

SAP XI/PI: 1..N Message Split Interface without BPM



Applies to:

SAP XI 3.0, PI 7.0 and 7.1.

For more information, visit the [Repository-based Modeling and Design homepage](#).

Summary

This paper discusses about a catch in SAP XI/PI for generating multiple output messages with a single 1..N mapping function thus sending the messages to the destination system independently.

Author: VijayaShankar Konam (VJ for the SDN Community Members☺)

Company: FUJITSU Consulting (USA)

Created on: 09 December 2008

Author Bio



VijayaShankar Konam is an SAP certified NetWeaver (SAP XI/PI) consultant at Fujitsu Consulting (USA). He has prior experience in Microsoft .NET technologies and SQL Server 2005. He has been recognized as one of the Top Contributors 2008 group at sdn.sap.com.

Table of Contents

Introduction	3
The Problem	3
The Catch	4
Recap.....	7
Related Content.....	7
Disclaimer and Liability Notice.....	8

Introduction

I would like to discuss a simple technique to achieve a very common scenario in this article.

A requirement like multiple structures (combination of Header and Line Items repeating) from the source might need to be split into separate messages (Set of Header and Multiple Line items) on the target side. To achieve this scenario, one would normally think of using 1..N mapping in the message mapping and thus producing multiple messages for the target. But creation of a 1..N message mapping mandates 1..N Interface mapping and henceforth the inevitable BPM.

Can't we do it without a BPM?

The answer is Yes. I would like to demonstrate this in detail in the following sections.

The Problem

When one creates a mapping with the target message cardinality as 1..N, XI constraints the user to create an Interface mapping in which the target interface cardinality must also be 1..N.

When one tries to create 1..1 Interface mapping with 1..N mapping in it, XI does not let you activate the interfaces.

1..N Message Map

Tree	Occurrences	Type	Details	De
Messages	1..1			
Message1	1..1			
DELVRY03	1..1			
IDOC	1..1	DESADV.DELV...		
BEGIN	required	xsd:string		
EDI_DC40	1..1	EDI_DC40.DES...		
E1EDL20	1..9999	DELVRY03.E1E...		

Tree	Occurrences	Type
Messages	1..1	
Message1	1..1	
Shipment_Delivery_Data	0..unbounded	p0:Shipment_D...
Header	1..1	
Line	0..unbounded	

1..1 Interface Map

Name	Namespace	Software Compo...	Occurrence
DESADV.DELVRY03	urn:sap-com:docur:TESTSWC , 1.0 of p 1		

Name	Namespace	Software Compo...	Occurrence
ia_ShipmentDeliveryData_Receive	urn.pelican.com:ForTESTSWC , 1.0 of p 1		

Type	Name	Namespace
Message Mapping	DESADVDELVRY03_2_ShipmentDeliveryData	urn.pelican.com:ForMultiFileGe

XI Error

Processing Log

Activation of the change list canceled

Check result for **Interface Mapping DESADVDELVRY03_2_ShipmentDeliveryData** | urn: .com:ForMultiFileGen:

⚠ Mapping program **Message Mapping DESADVDELVRY03_2_ShipmentDeliveryData** | urn: .com:ForMultiFileGen does not match the interface mapping. The number or frequencies of source or target messages for the message mapping are not identical to the number or frequencies of source or target interfaces.

Since XI does not let us create a 1..1 interface mapping, we would finally end up creating a 1..N interface mapping and that makes the requirement of BPM inevitable (Sadly, 1..N mappings are only supported by BPM).

The Catch

For realizing this requirement, One has to create a 1..1 mapping first and then a subsequent 1..1 Interface mapping.

1..1 Message Mapping

The screenshot shows two side-by-side tables representing the message mapping. The left table is for the source IDoc, and the right table is for the target message type.

Tree	Occurrences	Type	Details	Descri
IDOC	1..1	DESADV.DELV...		
BEGIN	required	xsd:string		
ED1_DC40	1..1	ED1_DC40.DES...		
E1EDL20	1..9999	DELVRY03.E1E...		
SEGMENT	required	xsd:string		
VBELN	0..1	xsd:string	maxLength...	Sales
VSTEL	0..1	xsd:string	maxLength...	Shippi
VKORG	0..1	xsd:string	maxLength...	Sales
LSTEL	0..1	xsd:string	maxLength...	Loadin
VKBUR	0..1	xsd:string	maxLength...	Sales
LGNUM	0..1	xsd:string	maxLength...	Wareh
ABLAD	0..1	xsd:string	maxLength...	Unloac
INCO1	0..1	xsd:string	maxLength...	Incoter
INCO2	0..1	xsd:string	maxLength...	Incoter

Tree	Occurrences	Type	Details	De
Shipment_Delivery_Data	1..1	p0:Shipment_D...		
Header	1..1			
Code	1..1	xsd:string	maxLength...	
Customer_Code_3PL	1..1	xsd:string	maxLength...	
Shipment_Number	1..1	xsd:string	maxLength...	
Prepaid_Collect	1..1	xsd:string	maxLength...	
Forwarding_Agent	1..1	xsd:string	maxLength...	
Separator1	1..1	xsd:string	maxLength...	Hip
Forwarder_Phone	1..1	xsd:string	maxLength...	
Separator2	1..1	xsd:string	maxLength...	Hip
Inco_Terms1	1..1	xsd:string	maxLength...	
Separator3	1..1	xsd:string	maxLength...	Sin
Inco_Terms2	1..1	xsd:string	maxLength...	
Invoice_Cust_Num	1..1	xsd:string	maxLength...	

Now, save this message mapping in to a mapping template for using at the later time.

The screenshot shows the 'Create Template Based on Mapping' dialog box in the SAP XI/PI Message Mapping tool. The dialog is open, and the 'Design' tab is selected. The source IDoc and target message type are visible in the background.

Create a 1..1 Interface mapping and activate it.

The screenshot shows the SAP XI/PI Interface Mapping tool. The 'Source Interface' and 'Target Interface' are defined, and a 'Mapping Program' is created for the interface mapping.

Name	Namespace	Software Comp...	Occurrence
DESADV.DELVRY03	urn:sap-com:docur:TESTSWC , 1.0 of p 1		

Name	Namespace	Software Comp...	Occurrence
ia_ShipmentDeliveryData_Receive	urn.pelican.com:ForTESTSWC , 1.0 of p 1		

Type	Name	Namespace
Message Mapping	DESADVDELVRY03_2_ShipmentDeliveryData	urn.pelican.com:ForMultiFileGe

Source Message	Target Message
DESADV.DELVRY03	Shipment_Delivery_Data

You should observe no errors.

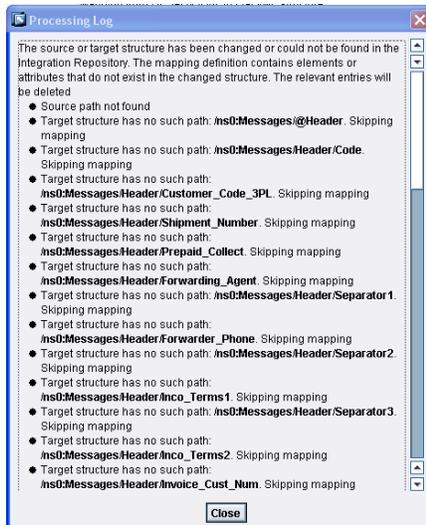
Now comes the tricky part. Keeping the 1..1 interface mapping as it is, change the message mapping to 1..N.

The screenshot shows the SAP XI/PI Message Mapping tool with the 'Messages' tab selected. The 'Source Message(s)' and 'Target Message(s)' are defined, and the message mapping is set to 1..N.

Object Type	Name	Namespace	Occurrence
IDoc	DESADV.DELVRY03	urn:sap-com:document:sap:idoc:messages	1

Object Type	Name	Namespace	Occurrence
Message Type	Shipment_Delivery_Data	urn.pelican.com:ForMultiFileGen	0..unbounded

XI would complain that the target message mapping will no longer be valid because the target root node changed.



Now we can use the template that we have created previously to restore the mapping.



Click on  in the mapping control to choose the template.



Now save the mapping and activate.

XI still thinks that the 1..1 interface mapping is still referring to a 1..1 message mapping. Now that we are done with the repository objects, move on to Directory objects.

Note: I would not be showing all the steps needed here except for the content conversion parameters of the receiver File adapter as an example. One can use any type of the receiver adapter for this. Everything is normal nothing tricky here..!!

Create the objects normally as we do for any of the asynchronous XI scenario.

Select file content conversion as the Message protocol.

Parameters	Identifiers	Module
Adapter Type *	File	
<input type="radio"/> Sender	<input checked="" type="radio"/> Receiver	
Transport Protocol *	File System (NFS)	
Message Protocol *	File Content Conversion	
Adapter Engine *	Integration Server	

Configure the FCC parameters normally

Target	Processing	Content Conversion	Advanced														
Content Conversion Parameters																	
Recordset Structure * Header,Line																	
<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Header.fieldFixedLengths</td> <td>2,4,15,10,16,1,13,1,3,1,13,10,35,35,25,30,8,10,35,35,25,30,8,50,15,15,30,20,9,3,...</td> </tr> <tr> <td>Header.endSeparator</td> <td>'\n'</td> </tr> <tr> <td>Header.fixedLengthTooShortHandling</td> <td>Cut</td> </tr> <tr> <td>Line.fieldFixedLengths</td> <td>2,15,9,5,10,6,15,15</td> </tr> <tr> <td>Line.endSeparator</td> <td>'\n'</td> </tr> <tr> <td>Line.fixedLengthTooShortHandling</td> <td>Cut</td> </tr> </tbody> </table>				Name	Value	Header.fieldFixedLengths	2,4,15,10,16,1,13,1,3,1,13,10,35,35,25,30,8,10,35,35,25,30,8,50,15,15,30,20,9,3,...	Header.endSeparator	'\n'	Header.fixedLengthTooShortHandling	Cut	Line.fieldFixedLengths	2,15,9,5,10,6,15,15	Line.endSeparator	'\n'	Line.fixedLengthTooShortHandling	Cut
Name	Value																
Header.fieldFixedLengths	2,4,15,10,16,1,13,1,3,1,13,10,35,35,25,30,8,10,35,35,25,30,8,50,15,15,30,20,9,3,...																
Header.endSeparator	'\n'																
Header.fixedLengthTooShortHandling	Cut																
Line.fieldFixedLengths	2,15,9,5,10,6,15,15																
Line.endSeparator	'\n'																
Line.fixedLengthTooShortHandling	Cut																

Run the interface. Surprisingly, XI creates multiple files for each of the Header and Line Items pairs from the source.

If the receiver side is a SOAP adapter, the web service would be called as many times as the number messages got split from the source.

Recap

To create an XI interface which can split the source in to multiple messages and send to the target system, BPM is not mandatory.

In order to create a 1..1 Interface mapping that points to a 1..N message mapping, one has to first create 1..1 message mapping and Interface mapping and activate the changes. Then we can change the mapping to 1..N to split the source in to multiple messages.

This scenario can be extended to any type of receiver adapter, so that the receiver would be called as many times the number of messages gets generated in the target message.

Examples could be like calling a web service for each of the data line or pairs of Header and Line Items from an IDoc etc.

Congratulations! You now have an interface which generates multiple output messages with a single configuration scenario and without BPM.

Related Content

For more information, visit the [Repository-based Modeling and Design 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.