

How to Execute MDM Import Process Using MDM JAVA API



Applies to:

SAP NetWeaver MDM 5.5. For more information, visit the Master Data Management homepage.

For more information, visit the [Master Data Management homepage](#).

Summary

This article aims at explaining step by step process of SAP MDM import process using MDM JAVA API

Author: Vinay Swarup

Company: Accenture Services

Created on: 15 September 2009

Author Bio



Vinay Swarup is working as SAP MDM EP Consultant with Accenture Services and has total 4+ years of experience in SAP NetWeaver MDM and EP.

Table of Contents

Overview	3
Scenario Detail	3
Steps Involved	3
Step 1: Creation of Web Dynpro Application	3
Step 2: Code Snippet	4
Step 3: Executing the Application and Checking Result	6
Use	8
Miscellaneous	8
Related Content	10
Disclaimer and Liability Notice	11

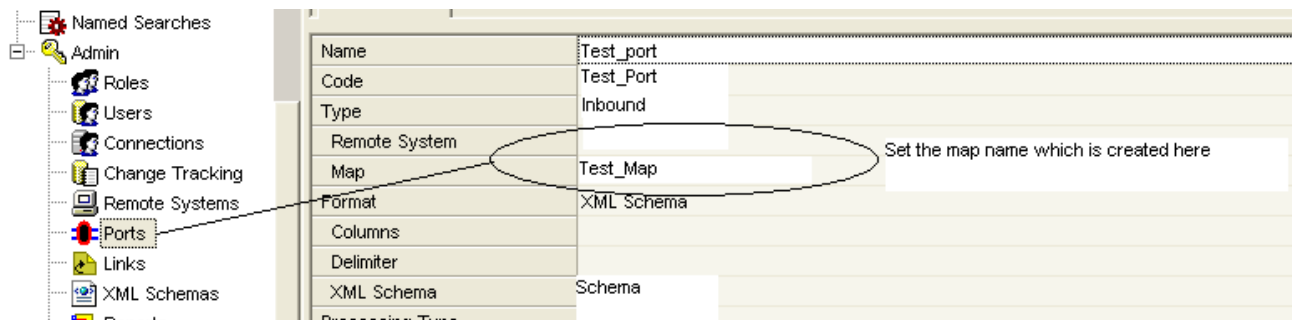
Overview

This article aims at explaining step by step process of SAP MDM import process using MDM JAVA API .It basically describes a scenario where you have created a map for the import process and assigned it to a port. Needless to say MDIS is up and running. Now when PI (Process Integration team) sends the XML file for processing in MDM, MDM JAVA API would do that for us.

Scenario Detail

Let us consider a case where PI is generating XML to be imported in SAP MDM. In this we will create a small web dynpro application and upload the file and then API would import it to MDM .What all we need handy before this exercise:

- MDIS is running. You can check this as a service
- Map is created for import in MDM Import manager and saved
- Port is created in MDM Console and map assigned to it.

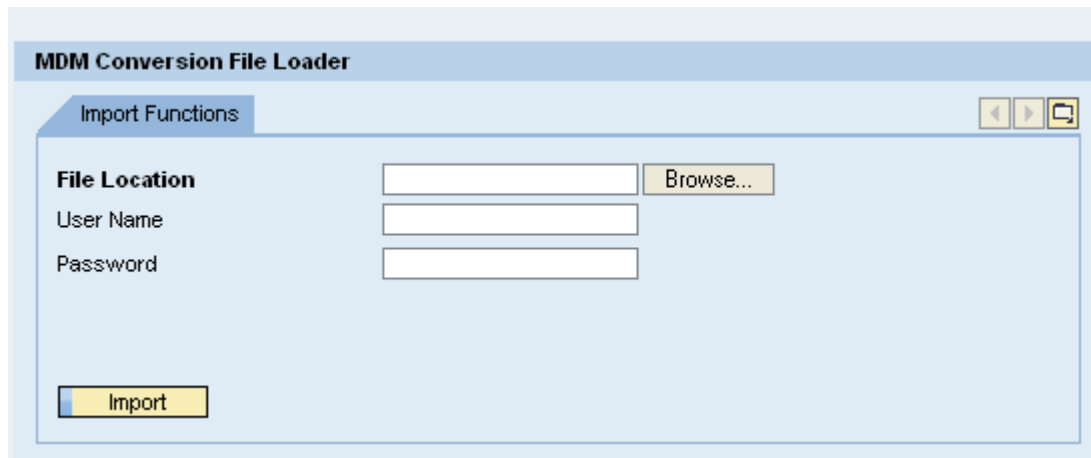


Note: Creation of webdynpro application is not necessary for running MDM JAVA API for importing files. It is just to have GUI for the user

Steps Involved

Step 1: Creation of Web Dynpro Application

You can create a web Dynpro based simple application like the one with a file upload GUI and Import button. My application has logon credentials entries also. Something that would look like this:



The Web Dynpro coding is not in the scope of this document .This application will browse the file and upload it for processing by MDIS (it will submit it to the Ready folder of MDIS port).The Import button is point of action for the user.

Step 2: Code Snippet

The following code is needed for JAVA MDM API which can be written in the onActionImport method associated with the button.

1. Get the file name stored

```
String FileLocation = wdContext.currentContextElement().getFileLocation();
```

2. Now convert this file to bytes as MDM JAVA API for Import would need it

```
byte[] bytes=null;
try {
    File file = new File(Resource.getResourceName().toString());

    InputStream inputStream = Resource.read(true);
    int Size = inputStream.available();
    // Get the size of the file
    long length = (long)Size;
    // You cannot create an array using a long type.
    // It needs to be an int type.
    // Before converting to an int type, check
    // to ensure that file is not larger than Integer.MAX_VALUE.
    if (length > Integer.MAX_VALUE) {
        // File is too large
    }
    // Create the byte array to hold the data
    bytes = new byte[(int)length];

    // Read in the bytes
    int offset = 0;
    int numRead = 0;
    while (offset < bytes.length && (numRead=inputStream.read(bytes, offset, bytes.length-offset)) >= 0) {
        offset += numRead;
    }
    // Ensure all the bytes have been read in
    if (offset < bytes.length) {
        throw new IOException("Could not completely read file "+file.getName());
    }
    // Close the input stream and return bytes
    inputStream.close();

    wdComponentAPI.getMessageManager().reportSuccess(" Byte : " + bytes.length);
}
```

3. Set MDM Server Connection and Fetch Repository session ID

These are all standard MDM JAVA APIs and code for the same is available in the examples section of the following link:

<http://help.sap.com/javadocs/MDM/current/index.html>

```

ConnectionPool con = serverconnection(ServerName);
RepositoryIdentifier rep_id=get_repositoryid(con,Repository,Database,DBMSType.ORACLE);
RegionProperties [] reg = get_regions(rep_id,con);
for(int i = 0;i<reg.length;i++)
System.out.println(reg[i].getRegionCode());
String repession_id= Repositorysession(con,rep_id,get_regions(rep_id,con),Username,Pwd);
String Port="TestPort";

```

name of the port you have
created in Console

- Now use the following API to submit file for processing to MDIS. In this fetching PortId and RemotesystemID is mandatory and Session Id used is repository session ID and not user Session ID

```

GetPortListCommand getport_list = new GetPortListCommand(con);
getport_list.setSession(repession_id);
try {
getport_list.execute();
} catch (CommandException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
// System.out.println(getport_list.isCommandComplete());
PortProperties[] port_prop= getport_list.getPorts();
for(int i = 0;i<port_prop.length;i++)
{
if(Port.equalsIgnoreCase(port_prop[i].getName().toString()))
{
SubmitPortDataFilesCommand submit= new SubmitPortDataFilesCommand(con);
submit.setPortId(port_prop[i].getId());
submit.setRemoteSystemId(port_prop[i].getRemoteSystemId());
submit.setSession(repession_id);

PortDataFile portdata = new PortDataFile(ResourceName,bytes);

PortDataFile [] portdataarray = {portdata};
submit.setFiles(portdataarray);
try {
submit.execute();
wdComponentAPI.getMessageManager().reportSuccess("File is uploaded successfully");
} catch (CommandException e2) {

wdComponentAPI.getMessageManager().reportSuccess(e2.toString());
}
}
}

```

Resource name is any name
for your file.bytes is your
converted file

Step 3: Executing the Application and Checking Result

1. No record in MDM for vendor number 0900000097

Operator	Value
contains	
contains	
contains	
contains	
contains	
contains	
contains	
contains	
contains	0900000097
contains	
contains	

2. Go to the hyperlink of the application and browse file to be uploaded (XML file from XI/PI team)

Use

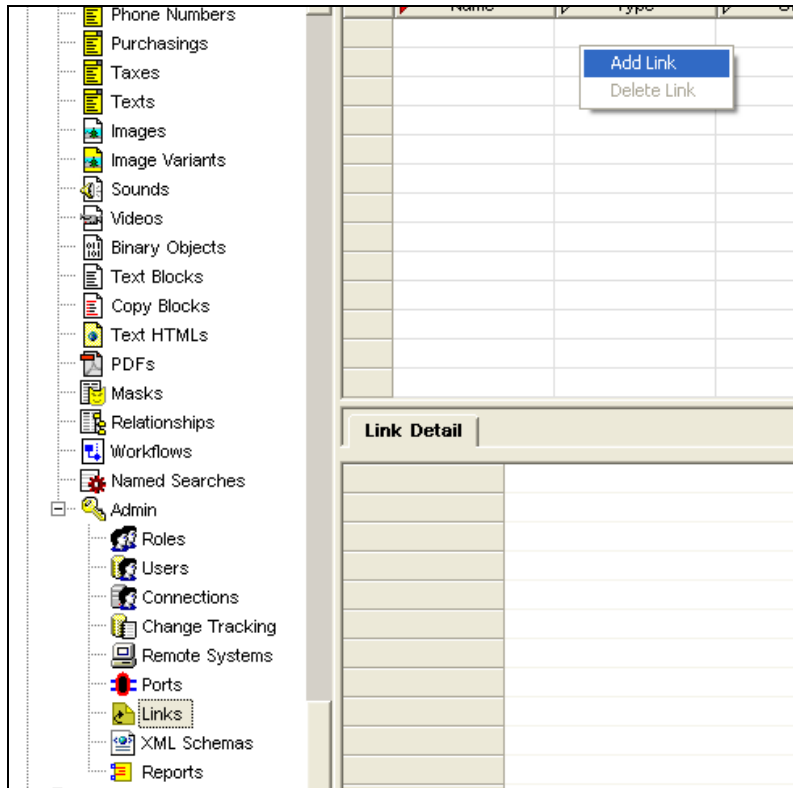
This application can be of great use in the Unit Testing phase where XMLs are processed in BULK and no direct access (FTP) to the MDIS server is given to XI team or where the automated file processing time for XI team to put them in the ready folder is considerably high. In such case it cuts down testing time significantly

Also it can be enhanced and exposed as web service where it will read the files from XI server and place it in the ready folder of MDIS. This scenario is very helpful when there is no FTP access permitted.

Miscellaneous

While testing the application can be integrated in the URL pane and all the xmls can be loaded from Sap MDM Data manager and result can be seen in the Data manager itself.

In console add this hyperlink as web address in Links table under Admin



Related Content

www.help.sap.com

<http://help.sap.com/javadocs/MDM/current/index.html>

www.sdn.sap.com

For more information, visit the [Master Data Management homepage](#).

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.