

ERP-MDM Integration through PI - Part 2: XML to IDoc Conversion



Applies to:

SAP NetWeaver 2004s/ MDM 5.5 SP 06/ SAP PI 7.0 SP10. For more information, visit the [Master Data Management homepage](#).

Summary

This article deals with the step by step procedure to be carried out in PI in order to harmonize master data from MDM to ERP system. For the first part click [here](#).

Author: Shweta Singh

Company: Satyam Computer Services Ltd.

Created on: 17 November 2008

Author Bio



She has been associated with Satyam for 25 months and has been a part of MDM practice since March 2008. She is certified in XI and has completed her Bachelor's degree in Information Technology.

Table of Contents

Introduction	3
Business Scenario	3
Conversion of XML to IDoc.....	4
Integration Repository.....	4
Summary.....	16
Related Contents	17
Disclaimer and Liability Notice.....	18

Introduction

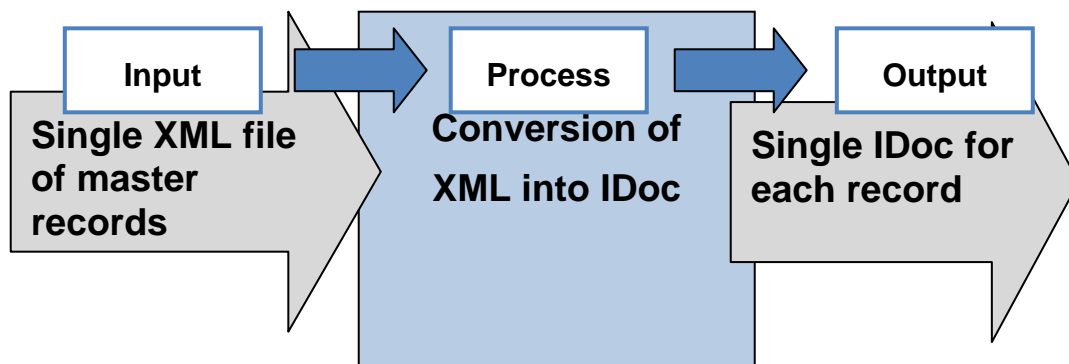
This article provides a step by step guide on the various configurations that need to be done in PI to send the XML file from MDM to PI which in turn will convert the XML file into IDoc and post into the ERP system. This is the second part of my article in which I have tried to explain XML to IDoc Conversion in PI for harmonizing master data. The first part had the details about the IDoc to XML Conversion in order to consolidate master data from ERP system to MDM.

Business Scenario

The scenario consists of 2 systems, ERP and MDM (Master Data Management) that are required to exchange messages through PI. The container that contains the message is an IDoc (Intermediate Document). The XML file consisting of Material, Customer or Vendor data is picked from the specified folder and sent from MDM to PI which further converts it into an IDoc and posts in the ERP system.

Pre requisite: Make sure that before starting with the PI specific configurations, the Integration (ALE) Settings between the ERP and the MDM system to establish connection between the 2 systems and the SLD(Software Landscape Directory) settings like configuring of Software Product, Software Component ,Technical and Business Systems in PI are already done.

Conversion of XML to IDoc

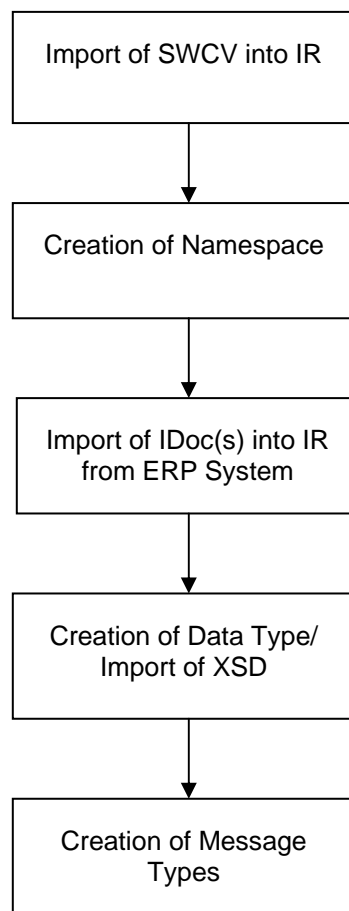


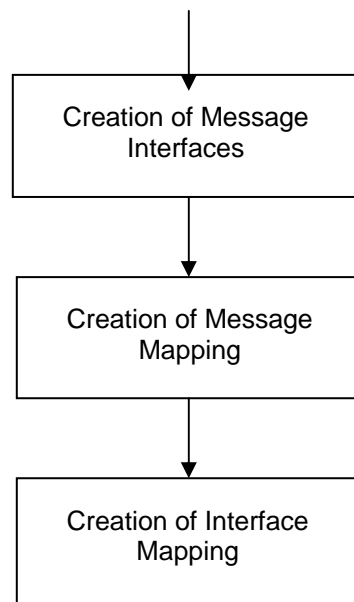
This article is in continuation with my article that I had posted on ERP- MDM Integration : Part 1.

The 1st part dealt with the configurations that need to be done in the PI system in order to post the IDocs from the ERP system to the MDM folder(IDoc to XML Conversion).The 2nd part deals with posting the IDocs back into the ERP system from MDM(XML to IDoc Conversion).

The sequence of steps in Integration Repository and Integration Directory is exactly the same here as in IDoc to XML Conversion. Only the Source and Target structures need to be changed, and of course the Sender and the Receiver systems too.

Integration Repository



**Steps:**

1. Import of SWCV from SLD into IR
2. Creation of Namespace
3. Import of IDoc from ERP System
4. Creation of Data Types
5. Creation of Message Types
6. Creation of Message Interfaces
7. Creation of Message Mapping
8. Creation of Interface Mapping

1. Import of objects and Creation of namespace in IR**1.1. Creation of Namespace in IR**

1.1.1. Considering that you have already imported the SWCV while doing IDoc to XML Conversion, the first step here is to create the namespace to hold the design objects relevant to this scenario together.

1.1.2. The procedure is the same : Open the SWCV, under the *Namespaces* tab, create a namespace and give it a suitable name(eg: `http://xmltoidoc`).

1.2. Import of IDocs from ERP system into PI system

1.2.1. Here, the IDoc that has to be imported will serve as the target structure.

Note: The Target IDoc here can't be a mass IDoc, the reason being ERP system doesn't accept mass IDOCs since there is no existing Process Code to process the same. Hence if we have the structure of DEBMDM or CREMDM in the source XML file, in PI we have to split it into individual IDocs that is DEBMAS (or CREMAS) and ADRMAS which have existing Process Codes. Hence, for this scenario we need to import DEBMAS or CREMAS and ADRMAS into the PI system from ERP system.

2. Designing Interface Objects

2.1. Creation of Source Data type/Import of Source External Definition

2.1.1. Here, we need to create the Data type for the source structure or import the structure under *External Definitions* depending on the availability of the source XSD.

2.1.2. The procedure for it is the same as that in Part 1.

2.2. Import of Target External Definition

2.2.1. Since the source structure has to be split into many IDocs depending on the number of records in the source file, the occurrence of the IDoc needs to be changed from one to unbounded.

That is done by exporting the IDoc first into local disk. The occurrence of the IDocs is changed as shown. It is then imported back as an External Definition into PI.

Display External Definition Status Active

Name: DEBMAS06_ED

Namespace: http://mdmpocfile2idoc

Software Component Version: MDMPOCCOMP.SATYAM.COM , 1.0 of satyam

Description:

Category: xsd Messages From All Available Global Elements

File *: DEBMAS.DEBMAS06.xsd

Source:

Imported Document Messages WSDL External References

Search:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="DEBMAS06">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="IDOC" type="DEBMAS.DEBMAS06" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="DEBMAS06.E1WRF6M">
    <xsd:annotation>
      <xsd:documentation>
        Segment for plant/material groups
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="MSGFN" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>
            Function
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

DEBMDM06_E... MI_CustomerM... DEBMAS06_E... ADRMAS03_E... EHS_XI, 1.0 of...


2.3. Creation of Message Interface

2.3.1.A Message Interface is created using the above Message Type.

2.3.2.It has to be created only for the source structure if Data Type and the Message Type have been already created for the the same.

2.3.3.For the Message Interface that is created here, the direction will be Outbound since it will be sending the message to PI.

Display Message Interface		Status	Acti
Name	MI_CustomerMaster_send		
Namespace	http://mdmpocfile2idoc		
Software Component Version	MDMPOCCOMP.SATYAM.COM , 1.0 of satyam		
Description			

Definition	Context Objects	WSDL	
Attributes			
Category	<input type="radio"/> Inbound	<input checked="" type="radio"/> Outbound	<input type="radio"/> Abstract
Mode	<input type="radio"/> Synchronous	<input checked="" type="radio"/> Asynchronous	
Message Types			
Output Message	Type Name of External Message *	Namespace of External Message *	
	 DEBMDM06		

3. Designing Mapping Objects

3.1. Creation of Message Mapping

3.1.1. There have to be two Message Mappings here, one for mapping address details(ADRMAS) and the other for general details(DEBMAS or CREMAS say).

3.1.2. Here the source structure has to be selected as the Message Type or the External Definition and the target structure as the IDoc.

3.1.3. Since we want different IDocs for different customer records, map E1EDKA1 to IDOC n once the occurrence is unbounded it will be created for every occurrence of the E1EDKA1.

3.1.4. Again, depending on the requirement, map some or all of the fields in both the mappings. Also do take care of the fields that are mandatory.

3.1.5. Test the mappings for any errors.

3.1.6. The two screenshots below show the mappings for ADRMAS and DEBMAS respectively.

Display Message Mapping Status Active

Name: MM_CustomerMaster_to_ADRMAS
 Namespace: http://mdmpocfile2idoc
 Software Component Version: MDMPOCCOMP.SATYAM.COM , 1.0 of satyam
 Description:

Design Test Messages

External Message: DEBMDM06 External Message: ADRMAS03

Tree	Occurrences	Type
[DEBMDM06]	1..1	
[IDOC]	1..1	DEBMDM.DEB.
BEGIN	required	xsd:string
[EDI_DC40]	0..1	EDI_DC40.DEB.
[E1KNA1M]	1..unbounded	DEBMDM06.E1
SEGMENT	required	xsd:string
[MSGFN]	0..1	xsd:string

Tree	Occurrences	Type
[ADMAS03]	1..1	
[IDOC]	1..unbounded	
BEGIN	required	xsd:string
[EDI_DC40]	1..1	
[E1ADRMAS]	0..1	
SEGMENT	required	xsd:string
[OBJ_TYPE]	0..1	

Display Message Mapping Status Active

Name: MM_CustMaster_to_Debmas
 Namespace: http://mdmpocfile2idoc
 Software Component Version: MDMPOCCOMP.SATYAM.COM , 1.0 of satyam
 Description:

Design Test Messages

External Message: DEBMDM06 External Message: DEBMAS06

Tree	Occurrences	Type
[DEBMDM06]	1..1	
[IDOC]	1..1	DEBMDM.DEB.
BEGIN	required	xsd:string
[EDI_DC40]	0..1	EDI_DC40.DEB.
[E1KNA1M]	1..unbounded	DEBMDM06.E1
SEGMENT	required	xsd:string
[MSGFN]	0..1	xsd:string

Tree	Occurrences	Type
[DEBMAS06]	1..1	
[IDOC]	1..unbounded	DEBMAS.DEBM.
BEGIN	required	xsd:string
[EDI_DC40]	1..1	EDI_DC40.DEB...
[E1KNA1M]	1..1	DEBMAS06.E1...
SEGMENT	required	xsd:string
[MSGFN]	0..1	xsd:string

3.2. Creation of Interface Mapping

- 3.2.1. There are two Interface Mappings here, one for ADRMAS and one for DEBMAS.
- 3.2.2. Again, the Interface Mappings have to be created as described earlier which will have the Inbound and the Outbound interfaces along with the mapping programs used.
- 3.2.3. The two screenshots below show the Interface mappings for ADRMAS and DEBMAS respectively.

The screenshot displays the 'Display Interface Mapping' configuration in SAP PI. The mapping is named 'IM_CustomerMaster_to_ADRMAS' and is currently 'Active'. It is associated with the namespace 'http://mdmpocfile2idoc' and software component version 'MDMPOCCOMP.SATYAM.COM , 1.0 of satyam'.

The configuration is shown in the 'Design' tab, which is divided into three main sections:

- Source Interface:** Lists the source interface 'MI_CustomerMaster_send' with namespace 'http://mdrMDMPOC1'.
- Target Interface:** Lists the target interface 'ADRMAS.ADRMAS03' with namespace 'urn:sap-c:MDMPOC1'.
- Message Interface:** A pop-up window showing the message interface 'MI_CustomerMaster_send' with namespace 'http://mdmpocfile2idoc' and software component version 'MDMPOCCOMP.SATYAM.COM , 1.0 of satyam'.

Below the interface configuration, the 'Read Interfaces' section shows the 'Request' flow:

- Source Message:** 'DEBMDM06' is selected.
- Mapping Program:** A table lists the mapping program 'MM_CustomerMaster_to_ADRMAS' with namespace 'http://mdmpoc'.
- Target Message:** 'ADRMAS.ADRMA' is selected.

Display Interface Mapping Status: Active

Name: IM_CustomerMaster_to_DEBMAS

Namespace: http://mdmpocfile2idoc

Software Component Version: MDMPOCCOMP.SATYAM.COM , 1.0 of satyam

Description:

Design | **Test**

Source Interface *

Name	Names...	Softwar...	Occurr...
MI_CustomerMaster_send	http://mdrMDMPOC1		

Target Interface *

Name	Names...	Softwar...	Occurr...
DEBMAS.DEBMAS06	urn:sap-cMDMPOC1		

Read Interfaces

Request

Source Message

DEBMDM06 →

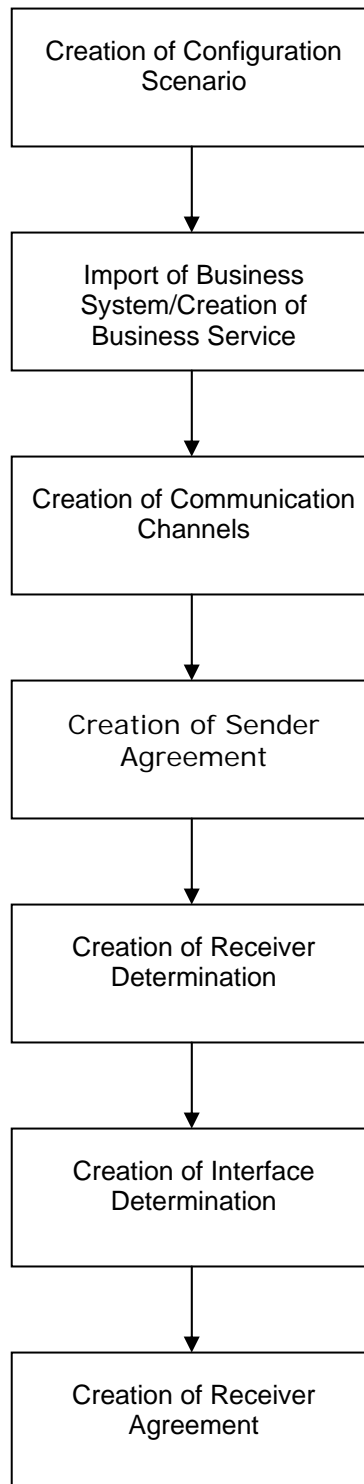
Mapping Program *

Type	Name	Namespace
Message M...	MM_CustMaster_to_Debmas	http://mdmpoc

Target Message

→ DEBMAS.DEBMA

Integration Directory



Steps:

1. Creation of Configuration Scenario
2. Import of Business System/Creation of Business Service
3. Creation of Communication Channels
4. Creation of Sender Agreement
5. Creation of Receiver Determination
6. Creation of Interface Determination
7. Creation of Receiver Agreement

1. Creation of Configuration Scenario

1.1. Create a configuration Scenario in order to group the ID objects together.

2. Import of Business System/ Creation of Business Service(s)

2.1. Again, if you have a Business System as the sender or the receiver, import it in ID as explained earlier in the first article.

2.2. Also, create the Business Service(s) as required.

3. Creation of Communication Channels

3.1. Two communication channels are required here, one for Sending and the other for Receiving.

3.2. The Sender Communication channel will use the File Adapter. So give all the necessary details like the Source Directory from where the file has to be picked up. The file name(not mandatory, you can give *.xml instead of a particular filename. Hence whenever a file will be placed in that path, it will be picked up by the Sender Communication Channel) etc.

3.3. The Receiver Communication Channel will use the IDoc Adapter to post IDocs into the ERP System. Details like RFC Destination to be used, Port, SAP Release are to be specified.

The screenshot displays the SAP Communication Channel configuration window. The 'Display Communication Channel' tab is active, showing details for channel 'CC_FILE1' with status 'Active'. The 'Parameters' tab is selected, showing the following configuration:

- Adapter Type:** File
- Transport Protocol:** File Transfer Protocol (FTP)
- Message Protocol:** File
- Adapter Engine:** Integration Server
- File Access Parameters:**
 - Source Directory: /CUSTOMER_SP04/Outbound/CGNSAP33MDM/CGNSAP33MDM_
 - File Name: R3_XI_MDM_CUST.xml
 - Advanced Selection for Source File:
- FTP Connection Parameters:**
 - Server: psimdm
 - Port: 21
 - Data Connection: Active
 - Timeout (secs):
 - Connection Security: None

Communication Channel Edit View

Display Communication Channel Status Active

Communication Channel CC_IDOC

Party

Service BS_MDM_ECC

Description

Parameters Identifiers Module

Adapter Type * IDoc http://sap.com/xi/XI/System SAP BASIS 7.00

Sender Receiver

Transport Protocol * IDoc

Message Protocol * IDoc

Adapter Engine * Integration Server

RFC Destination * ECC_510

Segment Version

Interface Version * SAP Release 4.0 or Higher

Port * SAPECC

SAP Release * 700

Queue Processing

Apply Control Record Values from Payload

Take Sender from Payload

Take Receiver from Payload

Restore Original Parties for Acknowledgments

4. Creation of Sender Agreement

- 4.1. Next is to create the Sender Agreement.
- 4.2. For this, go to *Sender Agreement->New*. Select the Sender Service, Sender Interface and the Receiver Service.
- 4.3. Also, select the Sender Communication Channel.
- 4.4. Save.

5. Creation of Receiver Determination

- 5.1. Go to *Receiver Determination->New*.
- 5.2. Choose the appropriate Sender and Receiver services, the Outbound Interface etc.
- 5.3. Save.

6. Creation of Interface Determination

- 6.1. Next is to create the Interface Determination.
- 6.2. Go to *Interface Determination->New*.
- 6.3. Here two inbound interfaces have to be selected, one each for ADRMAS and DEBMAS.
- 6.4. Save.

Interface Determination Edit View

Display Interface Determination Status Active

Sender

Party

Service BS_MDM_ECC

Interface MI_CustomerMaster_send

Namespace http://mdmpocfile2idoc

Receiver

Party

Service BS_MDM_ECC

Description

Type of Interface Determination **Quality of Service**

Standard Enhanced Maintain Order At Runtime

Configured Inbound Interfaces

Inbound Interface		Interface Mapping	
Name	Namespace	Name	Namespace
1 ADRMAS.ADRMAS03	urn:sap-com:document:sap	IM_CustomerMaster_to_ADRMA	http://mdmpocfile2idoc
2 DEBMAS.DEBMAS06	urn:sap-com:document:sap	IM_CustomerMaster_to_DEBMA	http://mdmpocfile2idoc

7. Creation of Receiver Agreements

7.1. The last step is to create the Receiver Agreement.

7.2. Go to *Receiver Agreement->New*.

7.3. We have two receiver Agreements here, one for ADRMAS and the other for DEBMAS.

7.4. Select the appropriate values for the other fields. Also, select the Receiver Communication channel that will be used in each agreement.

7.5. The two screenshots below show the Receiver agreements for ADRMAS and DEBMAS respectively.

Receiver Agreement Edit View

Display Receiver Agreement Status Active

Sender

Party

Service BS_MDM_ECC

Receiver

Party

Service BS_MDM_ECC

Interface ADRMAS.ADRMAS03

Namespace urn:sap-com:document:sap:idoc:messages

Description

Receiver Communication Channel * CC_IDOC

Header Mapping

Sender Party

Sender Service

Receiver Party

Receiver Service

Receiver Agreement Edit View

Display Receiver Agreement Status Active

Sender

Party

Service BS_MDM_ECC

Receiver

Party

Service BS_MDM_ECC

Interface DEBMAS.DEBMAS06

Namespace urn:sap-com:document:sap:idoc:messages

Description

Receiver Communication Channel * CC_IDOC

Header Mapping

<input type="checkbox"/> Sender Party	<input type="text"/>	
<input type="checkbox"/> Sender Service	<input type="text"/>	
<input type="checkbox"/> Receiver Party	<input type="text"/>	
<input type="checkbox"/> Receiver Service	<input type="text"/>	

Note: For Glossary, kindly refer 1st part of this article, titled ERP-MDM Integration through PI - Part 1: IDoc To XML Conversion.

Summary

Through this article we have seen the various steps that need to be followed in PI to build an XML to IDoc scenario for exchange of messages between ERP and MDM systems.

Related Contents

[ERP-MDM Integration through PI, Part 1: IDoc to XML Conversion](#)

<https://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/9757>

<https://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/1819>

<https://help.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.