How to Perform Value Mapping – A Walkthrough

Applies to:

SAP XI 3.0 /SAP PI 7.0

Summary

In general, there might be a scenario where a value being sent depends upon various constraints and needs to be looked up in a table and sent across accordingly. For example, let us take a scenario where a company maintains the information of its Stream ID (for e.g. Finance ID, HR ID, etc...) based on the employee number. An employee might be in more than 1 stream at a time. So maintaining the stream ids of all the employees at a time and accessing them where ever it is required in XI is a tough job. For avoiding any confusion and for easy accessibility, XI has introduced a concept called Value Mapping meaning Mapping by Values based on the Identification groups and schemes. This document walks the reader through the concept and how to implement it in real-time scenarios.

Author: Sarath Chandra Kandadai

Company: Infosys Technologies Limited

Created on: 31 May 2007

Author Bio

Sarath Chandra Kandadai is a Technical Specialist at Infosys Technologies Ltd. and has been working on SAP XI for the past two years. He has an overall EAI experience of 4 years.

© 2007 SAP AG

Table of Contents

Maintaining the Value Mapping Table 3
Viewing the Value mapping entries in RWB 3
Manual creation/edition of the table in ID....5
Creation of the Value Mapping Group in ID 6
Editing / Viewing the Value Mapping Group in ID 6
Value Mapping Replication for updating entries in Runtime Cache 7
Steps to be performed for updating Java Runtime Cache 8
Steps for Creating a Java Proxy out of an Interface in Repository 8
Steps for creating the scenario for VMR in Directory 12
1. Implementing Value Mapping in Mapping....13
Using Value mapping in Graphical Mapping....13
Using Value mapping in Java and XSLT Mapping 14
2. Appendix 17
3. Reference 18
Disclaimer and Liability Notice....19

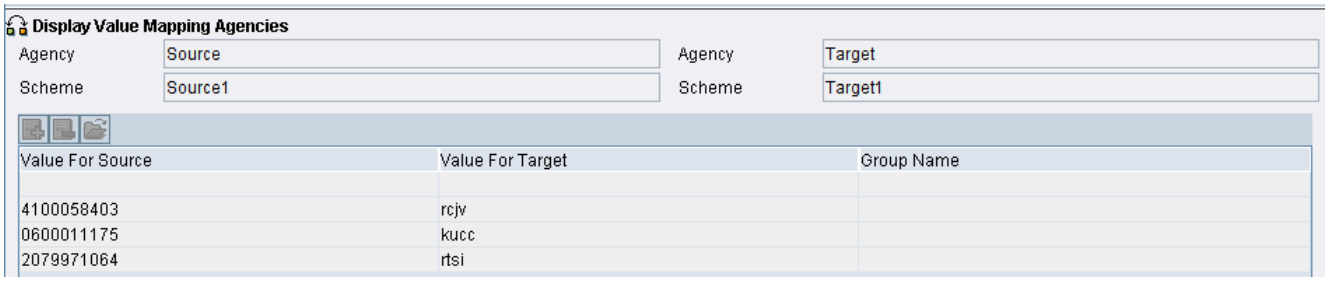
© 2007 SAP AG

Maintaining the Value Mapping Table

The Value Mapping table consists of the following columns:

1. Scheme – The identification scheme (in general terms the Company)
2. Agency – The identification Agency (in general terms the Stream)
3. Value – The value for a particular agency and for a particular scheme (in here the ID)
4. Context – The context in which the value is set (In XI, the context by default is http://sap.com/xi/XI)
5. Group ID – The group ID of the value mapping group – generally in hex format.

The representation of the above in a tabular format looks like the following.



A different representation of the above table is given below

Agency	Scheme	Value
Source	Source1	4100058403
Target	Target1	rcjv

In both the above tables, we can see that the Target1 in Target, whose Source1 in Source is 4100058403, is rcjv. It is more like a name-value pair. Similarly, we can add more number of entries to the table.

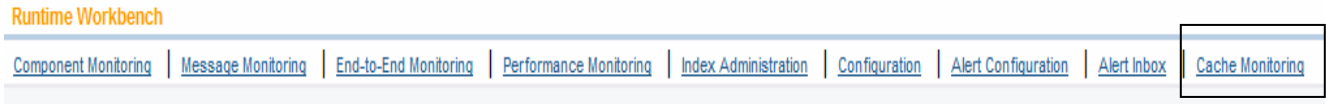
Viewing the Value mapping entries in RWB

This table, when activated, gets updated in the Java runtime cache and is available during mapping program execution. It can be viewed from SAP XI Runtime Work bench under Cache Monitoring.

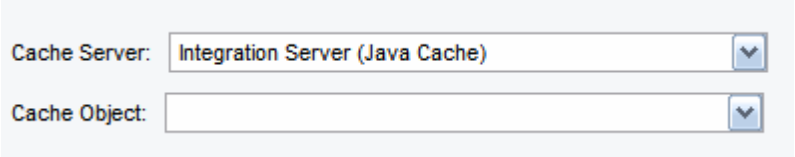
The following steps will show how the value mapping cache can be viewed:

1. Open Runtime Work Bench

2. Go to the tab Cache Monitoring

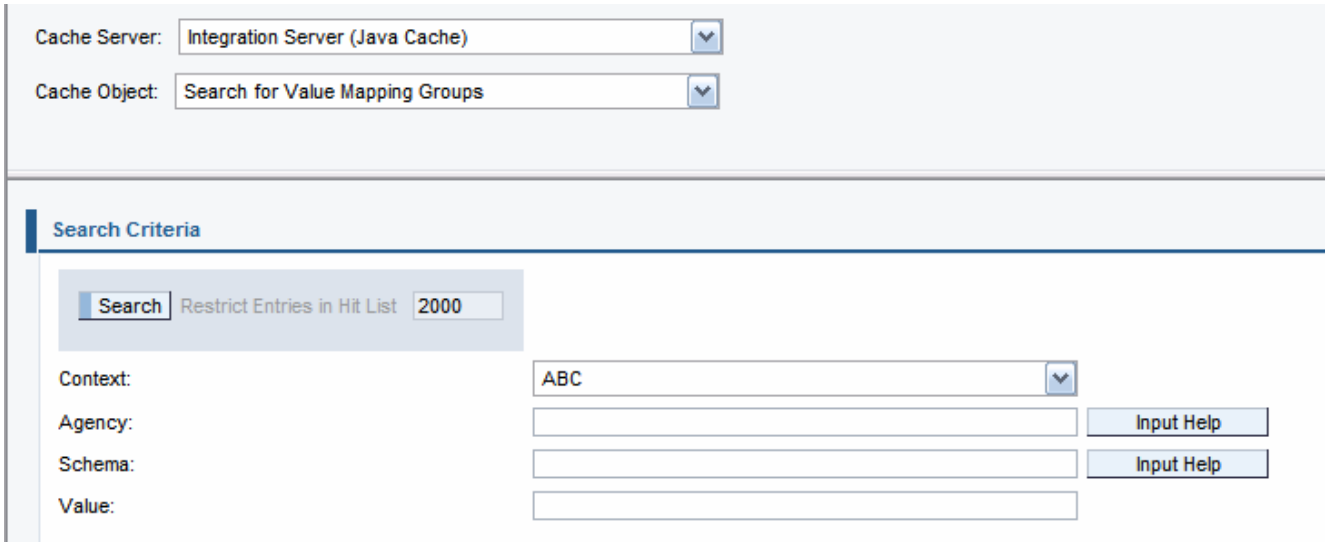


3. Select the Cache server as Integration Server (Java Cache)

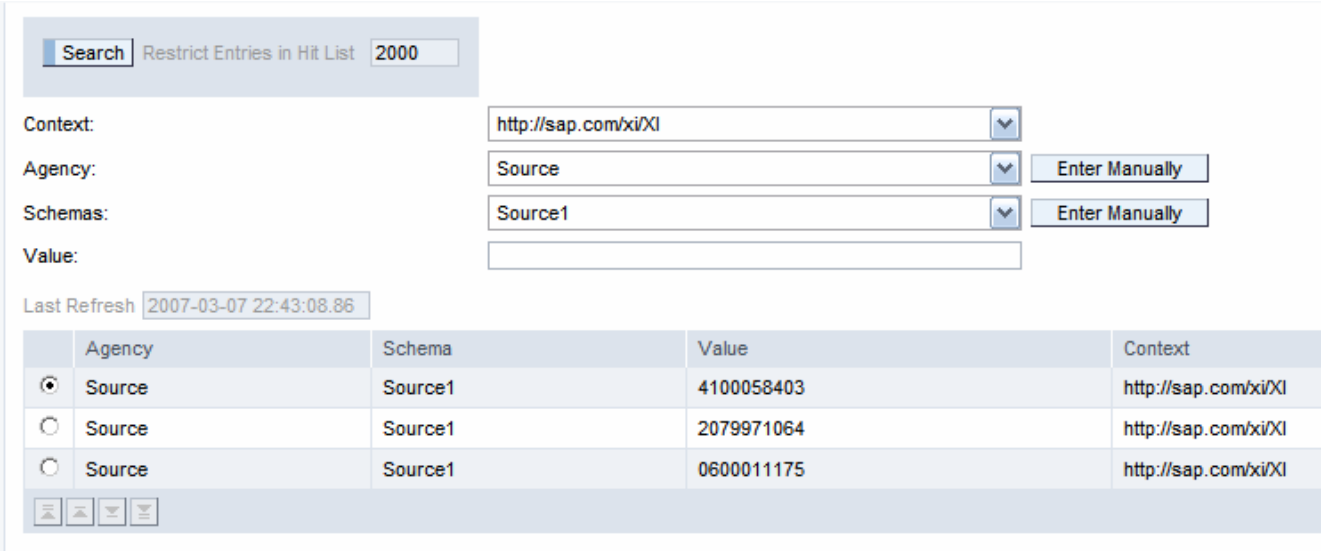


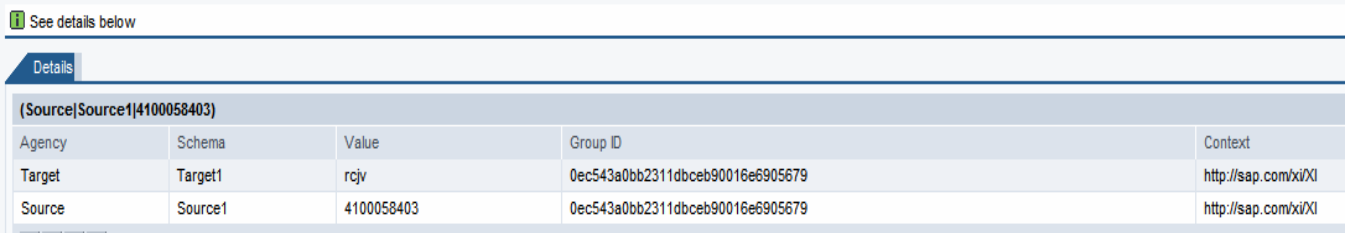
4. Select the Cache Object as “Search for Value Mapping Groups”. This will result in a screen as shown below.

© 2007 SAP AG



5. In the sections, fill in the details for Context, Agency, Schema and Value if a specific entry is required. Or just press Search and all the results will be displayed for that context. Context is a required field. The output will look like below diagram.





The data can be later accessed in the mapping programs. The steps to access this will be discussed later in the document.

The Value Mapping table that is created in the Integration directory can be updated by modifying the entries manually.

The java cache for value mapping can be updated by adding the entries using an interface provided by SAP along with XI Content. This method is also called Value Mapping Replication.

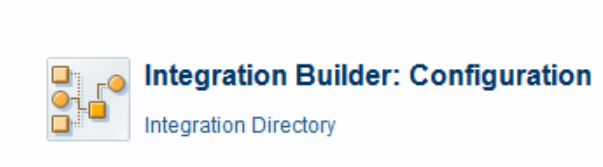
© 2007 SAP AG

Manual creation/edition of the table in ID

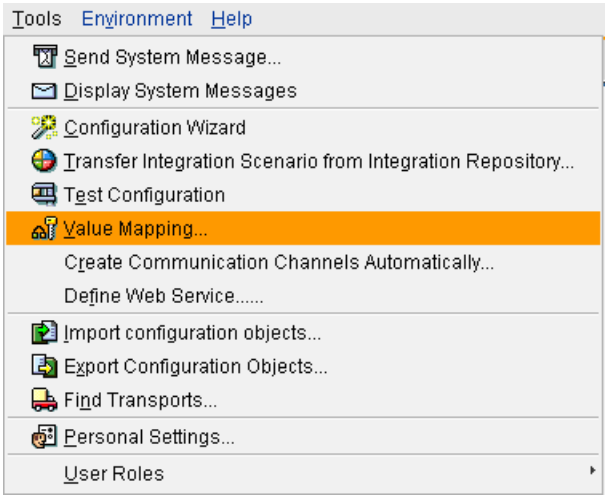
The Value Mapping table can be created / edited manually in the Integration Directory.

The following steps will provide an overview.

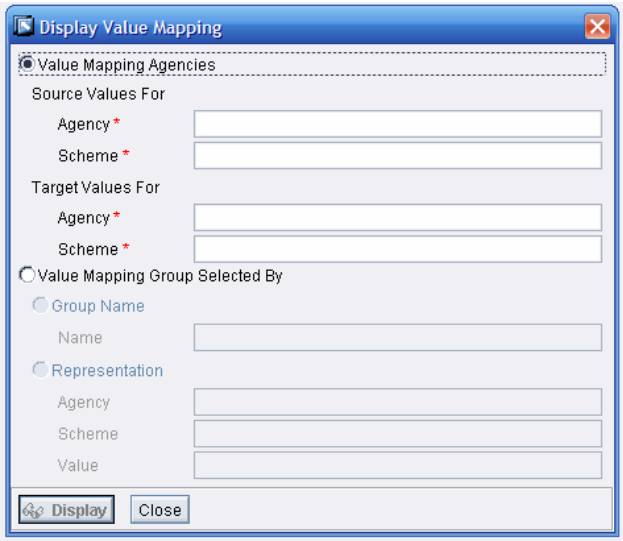
1. Open Integration Directory.



2. Go to Tools → Value Mapping.



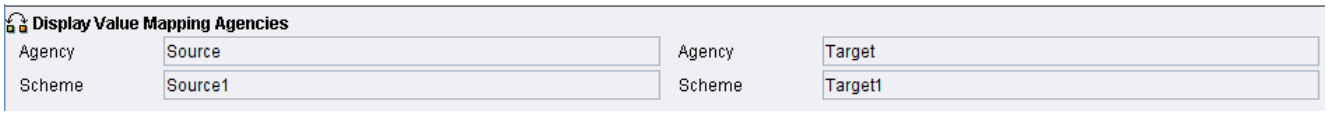
3. The following screen will appear when the above Value Mapping is opened



© 2007 SAP AG

Creation of the Value Mapping Group in ID

1. If a new table is to be created, enter the values for Source Agency, Source Schema, Target Agency and Target Schema. Then press Display. This will open a screen as shown below. (For example, the Source and Target Agency is Source and Target and the Source Scheme is Source1 and Target Scheme is Target1. This table will hold the relation between the Source1 value in Source and the relative Target1 value in Target).



2. Press the button to add entries to the table. The values can be grouped under IBUID. This will later help in easier identification of the entry.

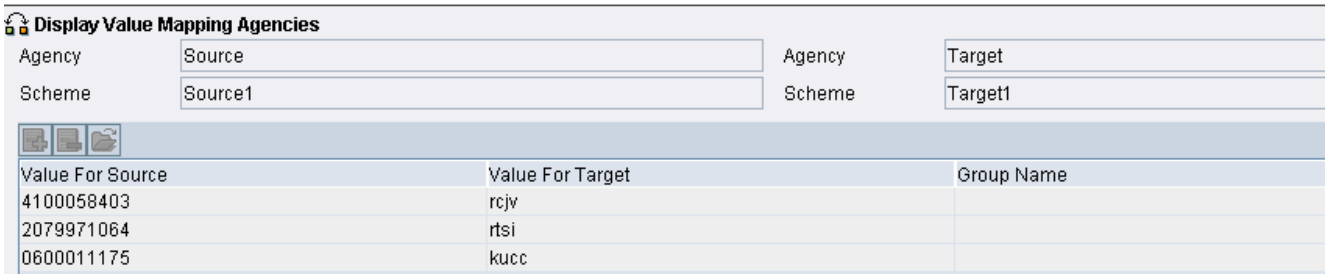
Value For Source	Value For Target	Group Name
4100058403	rcjv	

3. Double clicking on the entries will take to another screen which displays the same table based on GroupID.

4. After activation, this entry is available for mapping during mapping execution in runtime.

Editing / Viewing the Value Mapping Group in ID

1. If existing values are to be modified / viewed, then press the button to select the values. Then click to display the table containing the details.



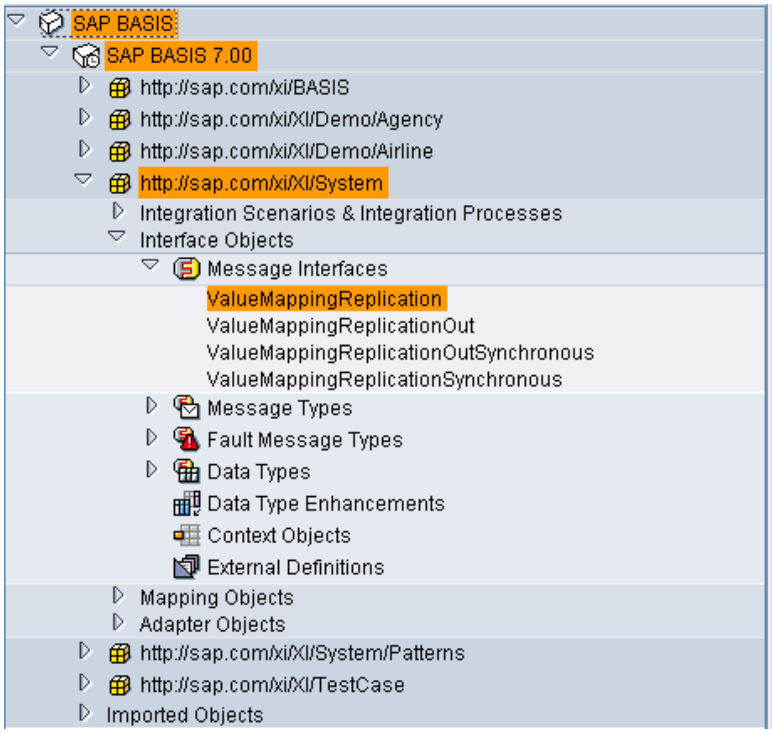
2. From then on, follow the steps from step 2 in section 2.2.1.

© 2007 SAP AG

Value Mapping Replication for updating entries in Runtime Cache

As mentioned in section 2.1, the Java Runtime Cache can be updated using Value Mapping Replication. The following sections will discuss in detail the steps that are required to achieve the same.

The Value Mapping replication is provided by SAP as standard content along with XI. It is present in the Integration repository under SAP BASIS → SAP BASIS <version> →http://sap.com/xi/XI/System. The following screenshot will provide a view of the same.



As it can be seen, there are four message interfaces.

1. An Outbound Asynchronous – ValueMappingReplicationOut
2. An Outbound Synchronous – ValueMappingReplicationOutSynchronous
3. An Inbound Asynchronous – ValueMappingReplication
4. An Inbound Synchronous – ValueMappingReplicationSynchronous

Basing on the requirement, any of the above four interfaces can be used. The synchronous interface will contain a response type as well. The response type is a single status field. The following screen shots will provide information about the ValueMappingReplication Type and the ValueMappingReplicationResponse Type.

ValueMappingReplication Type

ValueMappingReplication	Element	ValueMappingReplicationList				
Item	Element	ValueMappingReplicationItem	1..unbounded			
Operation	Element	xsd:string	1	enumeration="Insert ...		Insert = all fields must be
GroupID	Element	xsd:string	0..1	length="32"		Guid in Hex-Format neede
Context	Element	xsd:string	1	maxLength="120"		needed for Insert Delete, D
Identifier	Element	ValueMappingIdentifier	0..1			needed for insert, Delete
scheme	Attribute	xsd:string	required	maxLength="120"		
agency	Attribute	xsd:string	optional	maxLength="120"		

ValueMappingReplicationResponse Type

ValueMappingReplicationResponse	Element	ValueMappingReplicationStatus

© 2007 SAP AG

In the ValueMappingReplication Type, there are some fields which are mandatory (as can be seen above) for certain operations. The list of Mandatory fields for different operations is provided below.

- Insert – all fields must be set
- Delete – all fields must be set
- DeleteGroup – fields GroupID and context must be set
- DeleteContext – field Context must be set
- DeleteContextGeneric – Context contains the starting part for the context to be deleted

There is no Update operation. If some value has to be updated, then the existing value has to be deleted first and the new value has to be inserted.

Sample XML for Insert and Delete operations has been provided in the appendix.

Steps to be performed for updating Java Runtime Cache

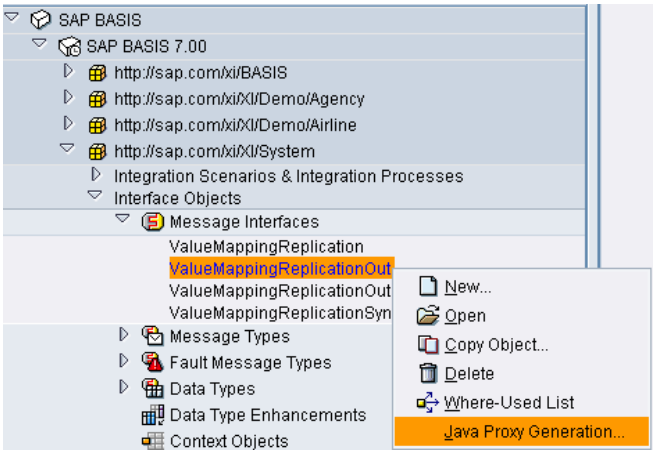
The following are the steps that are to be performed in order to send a message to the Java Runtime Cache to update the entries.

No new interface needs to be created as they already exist. The only thing that needs to be done is that Java proxies need to be generated out of these interfaces and registered in the J2EE Visual Admin.

Steps for Creating a Java Proxy out of an Interface in Repository

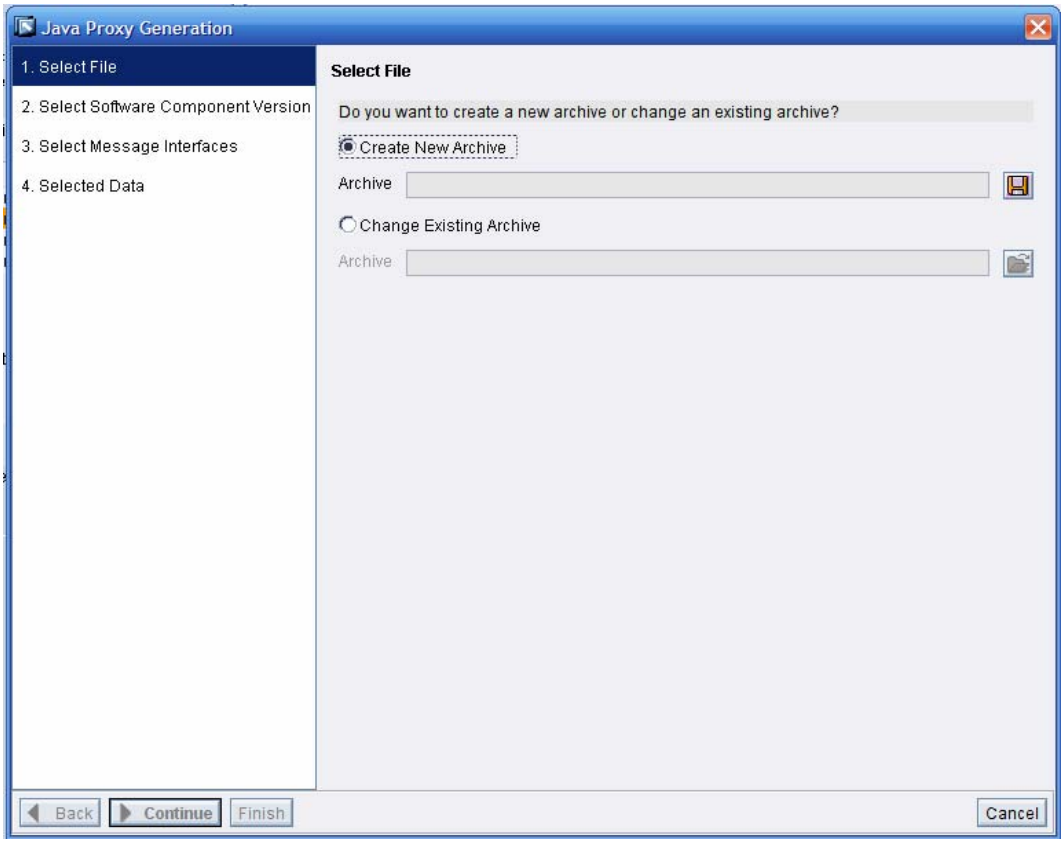
The following steps will describe how to create the Java proxy out of an interface.

1. Select the interface for which the java proxy needs to be generated and right-click on it.

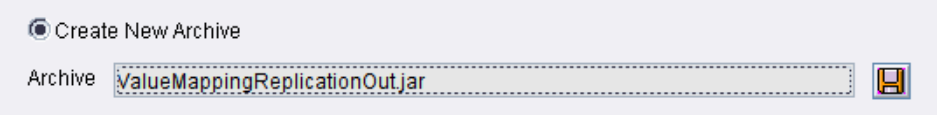


2. Click on Java Proxy Generation. The resultant will be a popup of a wizard which will guide through the creation.

© 2007 SAP AG

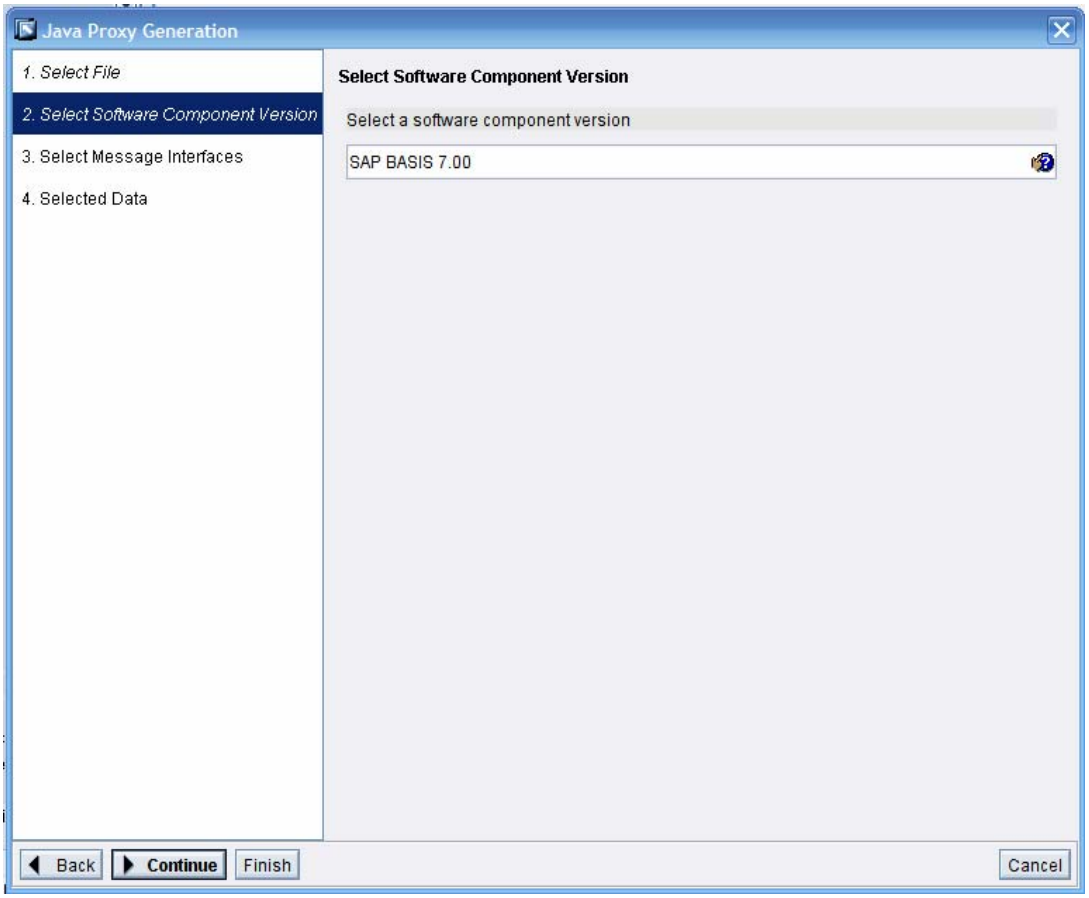


3. Select Create New Archive if a new one has to be created. Do not forget to give the extension as .jar as by default it will not create any extension. Click on the button to select the location where to save the file.

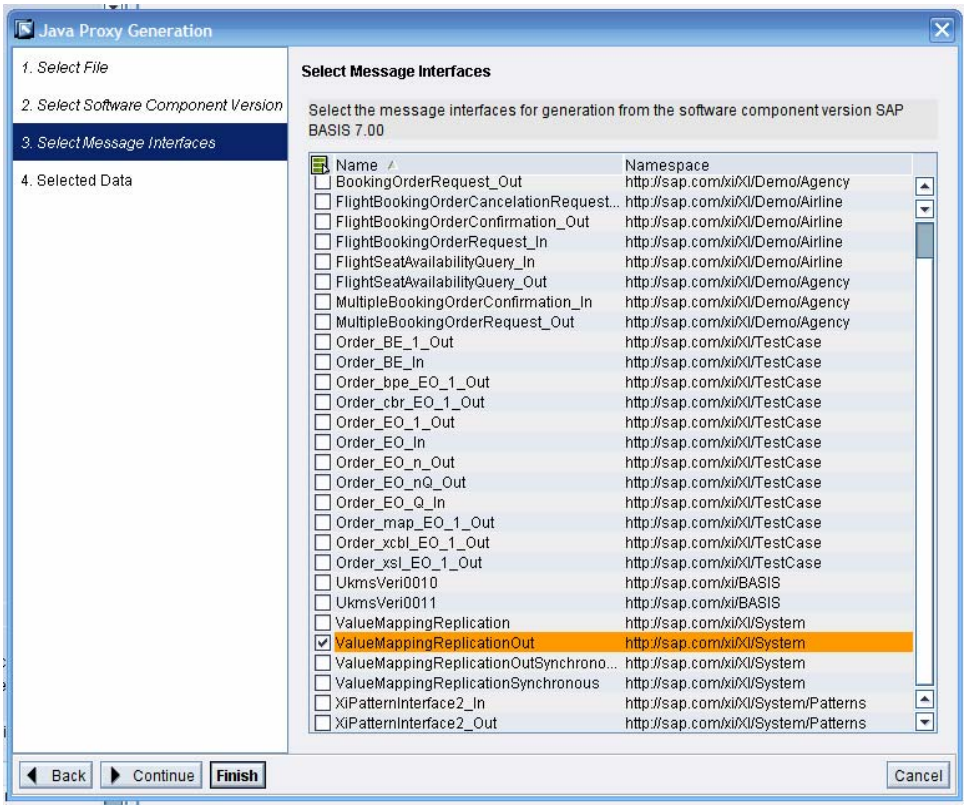


4. Click on Continue. In the next screen the software component version needs to be specified. Generally, the software component version will be SAP BASIS 7.00

© 2007 SAP AG

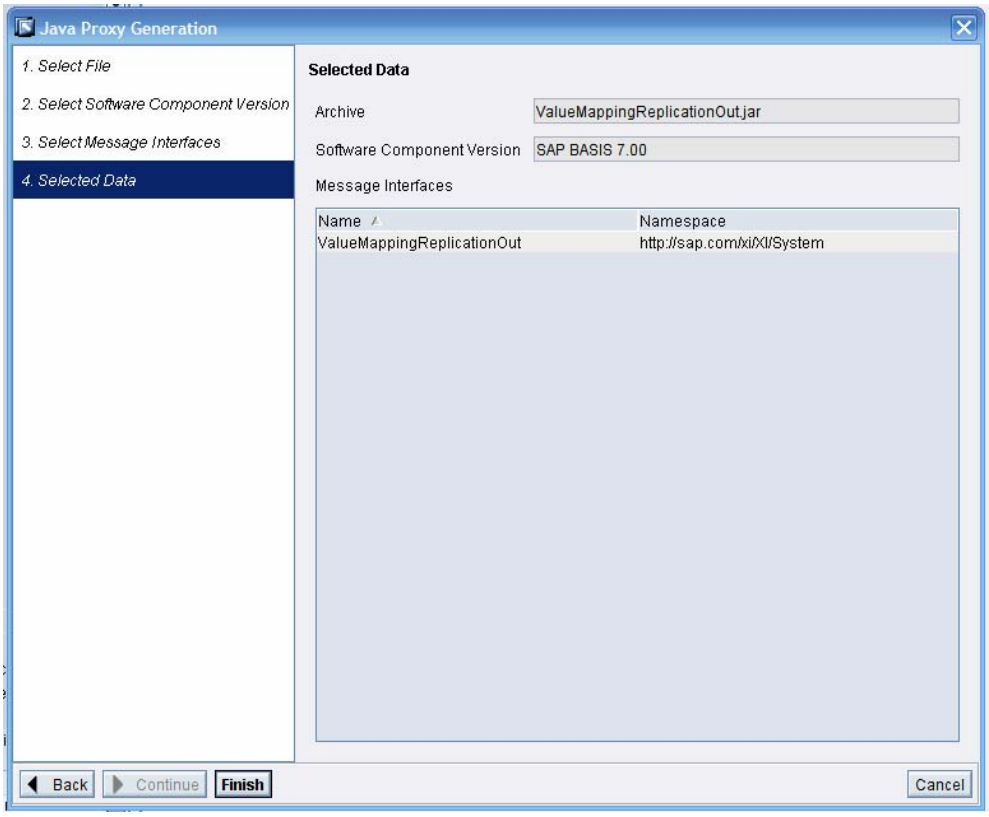


5. Click on Continue. In the next screen the Message Interface needs to be specified. Since we have already selected the Interface, it will be selected directly.

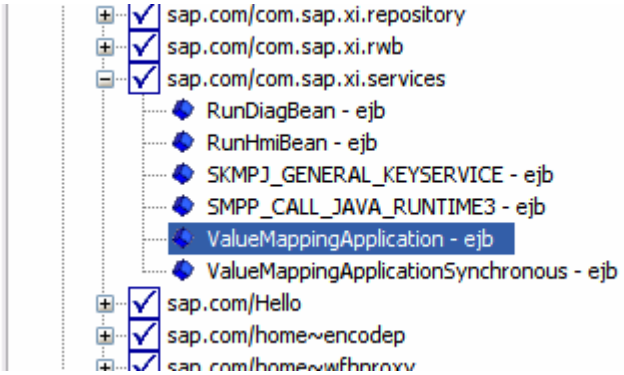


© 2007 SAP AG

6. Click on Continue. This will show the selected data. Finally click on Finish.



7. This will generate deployable jar files. These have to be deployed in the J2EE Visual Admin.



After it has been deployed, it has to be registered using the following URL

http://<host>:<port>/ProxyServer/register?ns=http://sap.com/xi/XI/System&interface=ValueMappingReplicationOut&bean=ValueMappingApplication&method=valueMappingReplication

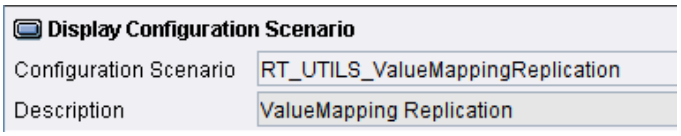
Then you will get a message "Interface http://sap.com/xi/XI/System#ValueMappingReplicationOut registered with value ValueMappingApplication:valueMappingReplication".

Once it has been registered, messages can be sent to the Interface. But for that a scenario has to be created in the Integration Directory. The following steps will guide through the creation.

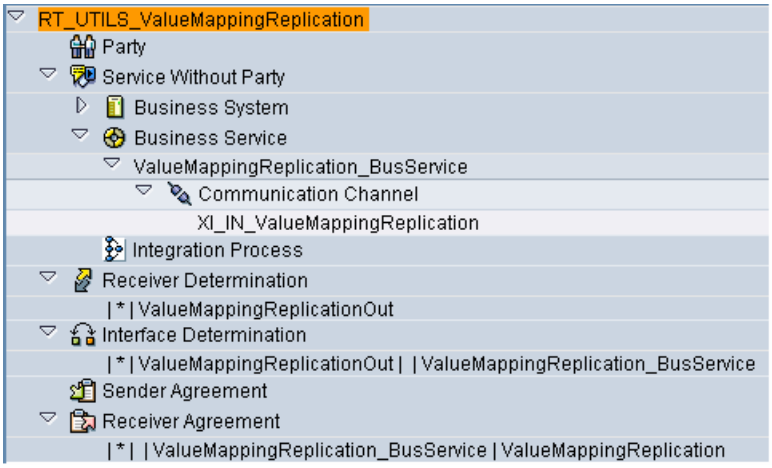
© 2007 SAP AG

Steps for creating the scenario for VMR in Directory

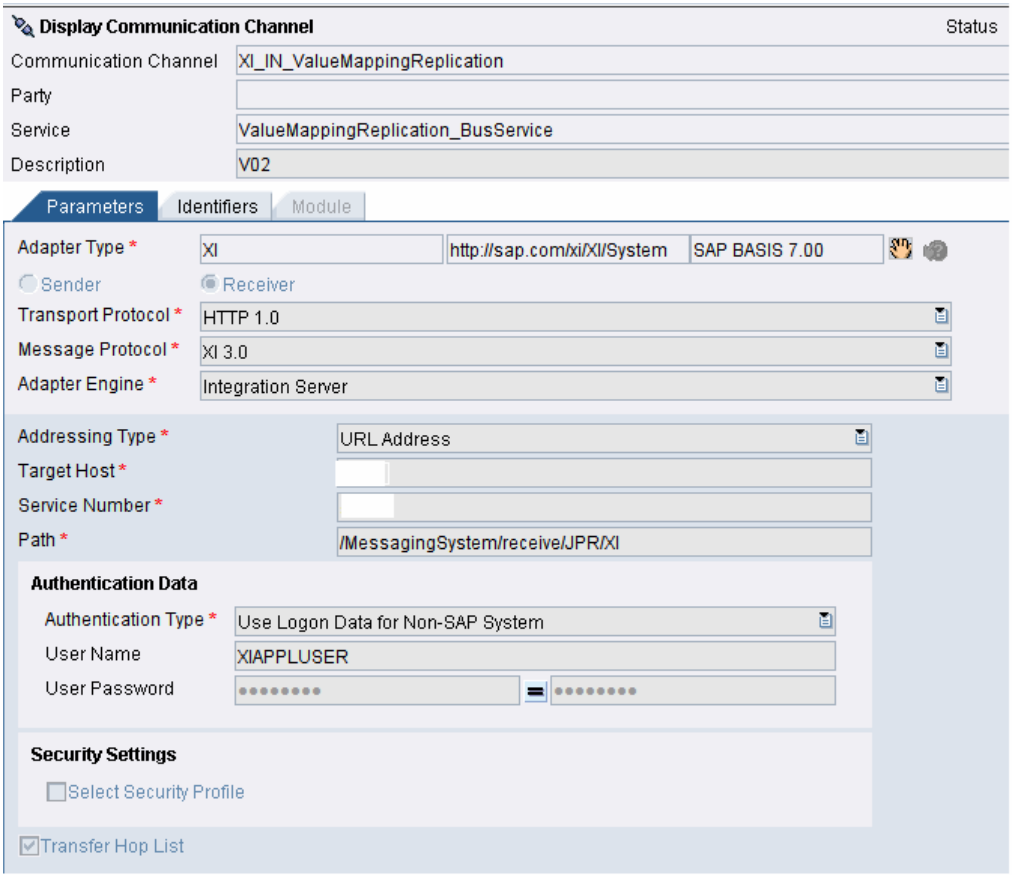
1. Create a scenario which can be used as a common one. In this case, it will be RT_Utils_ValueMappingReplication.



2. The sender can be anyone. Hence the sender is *. The sender channel used here is HTTP hence no sender agreement is required. The receiver service is ValueMappingReplication_BusService. The channel that should be used is XI type with the name XI_IN_ValueMappingReplication. Basing on this, the whole scenario is created.



3. The following figure shows the configuration that needs to be done in the channel. The message is sent to the JPR of XI using the XIAPPLUSER or any user having this role.



© 2007 SAP AG

1. Implementing Value Mapping in Mapping

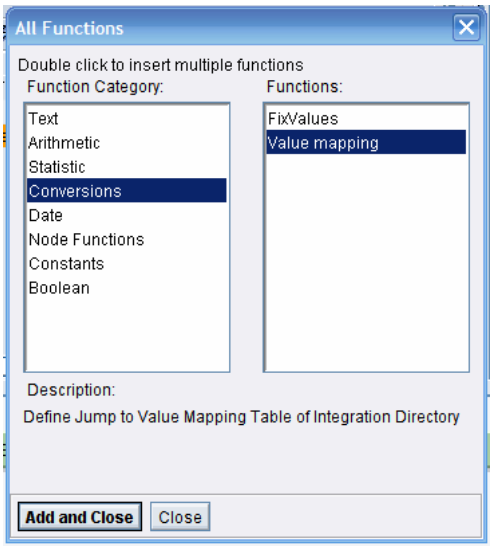
Till now, the steps that are required to maintain the Value Mapping Table using ID and VMR have been discussed. Now the steps that are required to use the created value mapping table in the mapping will be discussed.

As already mentioned in the section1 – Introduction, the value mapping can be used in

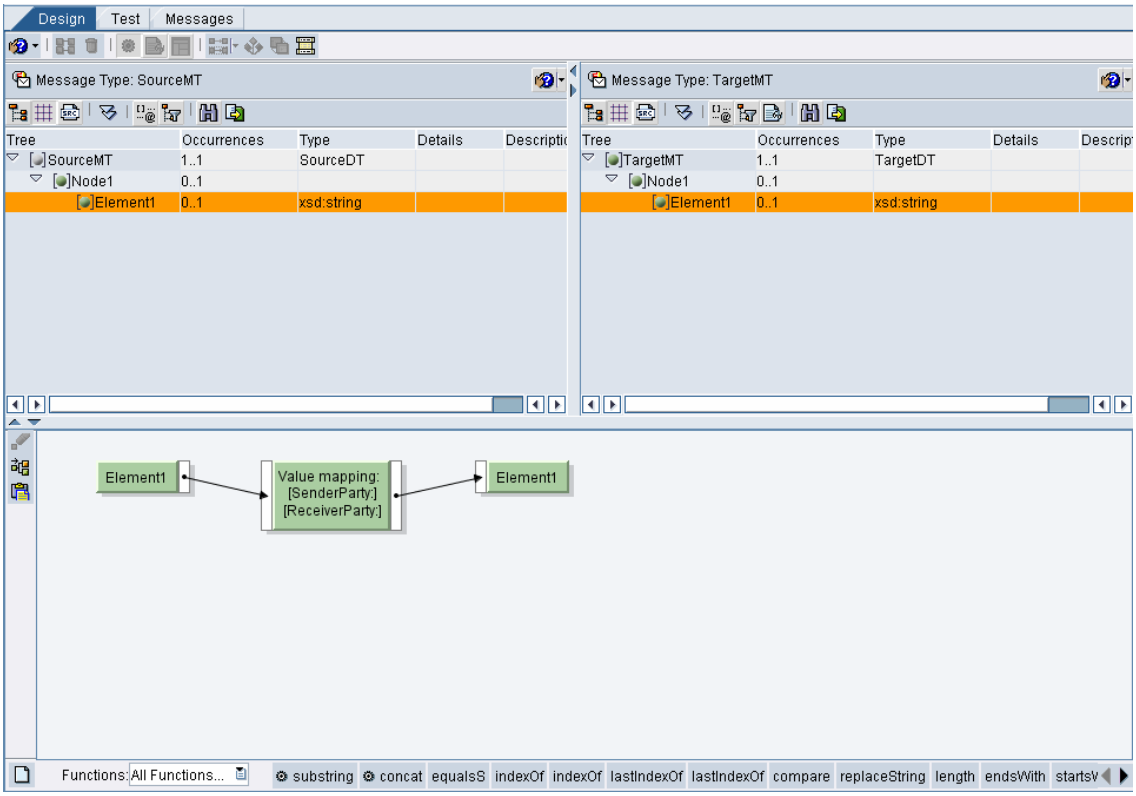
1. Graphical Mapping (Message Mapping).
2. Java Mapping (by using the Java API provided by SAP).
3. XSLT Mapping (by creating a Java class and calling it in XSLT).

Using Value mapping in Graphical Mapping

1. In Graphical Mapping, there is a function called Value Mapping under Conversions.

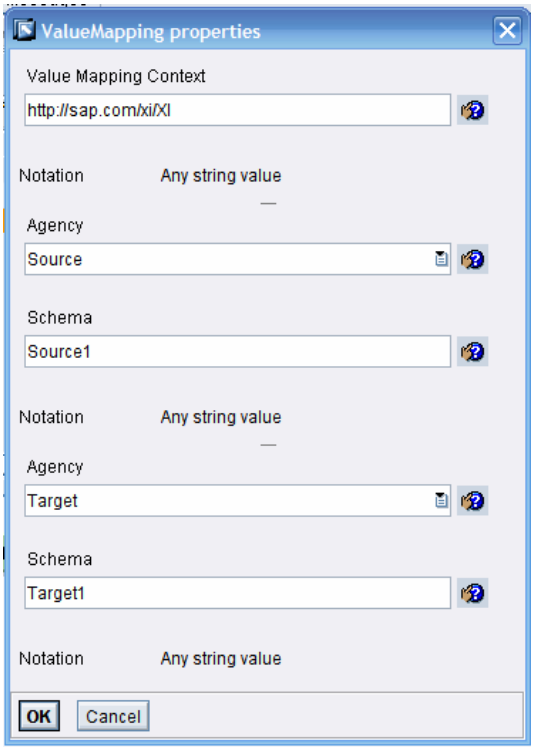


2. Click on Add and Close. Then map the Source Element to the Target Element using this function.



3. After adding, double click on the function which will open a window to configure it.

© 2007 SAP AG



4. This finishes the mapping. However, there is one issue when testing Value Mapping. It can not be tested during design-time. It has to be tested during runtime alone.

Using Value mapping in Java and XSLT Mapping

Java Classes can be created using the java API provided by SAP. The following link will provide the methods provided.

https://help.sap.com/javadocs/NW04S/current/pi/index.html (for NW 2004s)

https://help.sap.com/javadocs/NW04/current/pi/index.html (for NW 2004)

Using these, java mappings can be created and used in the Interface Mapping. Similarly, a java class can be created to be used in the XSLT mapping.

Packages	
com.sap.aii.mapping.api	API for Java mapping programs.
com.sap.aii.mapping.lookup	Lookup API for data enrichement in mapping programs.
com.sap.aii.mapping.value.api	API for value mapping requests.

A sample java class is provided for further information about the usage of the value mapping API.

```
//Start of the code
import com.sap.aii.mapping.value.api.*;
public class ValueMap
{
   public ValueMap()
   {
   }
   public static Object Lookup(String sourceContext, String sourceAgency, String sourceScheme, String
targetContext, String targetAgency, String targetScheme, String keyValue)
      throws Exception
```


© 2007 SAP AG

```
{
    String retValue = new String();
    try
    {
        com.sap.aii.mapping.value.api.IFIdentifier sourceIdentifier =
XIVMFactory.newIdentifier(sourceAgency, sourceScheme);
        com.sap.aii.mapping.value.api.IFIdentifier targetIdentifier = XIVMFactory.newIdentifier(targetAgency,
targetScheme);
        com.sap.aii.mapping.value.api.IFRequest mRequest = XIVMFactory.newRequest(sourceIdentifier,
targetIdentifier, keyValue);
        try
        {
            IFResponse mResponse = XIVMService.executeMapping(mRequest);
            if(mResponse.hasTargetValues())
            {
                retValue = mResponse.getSingleTargetValue();
            } else
            {
                String exceptionMessage = new String("LookupValueNotFound:");
                exceptionMessage = exceptionMessage + sourceAgency + sourceScheme + targetAgency +
targetScheme + keyValue;
                return exceptionMessage;
            }
        }
        catch(ValueMappingException e)
        {
            String exceptionMessage = e.getMessage();
            e.printStackTrace();
            throw new Exception(exceptionMessage);
        }
    }
    catch(Exception e)
    {
        String exceptionMessage = e.getMessage();
        e.printStackTrace();
        throw new Exception(exceptionMessage);
    }
    return retValue;
}
public Object LookupXSLT(String sourceContext, String sourceAgency, String sourceScheme, String
targetContext, String targetAgency, String targetScheme, String keyValue)
    throws Exception
{
    String retValue = new String();
```


© 2007 SAP AG

```
        String retValue2[][] = (String[][])null;
        int retValue3 = 0;
        try
        {
            com.sap.aii.mapping.value.api.IFIdentifier sourceIdentifier =
XIVMFactory.newIdentifier(sourceAgency, sourceScheme);
            com.sap.aii.mapping.value.api.IFIdentifier targetIdentifier = XIVMFactory.newIdentifier(targetAgency,
targetScheme);
            com.sap.aii.mapping.value.api.IFRequest mRequest = XIVMFactory.newRequest(sourceIdentifier,
targetIdentifier, keyValue);
            try
            {
                IFResponse mResponse = XIVMService.executeMapping(mRequest);
                if(mResponse.hasTargetValues())
                {
                    retValue = mResponse.getSingleTargetValue();
                } else
                {
                    String exceptionMessage = new String("LookupValueNotFound");
                    throw new Exception(exceptionMessage);
                }
            }
            catch(ValueMappingException e)
            {
                String exceptionMessage = e.getMessage();
                e.printStackTrace();
                throw new Exception(exceptionMessage);
            }
        }
        catch(Exception e)
        {
            String exceptionMessage = e.getMessage();
            e.printStackTrace();
            throw new Exception(exceptionMessage);
        }
        return retValue;
    }
}
// End of the code
```

The provided methods can be used in the XSLT mapping.

© 2007 SAP AG

2. Appendix

Sample XML for each operation has been provided below.

For Insert operation:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:ValueMappingReplication xmlns:ns0="http://sap.com/xi/XI/System">
    <Item>
        <Operation>Insert</Operation>
        <GroupID>0ec543a0bb2311dbceb90016e6905679</GroupID>
        <Context>http://sap.com/xi/XI</Context>
        <Identifier scheme="Source1" agency="Source">4100058403</Identifier>
    </Item>
    <Item>
        <Operation>Insert</Operation>
        <GroupID>0ec543a0bb2311dbceb90016e6905679</GroupID>
        <Context>http://sap.com/xi/XI</Context>
        <Identifier scheme="Target1" agency="Target">rcjv</Identifier>
    </Item>
</ns0:ValueMappingReplication>
```

For Delete operation:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:ValueMappingReplication xmlns:ns0="http://sap.com/xi/XI/System">
    <Item>
        <Operation>Delete</Operation>
        <GroupID>0ec543a0bb2311dbceb90016e6905679</GroupID>
        <Context>http://sap.com/xi/XI</Context>
        <Identifier scheme="Source1" agency="Source">4100058403</Identifier>
    </Item>
    <Item>
        <Operation> Delete </Operation>
        <GroupID>0ec543a0bb2311dbceb90016e6905679</GroupID>
        <Context>http://sap.com/xi/XI</Context>
        <Identifier scheme="Target1" agency="Target">rcjv</Identifier>
    </Item>
</ns0:ValueMappingReplication>
```

Similarly, the XML message can be created for other operations.

© 2007 SAP AG

3. Reference

The following are the references used:

1. https://help.sap.com/javadocs/NW04S/current/pi/index.html
2. http://help.sap.com/saphelp_nw2004s/helpdata/en/64/7e5e3c754e476ee10000000a11405a/frameset.htm
3. http://help.sap.com/saphelp_nw2004s/helpdata/en/f2/dfae3d47afd652e10000000a114084/frameset.htm
4. http://help.sap.com/saphelp_nw2004s/helpdata/en/13/ba20dd7beb14438bc7b04b5b6ca300/frameset.htm

© 2007 SAP AG

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

© 2007 SAP AG

Filename: How To Perform Value Mapping - A Walkthrough.doc
Directory: C:\Documents and Settings\c5080368\Desktop\links\to be done\2008\february\12
Template: C:\Documents and Settings\c5080368\My Documents\Templates\2007\SDN _ BPX Word Template for Community Contributor_V1.2.dot
Title: Summary
Subject:
Author: Sarath.Chandra
Keywords:
Comments:
Creation Date: 10/15/2007 1:42:00 PM
Change Number: 22
Last Saved On: 2/12/2008 4:07:00 PM
Last Saved By: c5080368
Total Editing Time: 74 Minutes
Last Printed On: 2/12/2008 4:07:00 PM
As of Last Complete Printing
Number of Pages: 19
Number of Words: 2,742 (approx.)
Number of Characters: 16,538 (approx.)