How To Implement Customizing Conversions in XI

Version 1.00 - July 2006

Applicable Releases:
SAP NetWeaver ‘04
Solution Manager 4.0
©Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, $/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver “How-to” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.
1 Business Scenario ........................................................................................................ 1
  1.1 Organizational Customizing Exchange ................................................................. 1
2 Introduction ................................................................................................................. 2
  2.1 Conversion Implementation Considerations ....................................................... 2
  2.2 Decision Analysis ............................................................................................... 4
  2.3 Value Mapping Concepts .................................................................................... 4
  2.4 Analysis of Value Mapping Implementation ..................................................... 6
  2.5 Solution Manager – XI Value Mapping Objects Equivalence ......................... 8
  2.6 Additional Comments ....................................................................................... 8
3 The Step By Step Solution ....................................................................................... 9
  3.1 Mapping Design Time: Integration Repository ............................................... 9
  3.2 Generating Solution Manager Customizing IDs ............................................ 11
    3.2.1 Obtaining reference objects from the source application ABAP system ... 11
    3.2.2 Solution Manager ...................................................................................... 16
  3.3 Generating Value Mapping Records in XI ....................................................... 27
    3.3.1 Configuration Time: Integration Directory ............................................ 27
    3.3.2 Value Mapping Records Maintenance in XI ......................................... 29
4 Appendix A: – Links to Related Documentation ............................................... 32
1 Business Scenario

An example company “Speed Inc.” needs to exchange information between its systems. Some systems are legacies from the company’s acquisitions, some are custom developments, and others are SAP application systems.

1.1 Organizational Customizing Exchange

This figure illustrates the example system landscape:

Let us suppose that the company needs to replicate financial information and create MySAP Financials accounting documents (SAP R/3 and MySAP ERP CC). Since the company produces and sells cars, motorcycles, and trucks, these documents will contain accounting information for particular business units.

Even though the generic concept of the business unit exists in every system, both the entity and the name vary from system to system:

- Both My SAP ERP and SAP R/3 recognize the concept as **business area**. The data element is coded BUS_AREA and represents a business unit.

- The car production system and sales system deal with cars in their different **centers** (coded CEN001).

- The Trucks and Motorcycles Sales System separates sales orders according to the **sales division** (technical name DIVSN), that is, trucks or motorcycles.
Now, you must implement a conversion method that allows you to translate this organizational concept as required.

2 Introduction

2.1 Conversion Implementation Considerations

Once you have understood the customer requirement, you must decide how to implement the conversion method.

There are several methods, which can be combined to achieve the required functionality. The following table shows each method:

<table>
<thead>
<tr>
<th>Conversion Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Side</td>
<td>The application system converts the information prior to sending it or after receiving it. In general, this is not the preferred method, since the middleware performs the conversion, but if all the information required for the conversion is located in this system, it may make sense to use this method.</td>
</tr>
<tr>
<td></td>
<td>Using IDoc interfaces, you have three tools:</td>
</tr>
</tbody>
</table>
- ISO conversions: Executed automatically for specified entities. Access using transaction SALE.

- Organizational Units conversions: Translate organizational entities to external or global company codes. This applies to company codes and business areas. Access using transaction SALE.

- Conversion rules: Very powerful tool for converting values at record and field level. Access using transaction SALE.

XI Local Cache

You use this method for low volume, constant information. This is the preferred method for customizing and organizational value conversions. It does not apply to attribute searches, only to key conversions.

You have two tools to implement this:
- **XI value mappings**: Provide transactions for definition and configuration.
- **Solution Manager customizing IDs**: Provide a more powerful environment to define the attributes and
| Data Lookup | There is no data replication involved in this method. External sources of information are accessed at runtime to convert the information or to retrieve attributes. This method is more applicable when working with master or transactional data. One important thing to consider is the SLA of the server system. It must not only be active when XI requires a conversion, but the performance must also satisfy the requirements. There are several methods for performing a data lookup:  
- **XI Lookup APIs**: This is the preferred method for JDBC- or RFC-compliant systems. It uses the communication channel and is executed during mapping.  
- **Access underlying Web AS EJBs**: For example, you can access JDBC services available in the SAP Web AS (Java). This is also used inside the mapping program. Note that you must always comply with the J2EE rules when coding inside the mappings.  
- **Implement interfaces with BPM**: It is also possible to use a mixture of BPM and synchronous outbound interfaces to perform data lookups.  

Each of these methods has advantages and disadvantages and gives you a wide set of tools so that you can select the one that works best in your scenarios. Also note that in ABAP mappings, you can use RFCs to perform lookups or data enrichment. |

### 2.2 Decision Analysis

The example scenario has the following distinguishing features:

- Low data volume
- SLA of information system (the one that contains the conversion logic) must be the same or higher than SAP NetWeaver XI.
- Information rarely changes.
- Conceptually, the mapping is performed by the middleware.
- Works with organizational information.

We will use a value mapping technique. We will implement it in two ways: using the Integration Builder capabilities of SAP NetWeaver XI, and also with help of Solution Manager as an example.

### 2.3 Value Mapping Concepts

To correctly implement the case, let us clarify the concepts involved.
<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Group** | Identifies one entity across the systems. Allows you to see the representations of an object in different places.  
The group has three attributes:  
- **Name**: This concept is used at configuration time only. It helps the user to recognize the shared concept.  
- **ID**: This internal identifier links the information from different systems, that is, it has the translation knowledge. It is internal and automatically generated by the system.  
- **Description**: Additional information for the group. It is not possible to use this description to select the group to configure.  
**Best Practices:**  
- The group is not a “type” of object but the concrete instance of a type. That is, if you are translating currencies, the group name should represent the dollar currency and not a generic currency.  
- If different types of object are assigned the same group name, it is not possible to identify each one (for example, two groups called “Pound”: one for currencies and one for units of force). Therefore, we recommend adding a generic type of object related to all the schemes involved. In this case “Currency – Pound” and “Unit of Force – Pound”. |
| **Context** | Information about the origin of the information.  
- [http://sap.com/xi/XI](http://sap.com/xi/XI) means the values are maintained in the Integration Directory.  
You enter the context as a fixed value in the mapping program and also when loading the configuration records. |
| **Agency** | Represents the agent that manages the identification scheme.  
Depending on the case, the agency is usually one of the following:  
- Particular application software  
- Company  
- Generic business service  
- Standards organization, agency, or consortium  
When writing the mapping program, you can choose whether to specify a fixed value or use the constants “SenderService” or “SenderParty”. In this case, the real value is determined at runtime.  
**Best Practice:** Although this consideration depends on the business case, you should only specify fixed values when the agency is treated as application software, and the group refers to fixed domain values. For example, the agency is “SAP R/3” and the values to convert are language identifiers. In this example you create a low number of conversion records that can be used in many mapping programs. |
provided one of the interfaces deals with SAP R/3 systems. If the concept to translate is the company, using SAP R/3 as agency is a bad solution, since the domain changes depending on the SAP R/3 implementation.

You should also take standard codes into account, for example ISO codes (bear in mind that the SAP ERP system (also SAP R/3) translates some configuration information to or from the ISO values before sending or importing it). Implementing value mapping using standard codes means a reduction in the number of records since the involved agencies are not the systems but the organization that defines the code.

<table>
<thead>
<tr>
<th>Identification Scheme</th>
<th>Specifies the code or role of the concept in the agency.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It is split into two concepts, separated by a space:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Namespace</strong>: Universal identifier for the concept. It is coded as URI (URN or URL). Represents a hierarchy of the company or software component, an organizational division or business process, and the name or the concept.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Object Type</strong>: Used by the agency to identify the concept. When writing the mapping, it should have the same name as the message interface element to translate.</td>
</tr>
<tr>
<td>Value</td>
<td>The concrete object to translate, and provides the context, agency, and scheme.</td>
</tr>
</tbody>
</table>

Although most of these concepts are freely defined, it is important to standardize the names across the company before starting, because during normal operation each system will have to update XI with its value mapping information. It will therefore be necessary to name beforehand, every agency, scheme, and group ID representing each application system and the organizational entities inside those application systems.

### 2.4 Analysis of Value Mapping Implementation

It is now time to work out the technical implementation, considering the business scenario and solution selection.

We can imagine two types of objects when creating the value mapping design:

- **Records**: Information provided by each system, showing how they understand the values involved. They are represented by the agency and ID scheme (both metadata), and the value itself.

- **Links**: Hold the translation information for all the available records. They are implemented as groups. Although groups are optional, an internal identifier (GroupID) is always created by the system.

If we look at the example, there are 3 groups:

- Cars business area
- Motorcycles business area
- Trucks business area

The “Business Area Cars” group should look like this:
At runtime, when the system looks up the My SAP ERP business area code for cars (CARS) from the production system source value (A10), it will access the following fields:

- **Sending agency**: (Defined in the mapping program or at runtime)
- **Source Id. Scheme**: (Defined in the mapping program)
- **Source value**: (Taken from the outbound interface at runtime)
- **Receiving agency**: (Defined in the mapping program or at runtime)
- **Target Id. Scheme**: (Defined in the mapping program)

These five fields are required by the mapping function.

The internal group ID will help you select the correct translation from all the available sending and receiving agencies and schemes, but no group information is specified when writing the mapping program.

Note that what is defined in the mapping program (design time) should not be linked to any particular service from the Integration Directory (configuration time). Moreover, the names of services change when the mapping is exported to the QA system and production system.

Finally, the records to load into the Integration Directory are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Agency</th>
<th>Identification Scheme</th>
<th>Obj Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Unit - Cars</td>
<td>ETC800</td>
<td>urn:speed:tyc:erp BUS_AREA</td>
<td>BUS_AREA</td>
<td>CARS</td>
</tr>
<tr>
<td></td>
<td>CytProdSys</td>
<td>urn:speed:cyt:prodsys CEN001</td>
<td>A10</td>
<td></td>
</tr>
<tr>
<td>Business Unit - Trucks</td>
<td>ETC800</td>
<td>urn:speed:tyc:erp BUS_AREA</td>
<td>BUS_AREA</td>
<td>TRCK</td>
</tr>
<tr>
<td></td>
<td>CytProdSys</td>
<td>urn:speed:cyt:prodsys CEN001</td>
<td>A20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TYMSales</td>
<td>urn:speed:tym:sales DIVSN</td>
<td>00012</td>
<td></td>
</tr>
<tr>
<td>Business Unit - Motorcycles</td>
<td>TYMSales</td>
<td>urn:speed:tym:sales DIVSN</td>
<td>00032</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMO800</td>
<td>urn:speed:mc:r3 BUS_AREA</td>
<td>BUS_AREA</td>
<td>MTCL</td>
</tr>
</tbody>
</table>
2.5 Solution Manager – XI Value Mapping Objects Equivalence

<table>
<thead>
<tr>
<th>Solution Manager</th>
<th>Exchange Infrastructure</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customizing</td>
<td>Identification scheme</td>
<td>urn:speed:cyt:prodsys CEN001</td>
<td>Look up ID schema definition</td>
</tr>
<tr>
<td>ID type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global ID type</td>
<td>N/A</td>
<td>BusinessUnit</td>
<td>Generic concept that groups similar object for conversion</td>
</tr>
<tr>
<td>Business system</td>
<td>Agency</td>
<td>CytProdSys</td>
<td>Look up Agency Definition</td>
</tr>
<tr>
<td>Type groups</td>
<td>(inside groups)</td>
<td></td>
<td>Relationship between business systems, customizing ID types,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and the Integration Server</td>
</tr>
<tr>
<td>ID mapping</td>
<td>(inside groups)</td>
<td></td>
<td>Defines the equivalence table</td>
</tr>
<tr>
<td>definition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>Group</td>
<td>Business Units – Cars</td>
<td>Look up Group definition</td>
</tr>
</tbody>
</table>

2.6 Additional Comments

- Even though no description is provided for the group in this example, we recommend that you add one.
- Since Solution Manager keeps a naming range for customer developments, we will add a preceding “z” or “y” to each value mapping object type in the Solution Manager example to comply with it, even though this is not required by SAP NetWeaver XI. Also, if Solution Manager already has a corresponding definition for the entity (for example, countries, regions, and plants) it is preferable to use it.
3 The Step By Step Solution

Now it is time to implement the solution. You need to create an interaction between business services, without parties, and using mapping programs as conversion methods.

The following steps show how to perform value-mapping-related activities for the “Cars” group.

3.1 Mapping Design Time: Integration Repository

You perform the mapping design time activities in the Integration Repository, independently of the source of the records.

1. Provided you already have the source and target messages, you create the mapping program.

2. To add the Value Mapping function, select the involved fields, select Conversions under Functions, and choose Value Mapping.
3. Link the fields and complete the value mapping information by double-clicking the Value Mapping function.

4. Specify the sender and receiver agencies and schemes.

Both agencies can be replaced by the environment variables representing either the service or the party.

Select SenderService and ReceiverService to represent the runtime service for the agency.

5. Complete, save, and activate the mapping program.
3.2 Generating Solution Manager Customizing IDs

In this section, you will generate value mapping records in XI that are generated in the Solution Manager.

3.2.1 Obtaining reference objects from the source application ABAP system

6. If you are defining a concept available in a reference ABAP system, first access that system to retrieve the attributes. Call transaction SPRO and choose SAP Reference IMG.

7. To access the attributes, first locate the object.
8. To get the field attributes, open the object.

9. To get the field technical details, position the cursor on the description field and press help (F1)
10. To get the field attributes, choose Technical info.

11. Note the Field name.
12. To get the key value description, repeat the same procedure (F1 + technical info), but with the cursor positioned on the business area field. Note the Field name.

13. To get other node attributes, exit Business Area maintenance and go back the IMG tree.
14. To display the node technical info, call the context menu for the node and choose *Display technical info*.

15. To access the required information, select the *Maint. objects* tab page, and note the customizing object and ID.
3.2.2 Solution Manager

16. To start, log on to the Solution manager and call transaction SCIDM_CONFIG.

17. To create the customizing ID type, choose ID Types.

18. To create a new Customizing ID Type, choose the Create icon.
19. Specify the following parameters:

- **Namespace**: as defined above
- **Description**: a free text
- **Classification**: C for Customizing data
- **Customizing Object**: The customizing object specified in the ABAP reference system
- **Customizing Object Type**: Since the customizing object has a description field, specify “S” for a table with text.
- **Text Field**: Enter the description field name taken from the ABAP system.
- **ID Structure**: Enter a number for the line, a field name of your choice, the field length, and the key field name from the ABAP system.

If the source system is not an ABAP system, most of these attributes will not be taken into account, since no list of values will be retrieved.

20. To save, choose the **Save** icon, specify the package, and save again.
21. Repeat the procedure for the center entity; if there is no ABAP reference system, simply complete the customizing ID transaction, specifying a suitable field length. Finally, go back to the menu.

22. To create a global ID type, choose *Global ID Types.*
23. To group all the similar objects for conversion, create a global customizing ID type with a suitable description and namespace. 
   → Add all the customizing ID types previously created. 
   → Specify single value cardinality for each. 
   → Also classify the object as customizing data.

Finally, save the object and exit the transaction.

24. To specify the business systems, choose Business Systems.
25. To create a business system, you do not need to create it first in the SLD. You can enter the application and XI business system names directly.

Select the Integration Server checkbox. This checkbox will only have an effect if you use the Integration Server of SAP XI.

Select the Fixed RFC checkbox and maintain a suitable RFC destination to access that system. This option is necessary regardless of whether you are using the XI Integration Server or an ABAP reference system.

Save and exit the transaction.

26. To create the type groups, choose the corresponding pushbutton.

27. To create a group, choose the Create icon
28. Enter the name of the XI business system that will receive the information, a type group name, and a description.

To complete the customizing ID types, choose **ID Types from Global ID Type**.

29. Select the appropriate row.

30. To specify the system where each entity is defined, specify the business system for each.

Finally, save your work.
31. To create the values relationship, choose *ID Mapping Definition*.

32. To create the records, select the Type Group and choose the *Change* icon.

33. To add the columns to the conversion table, select the *Display* checkbox, and also the *Display Text* checkbox (if available in the source system).
34. To get the records from the source system, select the source system row and choose *Display IDs from System*.

35. To add the record, first select the row, then drag and drop it in the *ID Mappings* area.

   A new record will appear.

36. To fill the row with values from the other system, select the source system and choose *Display IDs from System* again.

   Since the system is not connected to an ABAP system, an error log will appear and the reference table will appear empty.
37. To create values directly in the Solution Manager, choose Create Value Set Manually.

38. Enter the values as necessary.
39. To finish the conversion, select the New IDs row, then drag and drop the value in the ID Mappings row. Save and return to the menu.

40. To notify the system that the conversions are correct, switch the processing status to Saved, and repeat the process. Finally, mark it as Completed.

41. To send the information to the XI cache, choose ID Mapping Export

42. To generate the replication background job, specify the Integration Server, Type Group, and choose Execute.

Specify the background job parameters as usual.
43. To access the replication log, call transaction SLG1.

Select the application log object SCIDM, subobject SCIDM_IS_EXPORT.

Choose Execute.

44. To view the details, select your job run row.
3.3 Generating Value Mapping Records in XI

In this section, you will generate value mapping records in XI.

3.3.1 Configuration Time: Integration Directory

45. To create the value mapping records, choose Create Object.

46. Create the Group selecting Value Mapping Group.

   Complete the Group Name and Description as required.
Complete the records and save.
3.3.2 Value Mapping Records Maintenance in XI

To display or change the value mapping records, access the Integration Directory, choose Tools → Value Mapping, and execute one of the three different access methods.

A: Value Mapping Agencies:
Allows you to work with conversions of one entity type (group independent) from one agency to another.
For example: Business area conversions from system CytProdSys to system ETC800.
**B: Value Mapping Group Selected By - Group Name:**

Allows you to work with conversions of one entity instance (group) in every agency, specifying the group name.
C: Value Mapping Group Selected By - Representation: Allows you to work with conversions of one entity instance (group) in every agency, specifying one single record of the group.

![Value Mapping Group Selected By](image)

<table>
<thead>
<tr>
<th>Agency</th>
<th>Scheme</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CylFroSys</td>
<td>umspeed.ycprosys.CEN.001</td>
<td>A10</td>
</tr>
<tr>
<td>ETC800</td>
<td>umspeed.ucprosys.BUS_AREA</td>
<td>CAR3</td>
</tr>
</tbody>
</table>
4 Appendix A: – Links to Related Documentation

- SAP Help Portal

- SAP Solution Manager ID Mappings Framework:
  - You use the Customizing ID Mapping Framework in the SAP Solution Manager to exchange messages between unharmonized systems. In unharmonized systems, different IDs represent the same entity ([http://help.sap.com/saphelp_nw04/helpdata/en/cd/9ad4dfdb9ef04cba985a566422569c/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/cd/9ad4dfdb9ef04cba985a566422569c/frameset.htm)).
www.sdn.sap.com/irj/sdn/howtoguides