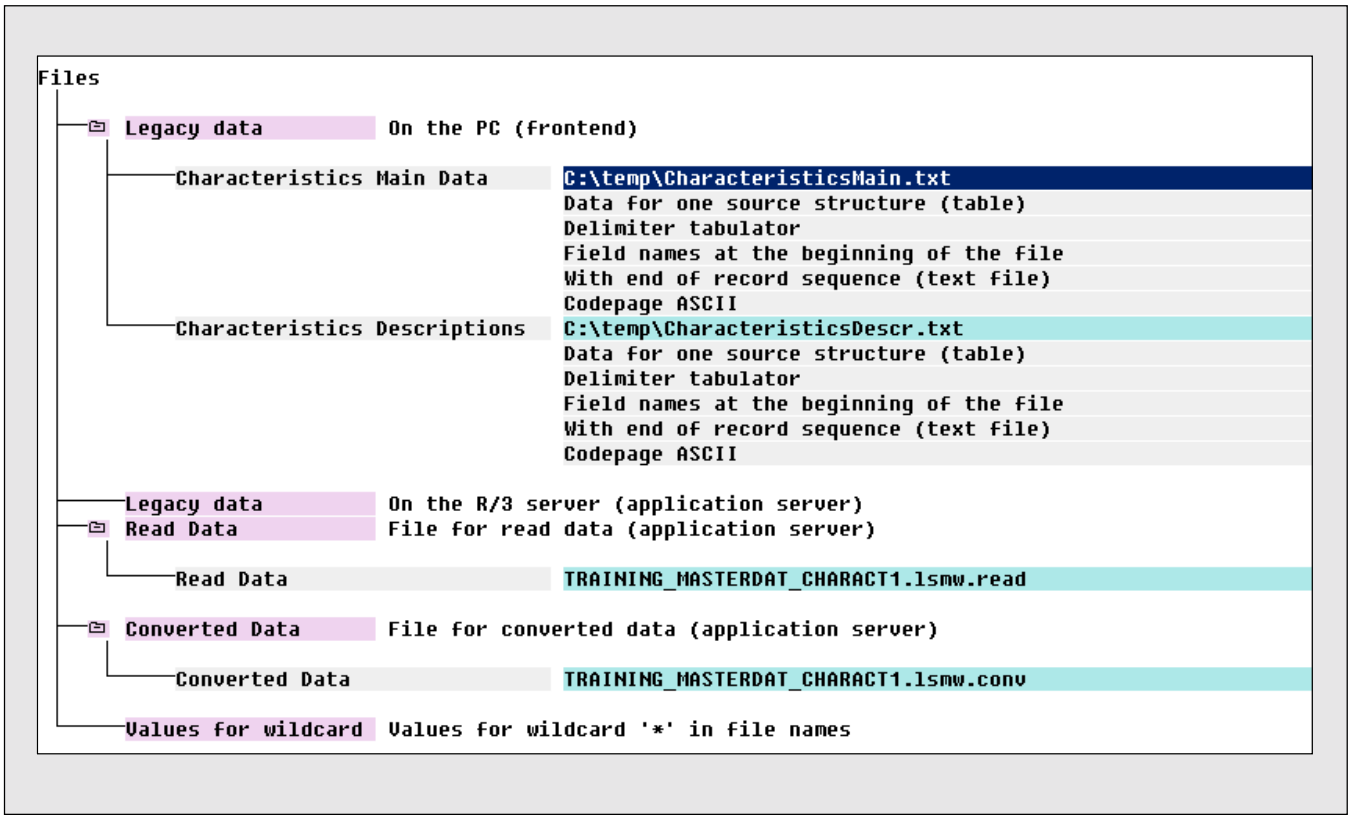


*Figure 16 The Resulting Input File Specifications*




### *Step 7: Specify the Input Files*

The last bit of work to be done in terms of creating the LSMW object is to define the external file and relate it to the logical source structure.

Follow these steps:

1. To define the external file, double-click on project step "Specify files." On the resulting screen, you will see that the LSMW has already generated file names for the read data and converted data.

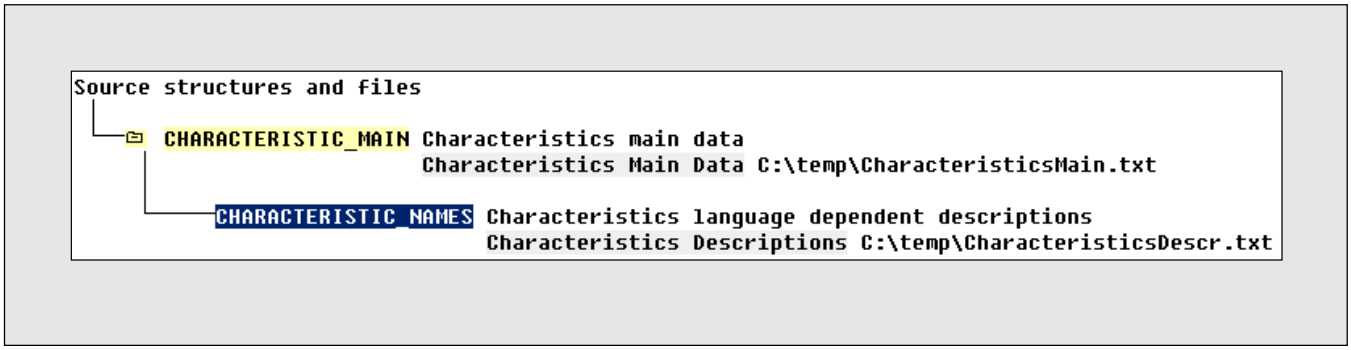
2. Place the cursor on the "Legacy data" label, which is on the PC (frontend), and click or press Ctrl+F2; you will get a pop-up, like the one in **Figure 15**, that lets you define the file specifications.

   Remember that when we created the external input files before beginning the data migration steps, we saved the Excel spreadsheets as tab-delimited text files called *CharacteristicsMain.txt* and *CharacteristicsDescr.txt*, so for *CharacteristicsMain.txt* the specifications should look like Figure 15. You will need to adapt the specifications for the "File" field according to your PC's file structure and the actual file name.

3. Create one entry for each source structure (that is, one file for CHARACTERISTICS_MAIN and one for CHARACTERISTICS_DESCR). In our example, the files are *C:\temp\CharacteristicsMain.txt* and *C:\temp\CharacteristicsDescr.txt* (**Figure 16**).

4. To relate the physical input files to the logical input structure, double-click on project step "Assign files." On the resulting screen, place the cursor on the structure you want to assign.

*Figure 17 Assignment of Source Structures to Input Files*




5. Click the Assignment button and select the corresponding input file. Do this for both structures. You should see an assignment similar to **Figure 17**.

Congratulations! You've created the LSMW project hierarchy. Now you can start importing the data.

### *Step 8: Migrate the Data*

In this step, we first upload the external input data into a read file, next convert the data, then create the transfer IDocs (one for each instance of the business object), and finally post the IDocs in the target system.

Follow these steps:

1. Double-click on project step "Read data." On the resulting screen, change the parameters if necessary and click "Execute" (), or press F8. Data will be uploaded from your PC to the read file and the status of the action will be displayed.

2. View the read file by double-clicking on project step "Display read data." Note that the lines are sorted by the instances of the business object.

3. With the BAPI import technology, you can convert data and create the transfer IDocs in one step. Double-click on project step "Convert data." Set the radio button to "Create IDocs directly" (instead of "Create file") and confirm (F8). The LSMW now applies data mapping and conversions to the read data and generates one IDoc for each instance of the business object.

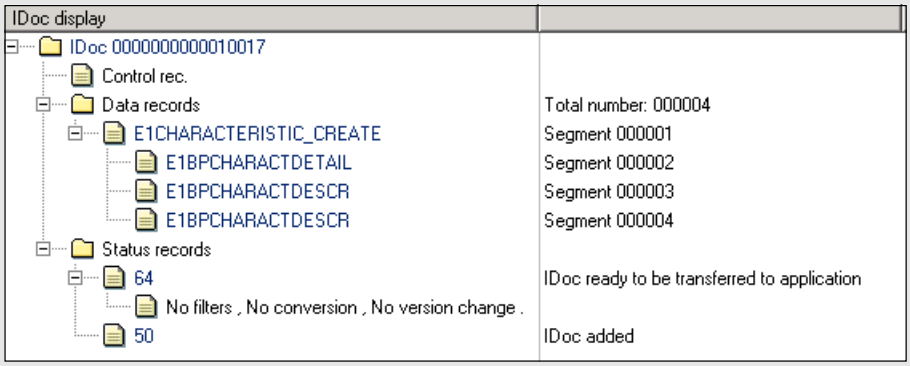
> ✓ ***Note!***
>
> *You cannot use project step "Display converted data" to view the converted data file because a converted data file doesn't exist — we have produced the IDocs directly, without creating a conversion file. Remember that the output file defined in transaction WE21 ("Step 3: Define IDoc Inbound Processing") is a dummy; even though it won't be used with the LSMW, transaction WE21 requires an entry for it. However, you can view the IDocs in project step "Create IDoc overview" or with transaction WE02 by selecting logical message CHARACTERISTIC_CREATE. Note that project step "Start IDoc creation" is obsolete since we created the IDoc directly.*

4. The IDocs have a status of "64," which means they are ready to be posted by the application-specific program — in our example, the BAPI Characteristic.Create. (See the sidebar on the next page for details on how IDocs are structured in the source system database.) To post the IDocs, double-click on project step "Start IDoc processing."

## The Structure of IDocs in the Database

An IDoc in the database consists of exactly one control record, one or more data records, and at least one status record. The IDoc is identified by its IDoc number. The screenshot below shows an example of IDoc CHARACTERISTIC_CREATE01 posted to the database.



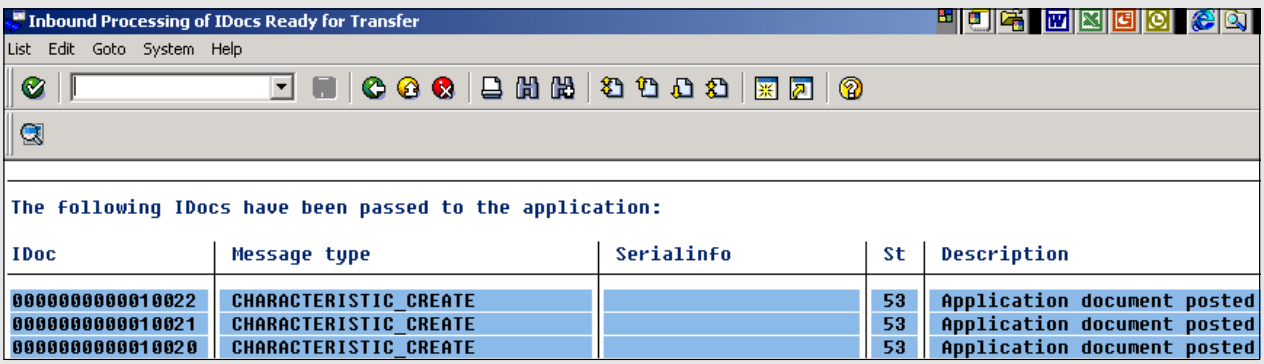

(continued on next page)

5. Confirm (F8) the selection screen, and you will immediately get the results displayed as a list, similar to the one shown in **Figure 18**.

If the application was able to post the IDoc data, the IDocs have a status of "53." To view the characteristics you've just created, go to transaction CT04.

If there are any posting error(s) — in other words, if any IDoc has a status other than 53 — check the status folder to see if it contains any information about the error. Navigate to the IDoc by double-clicking on the corresponding line in the display and selecting the status folder. Look for the status folder highlighted in red.

*Figure 18* *The IDoc Processing Results List*



(continued from previous page)

The control record stores information about the technical partner of the data exchange (sender/receiver), structural information about the data records (IDoc type, message type, and so on), and the processing information (timestamps, actual status, and the like).

The data records are organized in a tree structure of data segments, which might resemble the data model of the business object. The tree structure of the data records, with its corresponding fields, is called the *IDoc structure*. To browse through the IDoc structure of this IDoc, call transaction WE60 (Documentation for IDoc Types), type "CHARACTERISTIC_CREATE01" in input field "Obj.type," and click "Display tree."

Status records supply processing information and are added to the IDoc in the database in each processing step. In the screenshot on the previous page, you can see that the IDoc has been posted to the database (Status 50) and is ready to be transferred to the application (Status 64).

✓ ***Note!***

*The status information comes from the import program. If the import program is a BAPI (as it is in our example), the status is read from the BAPI's RETURN parameter. Sometimes this information is useful, and sometimes it is less than useful. See the section "Tips and Tricks" on page 84 for advice on other ways to investigate the error.*

## Data Migration with the IDoc Import Technology

In our example, we used a BAPI for inbound processing, which used an IDoc as a data container. Therefore, we had to begin by customizing the IDoc inbound processing tables.

If you want to use the IDoc import technology instead of the BAPI import technology, you need this same customizing (which supplies additional migration modules). To migrate data using the IDoc import technology, follow the same eight steps outlined for our BAPI-based example data migration project.

## Monitoring Ongoing Transfers

As soon as an IDoc has been created, you can examine it with standard SAP monitoring tools. The most important monitoring tools in this context are WE02 (Display IDoc), WE05 (IDoc List), and BD87 (Status Monitor for ALE Messages):

- **WE02/WE05:** These tools list the selected IDocs and allow you to view and change the content of a single IDoc's data record. The difference between WE02 and WE05 is in how they display the selected IDocs.
  - By default WE02 (Display IDoc) lists every single IDoc that meets the selection criteria. In a navigation tree, you can set the focus on inbound or outbound IDocs and on the message type.
  - WE05 (IDoc List) groups the selected IDocs according to their direction (inbound or outbound) and status. From this view, you can drill down to a single IDoc.

- **BD87:** This tool gives you a view of the selected IDocs that is similar to the view provided by WE05. In addition, it offers you some post-processing options and options for tracing IDocs (these functions are not relevant in an LSMW migration, however).

When using the BAPI and IDoc import technologies, the LSMW provides project step "Create IDoc overview" to view the IDocs. This project step calls the same ABAP report as transaction WE05 and submits the selection criteria that apply to the IDoc data generated by your last LSMW data migration.

## Analyzing Input Data for an Object Type

For the example in this article, we did not have to worry about supplying the BAPI with data. Normally, however, you begin a data migration project with a given business object, which should support a defined business process, and you have to search for an object type that fits the business object. You might even find more than one object type that works. Regardless, once you know the object type (or types), you will know which import technologies are available to you, and you will need to determine the input data required for that type. To this end, you might compare the behavior of the object types by testing them (with the LSMW and/or the DX Workbench) to see, for example, how fast they can be processed, the availability of appropriate input data structures, and the results they yield.

The main challenge is to find out how to satisfy the object type with data. This means you must analyze the input data of the business object, which is determined by the information needs of the business process and the system behavior, and is controlled by customizing. You need to find out:

- Which import technology to use
- The mandatory target fields of the business object
- Other target fields that are needed to support the business processes
- The target structures
- What information is required from the source data
- What the import structure looks like

You can analyze input data with the help of the LSMW and DX Workbench and its associated reports. To analyze the data using these tools, perform the following steps:

1. Create the business object manually with the corresponding online transaction (e.g., transaction MM02 for Material Master).
2. Generate sample data with the DX Workbench, if the generation report is available (see the "Download Files" section at **www.SAPpro.com** for more details on the DX Workbench).
3. In the LSMW, create an LSMW object, define the object attributes, create a source structure with any name you want, and add one field to the source structure.
4. In the LSMW, use project step "Maintain structure relations" to assign the source structure to all mandatory target structures and to all target structures that have values in the generated sample data. The purpose of this action is to print out all target structures and fields, which is only possible if the target structures are assigned to a source structure. If there is no report available for generating sample data, you will have to perform a manual analysis to determine which additional target structures you might need.

> ✓ ***Tip***
>
> *Sometimes you can find the mandatory structures and fields for BAPIs in the BAPI documentation. In transaction BAPI, select the BAPI you are using and go to the "Documentation" folder.*

5. In the LSMW, go to project step "Maintain field mapping and conversion rules" and print the initial mapping and conversion rules, or save the list into a local file. This list will help you to define data mapping and conversion rules because this is the target structure to which you have to refer.

6. The sample data generated from the DX Workbench tells you which target fields of the target structure you have to fill with values. For each field that has a value in the generated sample data, in the LSMW define the source (e.g., a field, a constant, a 1:1 translation table, etc.). If you don't have sample data, you will have to refer to the BAPI documentation (transaction BAPI) or go through a trial-and-error process to learn which target fields to supply with input data.

Analysis of the target information is now complete, and you can define the source structure in the LSMW.

## Tips and Tricks

☑ **Check the read file and the converted file when analyzing errors.** When analyzing errors (see "Step 8: Migrate the Data"), check the read file and the converted file or IDocs if you find that the information provided by the posting application is not clear to you. Fields might not have been loaded the way you thought they would be, data conversions might be behaving differently than intended, or you might have just forgotten to implement a rule.

☑ **Test business objects using sample files.** To test your newly defined LSMW project, create a sample file (or files) with data for two or three instances of the business object. Error analysis will be much faster and easier if you do this.

☑ **Start a data migration project with a representative amount of "live" import data in an integration environment.** A representative amount of import data is an amount that allows you to calculate the real load of the production data migration. If you have large amounts of data with dependent business objects (for example, sales orders that depend on a customer master), it's very important that you calculate, or prove, the time required for the planned production import — keeping in mind that the test environment might perform differently than the production environment and that other data migrations might be running at the same time. If performance is critical, make sure that no other programs — including other data migration runs — will occupy system resources during the data migration run. Another advantage of this kind of test is that it helps you to learn about the quality of source data and SAP customizing, because the data will be validated against the customizing of the SAP system.

☑ **Use the "Table maintenance" feature for quicker source field definition.** When defining source fields, use the "Table maintenance" feature instead of the "Create field" feature.

☑ **Use the confirmation option with the "Auto-Fieldmapping" feature.** "Auto-Fieldmapping" (*Extras → Auto-Fieldmapping*) supports data mapping definitions, but the system can only compare the field names, not verify the context. Therefore, I recommend leaving the confirmation option switched on when running the "Auto-Fieldmapping" function so that you are prompted to confirm each mapping rule before the system applies it to the target field.

☑ **Know your BAPI.** Read the BAPI documentation in transaction BAPI to learn about the properties of the BAPI you are using. You can also navigate to the IDoc interface via the "Detail" tab. Double-click on field "ALE message type" (which will take you to the Generate ALE Interface for BAPI transaction) and then click the "Display interface" button. You will see the IDoc structure of the BAPI. If there is no entry in field "ALE message type," it means that this BAPI has not been enabled for mass data transfer.

☑ **Consider archiving the IDocs.** When using the BAPI or IDoc import technologies, all of the

## Additional Resources

**Legacy System Migration Workbench (LSMW)**

The media library at the SAP Service Marketplace (**http://service.sap.com/LSMW/**) supplies a set of helpful documents on the LSMW:

- A must-read is the LSMW's "Quick Reference Guide." This Microsoft Word document is available for download and contains very good documentation on the LSMW. It supplies more details than the online documentation (which also can be downloaded from the same site).
- The "Installation Guide" also has some useful information for SAP administrators who have to install the LSMW.

**Data Transfer Workbench (DX Workbench)**

- The online help at **http://help.sap.com** for the Data Transfer Workbench (*Cross-Application Components → CA Data Transfer → Data Transfer Workbench*) contains some very helpful information, ranging from the general to the very specific, about migrating the data of various business objects, including detailed structure information.
- A download that discusses the DX Workbench further is also available at **www.SAPpro.com** in the "Download Files" section.

**BAPIs for Mass Data Transfer**

- For advanced users, I recommend SAP's online help (*Cross-Application Components → Business Framework Architecture → Enhancements, Modifications, … → BAPIs for Mass Data Transfer*). This library includes information that can help you gain a good understanding of the role and concept of BAPIs in data migrations. It also includes information on how to create and enable BAPIs for data migration.
- Thomas G. Schuessler's *SAP Professional Journal* articles on BAPIs are another good source of information that can help you gain a sound understanding of BAPIs. For a listing of these articles, see the Past Articles Index on pages 137-142 or **www.SAPpro.com**.

migrated data documentation is stored as IDocs, which can be viewed with transaction WE02. A mass data migration can result in a significant increase in the size of the IDoc database tables, so if you do not need the IDocs anymore (e.g., for documentation purposes) think about archiving them.

☑ **Consider using parallel processing.** In project step "Start IDoc processing," the SAP standard report "Inbound Processing of IDocs Ready for Transfer" will be called with the appropriate selection values. This report has another tab called "Parallel.proc." with a flag named "Parallel." If the flag is set, the IDoc packets can be processed in parallel on the application server; otherwise the packets will be processed one by one. This feature can really speed

up the inbound process, but it may take up most of the resources (work processes) available on the system. If you want to make use of the parallel-processing option, ask your system administrator to create a server group that you can use to define valid application servers for running parallel processes. You need to specify the server group when using the "Parallel.proc." feature.[4]

## Conclusion

The LSMW is a powerful tool for importing data into an SAP system. In our example, we focused on the BAPI import technology, but the other import technologies supported by the LSMW are worth further investigation as well.

When it comes to evaluating an import technology for a given business object, there are many factors to consider: processing speed, availability, and the "fit" of the source data structure to the import technology's target structure, to name just a few. For the business object in our example migration, "Characteristics," three different import technologies — BAPIs, batch input, and IDocs — are available, each with its own particular features and target structures. In cases where you have a real choice of import technology, I recommend building a prototype LSMW object with each import technology and then evaluating the results. If it turns out that IDocs or batch input is a better choice for your particular data migration, combine the knowledge you gained in this article with a little help from the LSMW's "Quick Reference Guide" and you should be able to employ these technologies just as well.

*Arthur Wirthensohn is a senior consultant at EDS Switzerland and a member of EDS's international Technical Leadership Network. He has worked as Project Manager and Product Manager in the ERP and IT integration business for many years, mainly in the Retail and Manufacturing & Trade industries. For the past two years, he was Product Manager responsible for Application Service Providing (ASP) for SAP systems. Currently, Arthur is the technical lead of the Enterprise Application Integration (EAI) department, which delivers SAP-related services like ALE/EDI, SAP Business Workflow, Web Application Server, Data Migration Services, and ABAP Programming. He can be reached at arthur.wirthensohn@eds.com.*

[4] For more on parallel processing, see the articles "Speed Up High-Throughput Business Transactions with Parallel Processing — No Programming Required!" and "How to Build Optional Parallel Processing into Your Applications for Increased Throughput and Reduced Processing Time" in the January/February and March/April 2002 issues of this publication.