Consuming Directory API in ABAP

Applies to:

SAP ECC 6.0, PI 7.0 – PI 7.1 For more information, visit the Business Process Modeling homepage. .

Summary

This article explains in detail how the directory API services can be consumed from ABAP programs with a special focus on using the “valuemapping” service. This article may also be used as a reference to consume a web service from an ABAP program

Author: Sumana Chakraborty

Company: SAP Global Delivery

Created on: 06 Nov 2008

Author Bio

Sumana Chakraborty is an associate consultant with SAP Global Delivery.

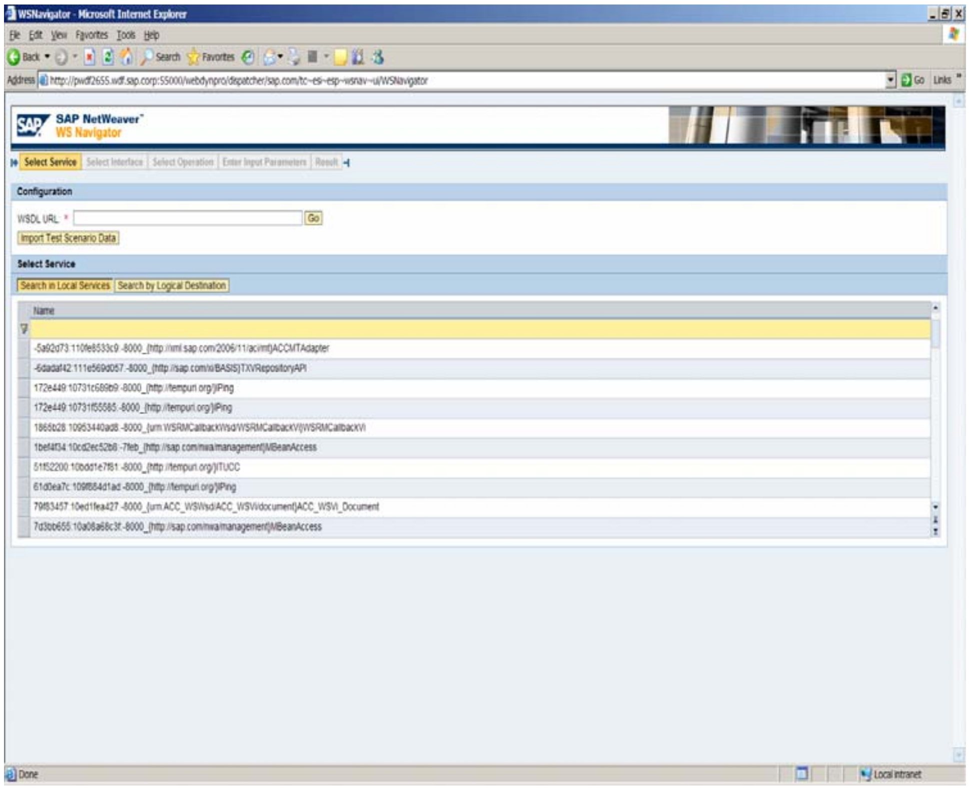
Table of Contents

Locating Directory API in PI

Navigate to the WSNavigator of the PI server using the following link:

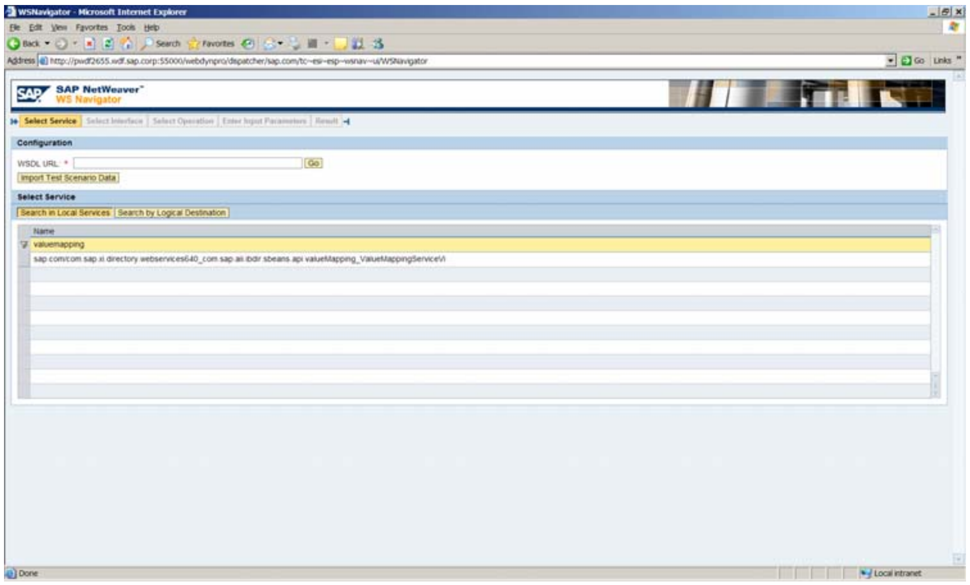
```
http://<host>:<port>/wsnavigator
```

The following screen with all the available web services would appear:


WSNavigator - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://pwdf2655.wdf.sap.corp:55000/webdynpro/dispatcher/sap.com/tc~esi~esp~wsnav~ui/WSNavigator
Go Links
SAP NetWeaver WS Navigator
Select Service | Select Interface | Select Operation | Enter Input Parameters | Result
Configuration
WSDL URL: *
Go
Import Test Scenario Data
Select Service
Search in Local Services
Search by Logical Destination
Done
Local intranet

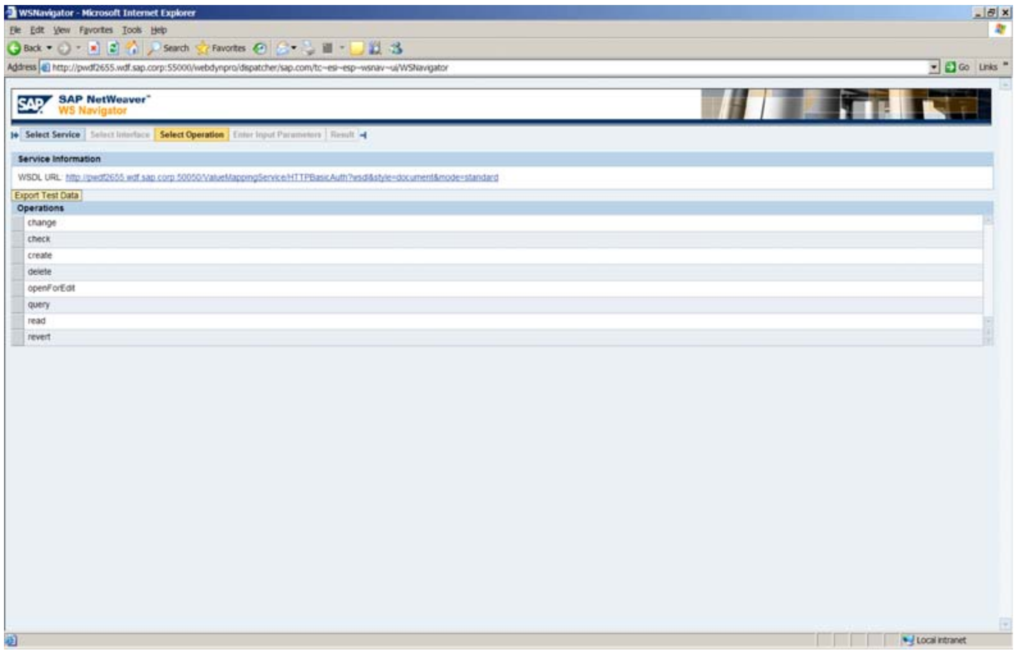
Name
-5a92d73:110fe8533c9:-8000_{http://xml.sap.com/2006/11/aci/mt}ACCMTAdapter
-6dadaf42:111e569d057:-8000_{http://sap.com/xi/BASIS}TXVRepositoryAPI
172e449:10731c689b9:-8000_{http://tempuri.org/}IPing
172e449:10731f55585:-8000_{http://tempuri.org/}IPing
1865b28:10953440ad8:-8000_{urn:WSRMCallbackWsd/WSRMCallbackVI}WSRMCallbackVI
1bef4f34:10cd2ec52b8:-7feb_{http://sap.com/nwa/management}MBeanAccess
51f52200:10bdd1e7f81:-8000_{http://tempuri.org/}ITUCC
61d0ea7c:109f884d1ad:-8000_{http://tempuri.org/}IPing
79f83457:10ed1fea427:-8000_{urn:ACC_WSWsd/ACC_WSVIdocument}ACC_WSVI_Document
7d3bb655:10a08a68c3f:-8000_{http://sap.com/nwa/management}MBeanAccess

1.1. From the list of WebServices, select the ValueMapping service. Click on the Webservice.


WSNavigator - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://pwdf2655.wdf.sap.corp:55000/webdynpro/dispatcher/sap.com/tc~esi~esp~wsnav~ui/WSNavigator
Go Links
SAP NetWeaver WS Navigator
Select Service | Select Interface | Select Operation | Enter Input Parameters | Result
Configuration
WSDL URL: *
Go
Import Test Scenario Data
Select Service
Search in Local Services
Search by Logical Destination
Done
Local intranet

Name
valuemapping
sap.com/com.sap.xi.directory.webservices640_com.sap.aii.ibdir.sbeans.api.valueMapping_ValueMappingServiceVi

1.2. The WebService WSDL URL and the operations are displayed. Clicking on any of the operations will display details regarding the input and output parameters. You may also test the operation.



WSNavigator - Microsoft Internet Explorer

Address http://pwdf2655.wdf.sap.corp:55000/webdynpro/dispatcher/sap.com/tc~esi~esp~wsnav~ui/WSNavigator

SAP NetWeaver WS Navigator

Select Service | Select Interface | Select Operation | Enter Input Parameters | Result

Service Information

WSDL URL: http://pwdf2655.wdf.sap.corp:50050/ValueMappingService/HTTPBasicAuth?wsdl&style=document&mode=standard

Export Test Data

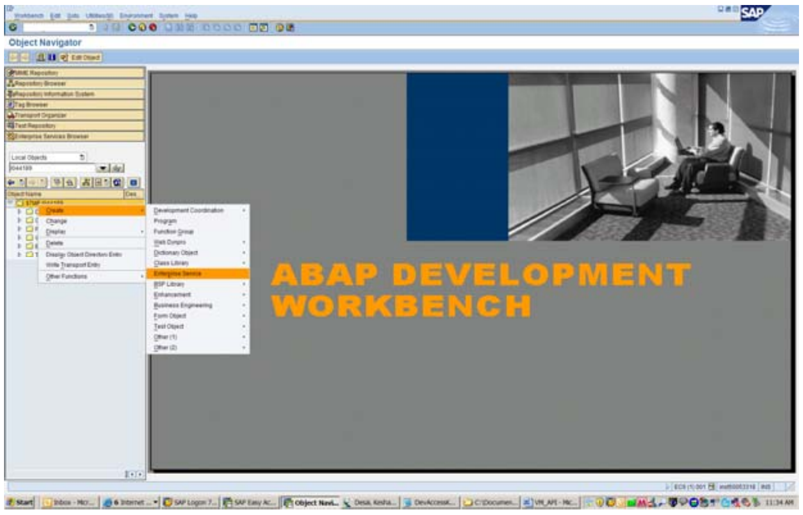
Operations

change
check
create
delete
openForEdit
query
read
revert

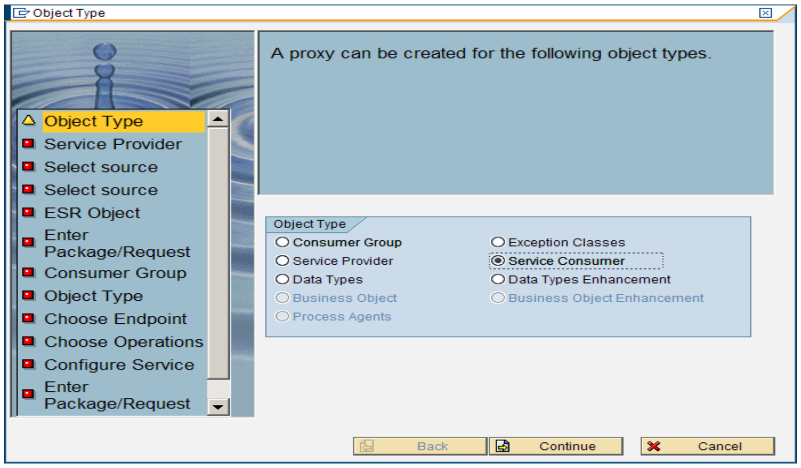
2. Consuming the Directory API in ABAP

Login to the system from which you wish to consume the WebService.

2.1. Run transaction se80. Create a Enterprise Service



2.2. Select Service consumer and click on continue.

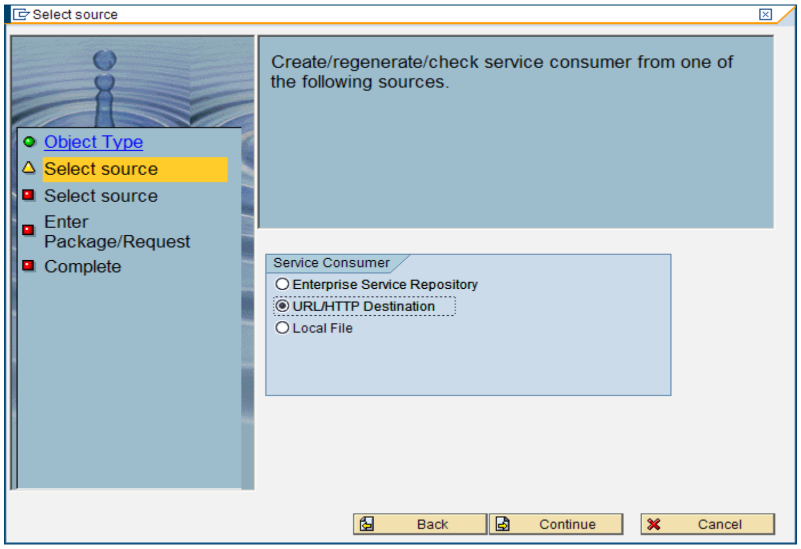


A proxy can be created for the following object types.

Object Type

- Consumer Group
- Service Provider
- Data Types
- Business Object
- Process Agents
- Exception Classes
- [✓ Service Consumer]
- Data Types Enhancement
- Business Object Enhancement

2.3. Select the option Url / HTTP destination, since you have the URL of the WebService

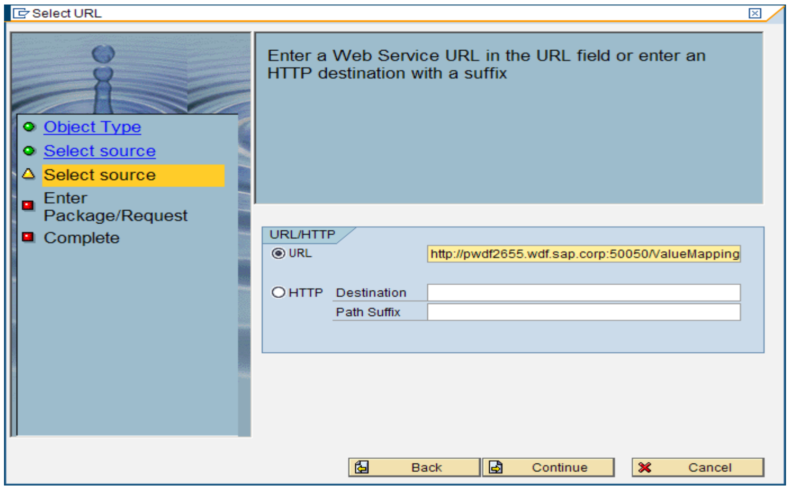


Create/regenerate/check service consumer from one of the following sources.

Service Consumer

- Enterprise Service Repository
- [✓ URL/HTTP Destination]
- Local File

2.4. Provide the URL from WSNavigator:


Select URL

Enter a Web Service URL in the URL field or enter an HTTP destination with a suffix

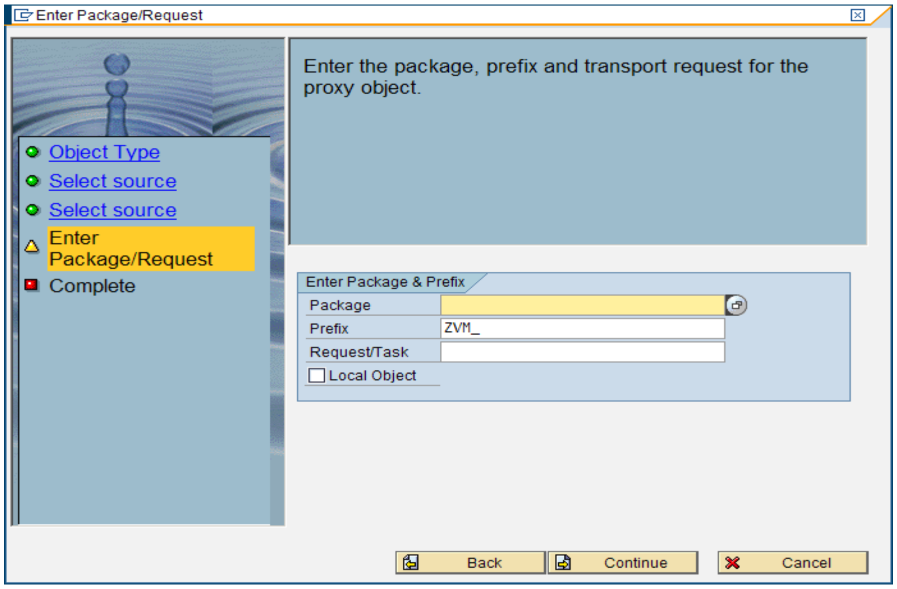
URL/HTTP

(●) URL http://pwdf2655.wdf.sap.corp:50050/ValueMapping

( ) HTTP Destination

Path Suffix

2.5. Provide Package name, prefix and request name. Check the ‘Local Object’ option if you wish to save it as a local object

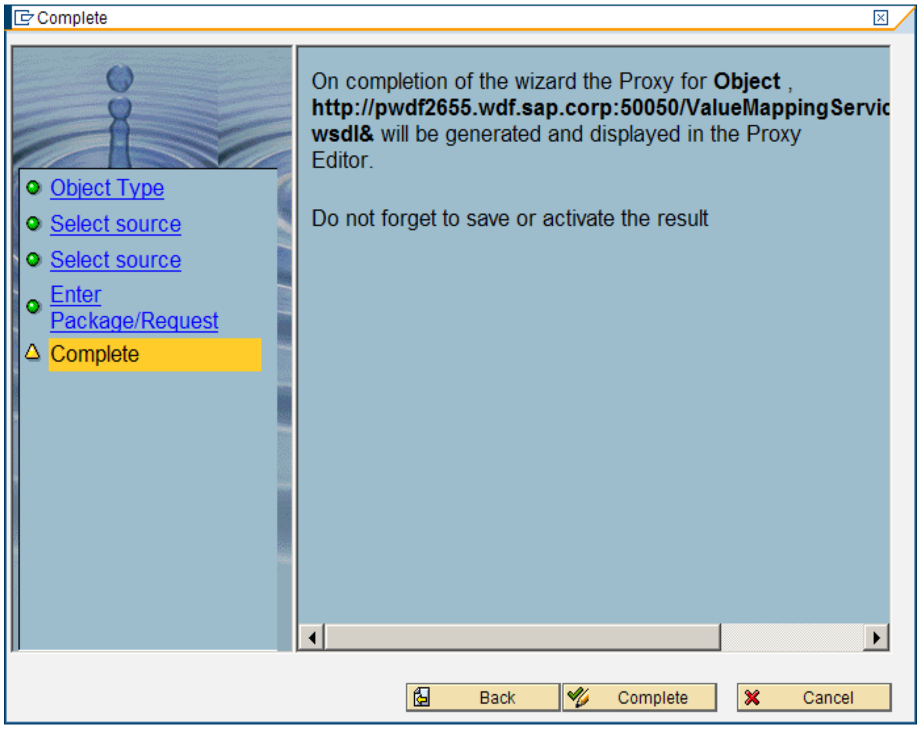

Enter Package/Request

Enter the package, prefix and transport request for the proxy object.

Enter Package & Prefix

Package	
Prefix	ZVM_
Request/Task	
☐ Local Object	

On successful creation the following screen would be displayed.

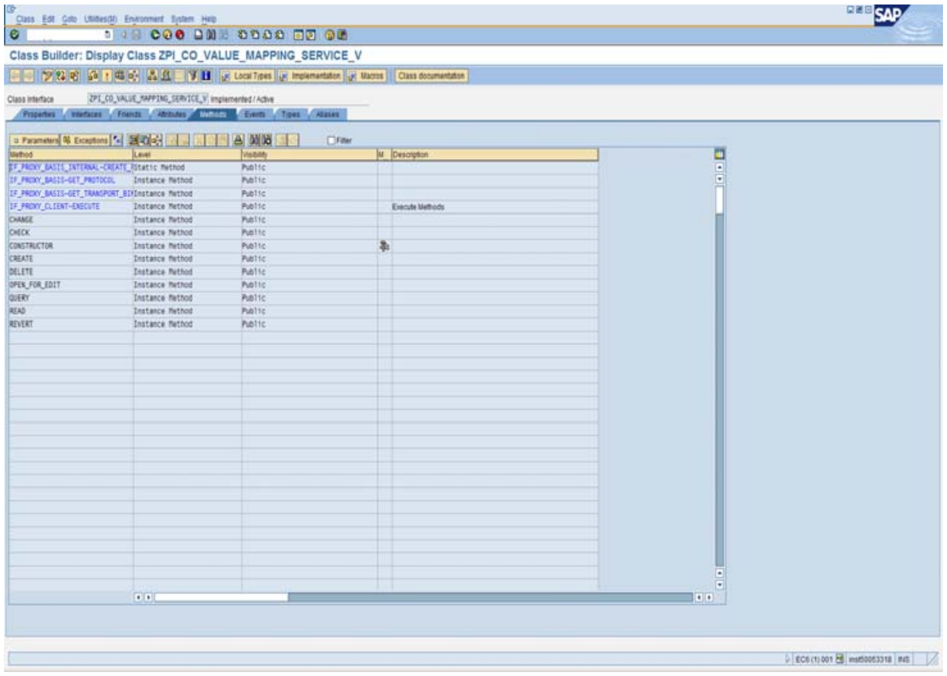


On completion of the wizard the Proxy for **Object** , **http://pwdf2655.wdf.sap.corp:50050/ValueMappingServic** **wsdl&** will be generated and displayed in the Proxy Editor.

Do not forget to save or activate the result

The proxy class would be created in the package which was provided during the generation of the proxy. Save and activate the proxy class.

2.6. The proxy class would have methods corresponding to each operation of the web service



Class Builder: Display Class ZPI_CO_VALUE_MAPPING_SERVICE_V

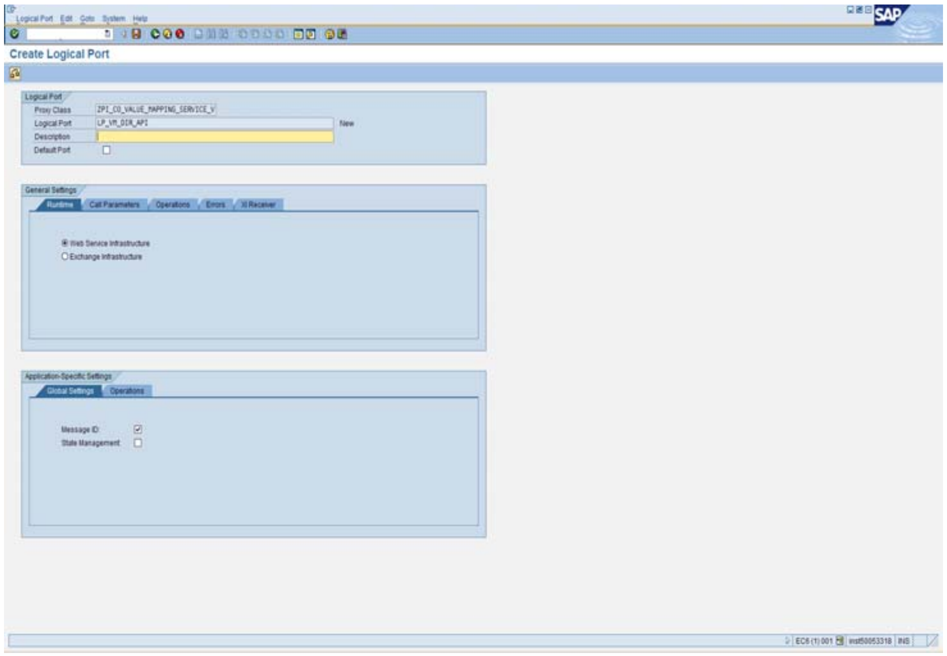
Class interface ZPI_CO_VALUE_MAPPING_SERVICE_V Implemented / Active

Method	Level	Visibility	M	Description
IF_PROXY_BASIS_INTERNAL~CREATE_I	Static Method	Public		
IF_PROXY_BASIS~GET_PROTOCOL	Instance Method	Public		
IF_PROXY_BASIS~GET_TRANSPORT_BI	Instance Method	Public		
IF_PROXY_CLIENT~EXECUTE	Instance Method	Public		Execute Methods
CHANGE	Instance Method	Public		
CHECK	Instance Method	Public		
CONSTRUCTOR	Instance Method	Public		
CREATE	Instance Method	Public		
DELETE	Instance Method	Public		
OPEN_FOR_EDIT	Instance Method	Public		
QUERY	Instance Method	Public		
READ	Instance Method	Public		
REVERT	Instance Method	Public		

3. Creating a Logical Port

To be able to use this WebService we need to create a logical port.

3.1. Go to transaction LPConfig. Specify the Proxy class name and a logical port name. Click on create. Choose web service infrastructure



Create Logical Port

Logical Port

Proxy Class	ZPI_CO_VALUE_MAPPING_SERVICE_V	
Logical Port	LP_VM_DIR_API	New
Description		
Default Port	☐	

General Settings

Runtime | Call Parameters | Operations | Errors | XI Receiver

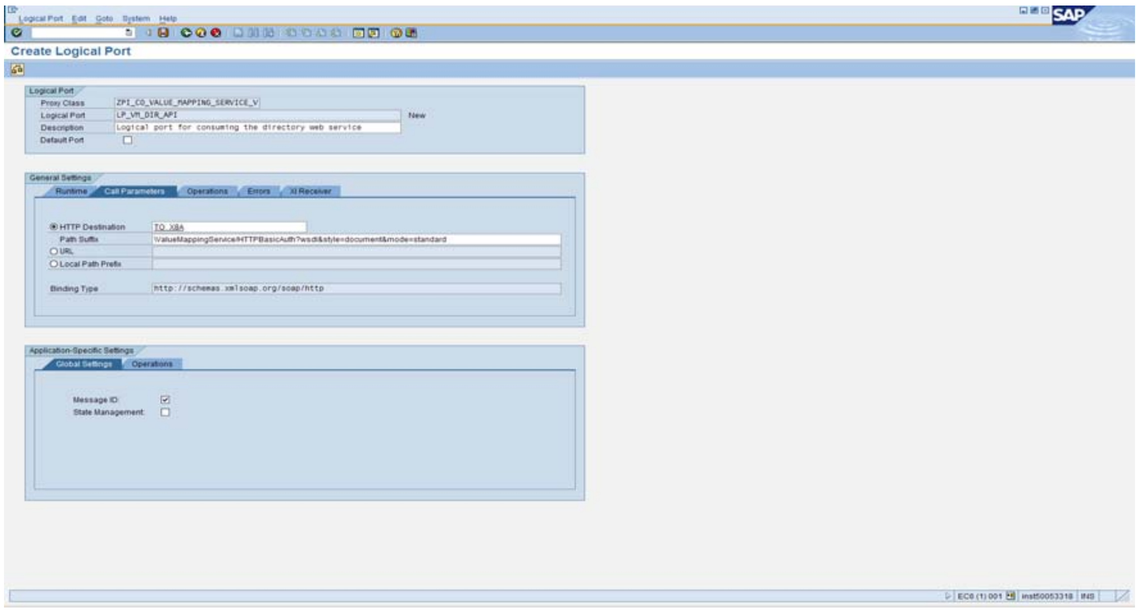
● Web Service Infrastructure
○ Exchange Infrastructure

Application-Specific Settings

Global Settings | Operations

Message ID: ☑
State Management: ☐

3.2. Select the Call Pararmeter tab and provide the HTTP destination and suffix. (Note that you need to have a HTTP destination created to the PI server)



Create Logical Port

Logical Port

Proxy Class	ZPI_CO_VALUE_MAPPING_SERVICE_V	
Logical Port	LP_VM_DIR_API	New
Description	Logical port for consuming the directory web service	
Default Port	☐	

General Settings

Runtime | Call Parameters | Operations | Errors | XI Receiver

● HTTP Destination	TO_XBA
Path Suffix	\ValueMappingService\HTTPBasicAuth?wsdl&style=document&mode=standard
○ URL	
○ Local Path Prefix	
Binding Type	http://schemas.xmlsoap.org/soap/http

Application-Specific Settings

Global Settings | Operations

Message ID: ☑
State Management: ☐

Now you may use the API in your ABAP program.

4. Using the ValueMapping API

The Valuemapping API provides a programming interface for manipulating the ValueMappings created in the integration directory. The following section describes how to use the various operations of the ValueMapping web service.

4.1. API Usage

Upon creation, the proxy class can be used like a typical ABAP class.

To start using the ValueMapping API you would have to create an object of the proxy class. Make a note of the instantiating object of the proxy class that mentions the LOGICAL PORT which we have created in section 3.

```
For Example:
```

```
TRY.
    CREATE OBJECT LOREF
      EXPORTING
        LOGICAL_PORT_NAME = 'TEST_WEBSERVICE_X8A'.
  CATCH CX_AI_SYSTEM_FAULT .

ENDTRY.
```

4.2. Query and Read method

You may invoke the query method to retrieve all value mapping group IDs defined in your integration directory. You may execute the query based on the user detail or the group description. If you do not pass any values as the input parameter, ‘ValueMappingID’, the method would return all the value mapping group ids present in the system.

Once you have the value mapping group ids you could use the read method to fetch more details about the value mapping group representations like the schema, agency and the corresponding value.

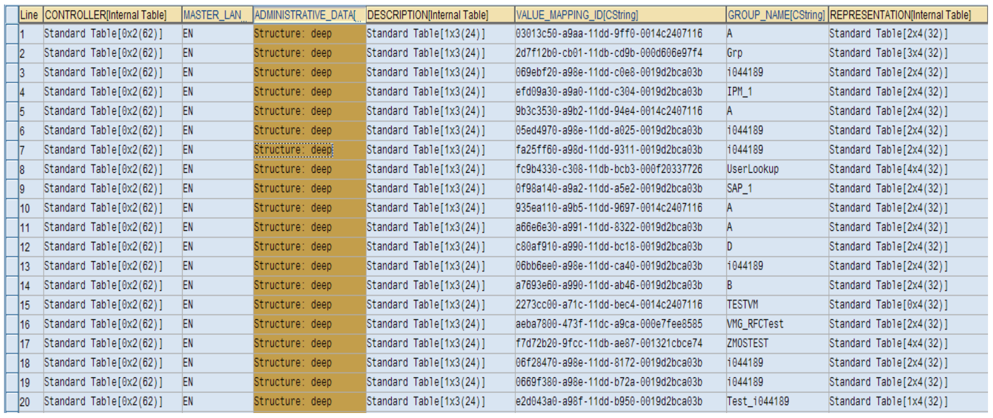
For Example:

The following code snippet fetchs all the group ids and then calls the Read method to get the details of each of the value mapping group.

```
TRY.
    CREATE OBJECT LOREF
      EXPORTING
        LOGICAL_PORT_NAME = 'TEST_WEBSERVICE_X8A'.
  CATCH CX_AI_SYSTEM_FAULT .

ENDTRY.
TRY.

    CALL METHOD LOREF->QUERY
      EXPORTING
        INPUT  = ls_input
      IMPORTING
        OUTPUT = ls_output.

  CATCH CX_AI_SYSTEM_FAULT .
  CATCH CX_AI_APPLICATION_FAULT .

ENDTRY.
```

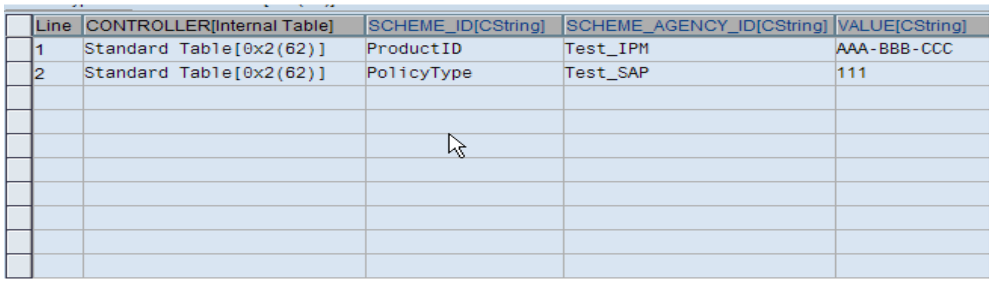
```
* Use the output of the query method and call the read method

    lt_mapping_ids = ls_output-response-value_mapping_ID.

    * Call the read method with all the value mapping ids returned by Query.

    ls_read_input-VALUE_MAPPING_READ_REQUEST-VALUE_MAPPING_ID = lt_mapping_ids.

    TRY.
        CALL METHOD LOREF->READ
          EXPORTING
            INPUT   = ls_read_input
          IMPORTING
            OUTPUT = ls_read_output.

      CATCH CX_AI_SYSTEM_FAULT .
      CATCH CX_AI_APPLICATION_FAULT .
    ENDTRY.
```

Read method returns details of all the value mapping groups like the group name and the group representation as shown in the following figure:



Line	CONTROLLER[Internal Table]	MASTER_LAN...	ADMINISTRATIVE_DATA[...	DESCRIPTION[Internal Table]	VALUE_MAPPING_ID[CString]	GROUP_NAME[CString]	REPRESENTATION[Internal Table]
1	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	03013c50-a9aa-11dd-9ff0-0014c2407116	A	Standard Table[2x4(32)]
2	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	2d7f12b0-cb01-11db-cd9b-000d606e97f4	Grp	Standard Table[3x4(32)]
3	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	069ebf20-a98e-11dd-c0e8-0019d2bca03b	i044189	Standard Table[2x4(32)]
4	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	efd09a30-a9a0-11dd-c304-0019d2bca03b	IPM_1	Standard Table[2x4(32)]
5	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	9b3c3530-a9b2-11dd-94e4-0014c2407116	A	Standard Table[2x4(32)]
6	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	05ed4970-a98e-11dd-a025-0019d2bca03b	i044189	Standard Table[2x4(32)]
7	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	fa25ff60-a98d-11dd-9311-0019d2bca03b	i044189	Standard Table[2x4(32)]
8	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	fc9b4330-c308-11db-bcb3-000f20337726	UserLookup	Standard Table[4x4(32)]
9	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	0f98a140-a9a2-11dd-a5e2-0019d2bca03b	SAP_1	Standard Table[2x4(32)]
10	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	935ea110-a9b5-11dd-9697-0014c2407116	A	Standard Table[2x4(32)]
11	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	a66e6e30-a991-11dd-8322-0019d2bca03b	A	Standard Table[2x4(32)]
12	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	c80af910-a990-11dd-bc18-0019d2bca03b	D	Standard Table[2x4(32)]
13	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	06bb6ee0-a98e-11dd-ca40-0019d2bca03b	i044189	Standard Table[2x4(32)]
14	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	a7693e60-a990-11dd-ab46-0019d2bca03b	B	Standard Table[2x4(32)]
15	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	2273cc00-a71c-11dd-bec4-0014c2407116	TESTVM	Standard Table[0x4(32)]
16	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	aeba7800-473f-11dc-a9ca-000e7fee8585	VMG_RFCTest	Standard Table[2x4(32)]
17	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	f7d72b20-9fcc-11db-ae87-001321cbce74	ZMOSTEST	Standard Table[4x4(32)]
18	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	06f28470-a98e-11dd-8172-0019d2bca03b	i044189	Standard Table[2x4(32)]
19	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	0669f380-a98e-11dd-b72a-0019d2bca03b	i044189	Standard Table[2x4(32)]
20	Standard Table[0x2(62)]	EN	Structure: deep	Standard Table[1x3(24)]	e2d043a0-a98f-11dd-b950-0019d2bca03b	Test_i044189	Standard Table[1x4(32)]

One of the parameters is the ‘Represesntation’ table which has the schema, agency and corresponding value as illustrated below:



Line	CONTROLLER[Internal Table]	SCHEME_ID[CString]	SCHEME_AGENCY_ID[CString]	VALUE[CString]
1	Standard Table[0x2(62)]	ProductID	Test_IPM	AAA-BBB-CCC
2	Standard Table[0x2(62)]	PolicyType	Test_SAP	111

4.3. Create

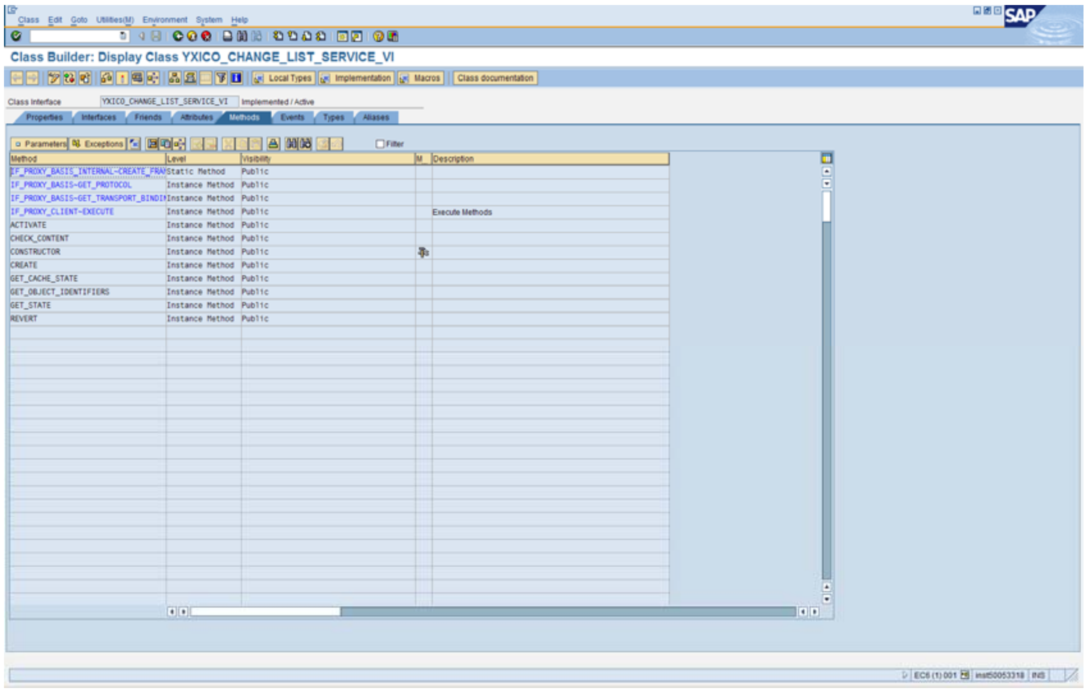
The create operation is used to create a new value mapping group.

The following code snippet is an example of invoking the create operation:

```
To create a new entry in an existing schema and agency:
```

```
ls_input1-MASTER_LANGUAGE = 'EN'.
ls_input1-GROUP_NAME = 'A'.


ls_create_rep-SCHEME_ID = 'PolicyType'.
ls_create_rep-SCHEME_AGENCY_ID = 'Test_SAP'.
ls_create_rep-VALUE = 'III'.
append ls_create_rep to lt_create_rep.

clear ls_create_rep.

ls_create_rep-SCHEME_ID = 'ProductID'.
ls_create_rep-SCHEME_AGENCY_ID = 'Test_IPM'.
ls_create_rep-VALUE = '666'.
append ls_create_rep to lt_create_rep.

ls_input1-REPRESENTATION = lt_create_rep.

append ls_input1 to lt_input1.

ls_input_create-VALUE_MAPPING_CREATE_REQUEST-VALUE_MAPPING = lt_input1.


TRY.
    CALL METHOD LOREF->CREATE
      EXPORTING
        INPUT  = ls_input_create
      IMPORTING
        OUTPUT = ls_output_create.

  CATCH CX_AI_SYSTEM_FAULT .
  CATCH CX_AI_APPLICATION_FAULT .
ENDTRY.
```

To activate the object which is created using the API you would need to use the ChangeList service which is available as a directory API. Therefore, you will have to create a service consumer for the ChangeList service as well.



Class Builder: Display Class YXICO_CHANGE_LIST_SERVICE_VI

Class Interface YXICO_CHANGE_LIST_SERVICE_VI Implemented / Active

Method	Level	Visibility	M	Description
IF_PROXY_BASIS_INTERNAL~CREATE_FRA	Static Method	Public		
IF_PROXY_BASIS~GET_PROTOCOL	Instance Method	Public		
IF_PROXY_BASIS~GET_TRANSPORT_BINDI	Instance Method	Public		
IF_PROXY_CLIENT~EXECUTE	Instance Method	Public		Execute Methods
ACTIVATE	Instance Method	Public		
CHECK_CONTENT	Instance Method	Public		
CONSTRUCTOR	Instance Method	Public		
CREATE	Instance Method	Public		
GET_CACHE_STATE	Instance Method	Public		
GET_OBJECT_IDENTIFIERS	Instance Method	Public		
GET_STATE	Instance Method	Public		
REVERT	Instance Method	Public		

Valuemapping service's Create method's returns a changelist ID that is used to create the object.

You could get that changelistid from the output of the create method and send that to the activate method of the changelist service to activate the valuemapping group that you have created.

```
ls_changelist_input-CHANGE_LIST_ACTIVATE_REQUEST =  ls_output_create-RESPONSE-
CHANGE_LIST_ID-CHANGE_LIST_ID.


TRY.
    CALL METHOD LOREF2->ACTIVATE
      EXPORTING
        INPUT   = ls_changelist_input
*   IMPORTING
*     OUTPUT =
        .
  CATCH CX_AI_SYSTEM_FAULT .
  CATCH CX_AI_APPLICATION_FAULT .
ENDTRY.
```

4.4. Change

This method is used to modify the value mapping group entries. However, before calling this method you would have to call the OpenForEdit method. The OpenForEdit method would return the entire group details including the group representation and a changelist ID. You would have to use the change list ID and the group representation with the modified value to the Change operation.

The following snippet of code gives the steps required to make use the change operation :

```
    *Testing the change operation
    DATA: ls_open_edit_input type ZPI_OPEN_FOR_EDIT_IN_DOC,
          ls_open_edit_output type ZPI_OPEN_FOR_EDIT_OUT_DOC,
          lt_edit_vid type table of string,
          lv_vid type string,
          lv_change_List_ID type string,
          lt_map_dets type ZPI_VALUE_MAPPING_TAB,
          ls_map_det type ZPI_VALUE_MAPPING,
          lv_group_name type string,
          lv_master_lang type string,
          lv_chng_id type string,
          lt_edit_rep type ZPI_VALUE_MAPPING_REPRESEN_TAB,
          ls_edit_rep type ZPI_VALUE_MAPPING_REPRESENTATI,
          ls_change_rep type ZPI_VALUE_MAPPING_REPRESENTATI,
          lt_change_rep type ZPI_VALUE_MAPPING_REPRESEN_TAB.

    DATA: ls_input_change type ZPI_CHANGE_IN_DOC,
          ls_output_change type ZPI_CHANGE_OUT_DOC,
          lt_change_mapping type ZPI_VALUE_MAPPING_RESTRICT_TAB,
          ls_change_mapping type ZPI_VALUE_MAPPING_RESTRICTED.

    * The value mapping group ID from the integration directory to test the API
    lv_vid = '2d7f12b0-cb01-11db-cd9b-000d606e97f4'.

    append lv_vid to lt_edit_vid.

    ls_open_edit_input-VALUE_MAPPING_OPEN_FOR_EDIT_RE-VALUE_MAPPING_ID = lt_edit_vid.
    TRY.
    CALL METHOD LOREF->OPEN_FOR_EDIT
      EXPORTING
        INPUT  = ls_open_edit_input
      IMPORTING
        OUTPUT =  ls_open_edit_output   .
     CATCH CX_AI_SYSTEM_FAULT .
     CATCH CX_AI_APPLICATION_FAULT .
    ENDTRY.

    * Go trough the output of open_for_edit and make the changes you need.
    Then use it as input for the change method .
    lv_change_List_ID = ls_open_edit_output-response-CHANGE_LIST_ID-CHANGE_LIST_ID.

    lt_map_dets = ls_open_edit_output-response-VALUE_MAPPING.



    loop at lt_map_dets into ls_map_det.
      lv_group_name = ls_map_det-group_name.
      lv_master_lang = ls_map_det-MASTER_LANGUAGE.
      lv_chng_id = ls_map_det-VALUE_MAPPING_ID.
      lt_edit_rep = ls_map_det-REPRESENTATION.
    endloop.

    LOOP AT lt_edit_rep into ls_edit_rep.
    move-corresponding ls_edit_rep to ls_change_rep.
    ls_change_rep-value = 'NewVal'.
    append ls_change_rep to lt_change_rep.
    endloop.

    *Prepare the input for the change method.
    ls_input_change-VALUE_MAPPING_CHANGE_REQUEST-CHANGE_LIST_ID = lv_change_List_ID.
    ls_change_mapping-MASTER_LANGUAGE = lv_master_lang.
    ls_change_mapping-GROUP_NAME = lv_group_name.
    ls_change_mapping-VALUE_MAPPING_ID = lv_chng_id.
    ls_change_mapping-REPRESENTATION = lt_change_rep.
    append ls_change_mapping to lt_change_mapping.
```

```
ls_input_change-VALUE_MAPPING_CHANGE_REQUEST-VALUE_MAPPING = lt_change_mapping.

TRY.
CALL METHOD LOREF->CHANGE
  EXPORTING
    INPUT  = ls_input_change
  IMPORTING
    OUTPUT = ls_output_change
    .
 CATCH CX_AI_SYSTEM_FAULT .
 CATCH CX_AI_APPLICATION_FAULT .
ENDTRY.

ls_changelist_input-CHANGE_LIST_ACTIVATE_REQUEST =  ls_output_create-RESPONSE-
CHANGE_LIST_ID-CHANGE_LIST_ID.
TRY.
    CALL METHOD LOREF2->ACTIVATE
      EXPORTING
        INPUT  = ls_changelist_input
*    IMPORTING
*      OUTPUT =
        .
  CATCH CX_AI_SYSTEM_FAULT .
  CATCH CX_AI_APPLICATION_FAULT .
ENDTRY.
```

Related Content

The following weblogs details about the directory API.
https://weblogs.sdn.sap.com/pub/wlg/11655
https://weblogs.sdn.sap.com/pub/wlg/11653
https://weblogs.sdn.sap.com/pub/wlg/11654

SAP Help documentation on Directory API:
http://help.sap.com/saphelp_nwpi71/helpdata/en/46/6dca42e5c269dfe10000000a11466f/content.htm

For more information, visit the Business Process Modeling homepage.

Copyright