

Multi Mapping in SAP Process Integration 7.1



Applies to:

SAP PI 7.1 (including Enhancement Package1), SP4 onwards.

Summary

This guide aims to acquaint the reader on how to perform Multi-mapping in SAP PI 7.1.

Author: Abhishek Salvi

Created on: 10 April 2010

Author Bio



Abhishek Salvi is a SAP XI/ PI developer working for an Indian IT company. He began his career in Jan-2007 and since then has been working in SAP XI/PI. His area of expertise include SAP XI/ PI and BPM.

Table of Contents

Introduction	3
Scenario Description.....	3
Scenario Execution	3
Multiple Operations in same Inbound Service Interface	4
Multiple Inbound Service Interfaces	8
Related Content	9
Disclaimer and Liability Notice.....	10

Introduction

With the introduction of SAP PI7.1, the option of *Enhanced* Interface Determination vanished. Many thought that SAP PI7.1 does not come with a support for multi-mappings as the earlier version(s) did. This assumption is however not true; SAP PI7.1 supports multi-mapping as much as the earlier versions did. The procedure to achieve multi-mapping is described in this document.

Scenario Description

There are **N** number of records in the source file, these N records should be segregated according to the Location of the concerned employee and send to same/ different target servers. This document showcases the following aspects of multi-mapping:

1. Multiple Operations in same Service Interface
2. Multiple target Service Interfaces

Scenario Execution

Source message type will have the below mentioned format and during testing the MT_SourceFile will repeat, with each occurrence having an employee record.

Name	Category	Type	Occurrence
▼ MT_SourceFile	Element	DT_SourceFile	
▼ EmployeeDetails	Element		0..1
▼ Name	Element		0..1
FirstName	Element	xsd:string	0..1
MiddleName	Element	xsd:string	0..1
LastName	Element	xsd:string	0..1
EmployeeID	Element	xsd:string	0..1
Location	Element	xsd:string	0..1

Fig1. Source Message Structure

Depending on the Location of the employee, different target files (with different message structure) will be created.

MT_TargetFile1 will be created for Location = India

Name	Category	Type	Occurrence
▼ MT_TargetFile1	Element	DT_TargetFile1	
▼ EmployeeDetails	Element		0..1
▼ Name	Element		0..1
FirstName	Element	xsd:string	0..1
MiddleName	Element	xsd:string	0..1
LastName	Element	xsd:string	0..1
EmployeeID	Element	xsd:string	0..1
Location	Element	xsd:string	0..1

Fig2. Target Message Structure

MT_TargetFile2 will be created for Location ≠ India

Name	Category	Type	Occurrence
▼ MT_TargetFile2	Element	DT_TargetFile2	
▼ EmpDetails	Element		0..1
Name	Element	xsd:string	0..1
EmpID	Element	xsd:string	0..1
Location	Element	xsd:string	0..1

Fig3. Target Message Structure

Multiple Operations in same Inbound Service Interface

From SAP PI7.1, a new feature has been provided to include multiple messages in the same outbound/inbound Service Interface. This feature is present with the name **Operation**.

Multiple Operations can be added to a Service Interface whose Interface Pattern is **not** Stateless (XI30-Compatible). Stateless (XI30-Compatible) pattern does not allow for the addition of multiple Operations. In our example, the Interface Pattern of source (outbound) Service Interface can have any (of the available) value, restriction is only on the target (inbound) Service Interface.

Maintaining multiple Operations in the same inbound Service Interface will help in sending the output files, of multi-mapping program, to the same receiver channel i.e. to the same target location. The logic required to generate/ omit the target messages need to be implemented in the mapping program itself.

Below figure shows the service Interface configuration for MT_TargetFile1:

The screenshot displays the configuration for a service interface in SAP PI7.1. The 'Matching Service Interfaces' tab is active, showing the following details:

- Attributes:**
 - Category: Inbound
 - Interface Pattern: Stateless
- Operations:**
 - SI_TargetFile1
 - SI_TargetFile2
- Operation SI_TargetFile1 Details:**
 - Description: Message Type for Employee Location = India
 - Attributes:
 - Operation Pattern: Normal Operation
 - Mode: Asynchronous
 - Messages:

Role	Type	Name	Namespace
Request *	Message Type	MT_TargetFile1	urn:multimappingpoc:pi
Fault	Fault Message Type		

Fig4. Operation and corresponding Message Type for Target-File1

Below figure shows the service Interface configuration for MT_TargetFile2:

Attributes

Category: Inbound

Interface Pattern: Stateless

Operations

Operation: SI_TargetFile2

Description: Message Type for Employee Location other than India

Attributes

Operation Pattern: Normal Operation

Mode: Asynchronous

Messages

Role	Type	Name	Namespace
Request	Message Type	MT_TargetFile2	urn:multipmappingpoc:pi

Fig5. Operation and corresponding Message Type for Target-File2

Below is the screenshot of the message mapping designed to create output files based on the value(s) of the *Location* field in the MT_SourceFile message.

Message Type: MT_SourceFile

Structure	Occurrences	Type	De
Messages	1..1		
Message1	1..1		
MT_SourceFile	0..unbounded	p5:DT_SourceFile	
EmployeeDetails	0..1		
Name	0..1		
FirstName	0..1	xsd:string	
MiddleName	0..1	xsd:string	
LastName	0..1	xsd:string	
EmployeeID	0..1	xsd:string	
Location	0..1	xsd:string	

Target Messages

Structure	Occurrences	Type
Messages	1..1	
Message1	1..1	
MT_TargetFile1	0..unbounded	p5:DT_TargetFile1
EmployeeDetails	0..1	
Name	0..1	
FirstName	0..1	xsd:string
MiddleName	0..1	xsd:string
LastName	0..1	xsd:string
EmployeeID	0..1	xsd:string
Location	0..1	xsd:string
Message2	1..1	
MT_TargetFile2	0..unbounded	n5:DT_TargetFile2

Mapping Logic:

```

graph LR
    Location[Location] --> equalsS[equalsS]
    Constant[Constant India] --> equalsS
    ns1_MT_S[ns1:MT_S...] --> if{if}
    equalsS --> if
    if -- then --> ns1_MT_S
    if -- else --> ns1_MT_TargetFile1[ns1:MT_TargetFile1]
  
```

Fig6. Mapping logic to generate MT_TargetFile1

Mapping logic shown in above figure will create the target structure when Location = India.

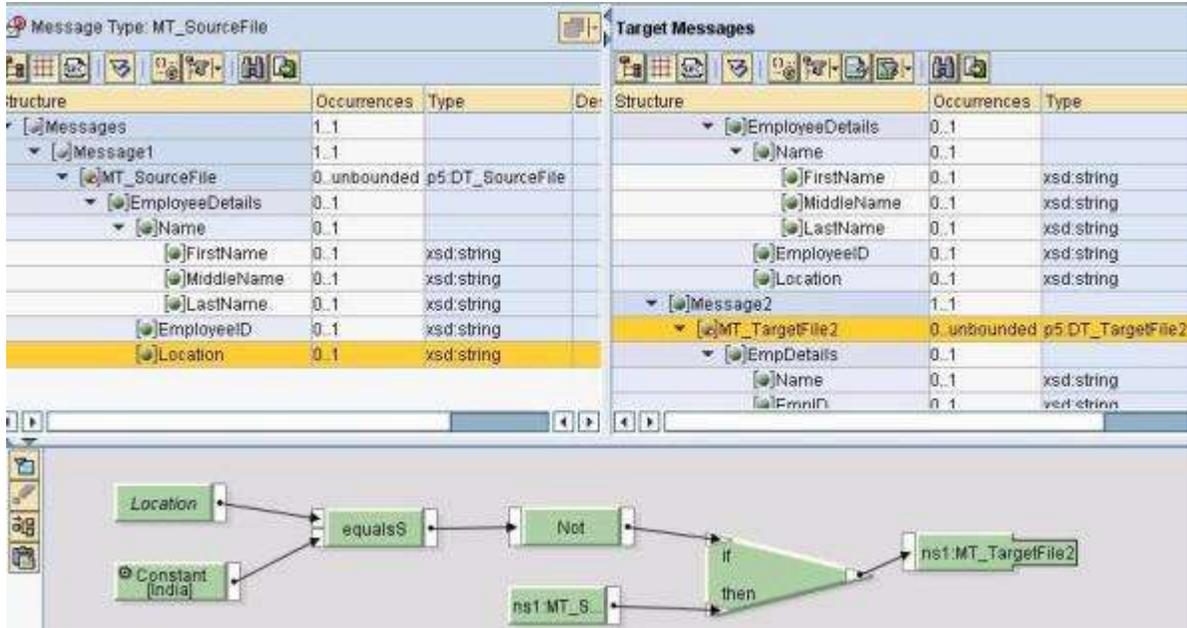


Fig7. Mapping logic to generate MT_TargetFile2

Mapping logic shown in above figure will create the target structure when Location ≠ India.

In the Signature tab of our mapping program, the occurrence of both Source and Target messages need to be adjusted properly; the output of multi-mapping program will not produce desired output if the occurrence value is not changed (default is 1..1), for more information check the below picture:



Fig8. Message Mapping Signature

Similarly the occurrence of the Service Interfaces should be set to 0..Unbounded in Operation Mapping. Only in Operation Mapping we will be able to view the Operations included in a Service Interface. For the considered example, the Operations SI_TargetFile1 and SI_TargetFile2 will be seen in the Target Operation section of Operation Mapping. This behavior is shown in the below figure:

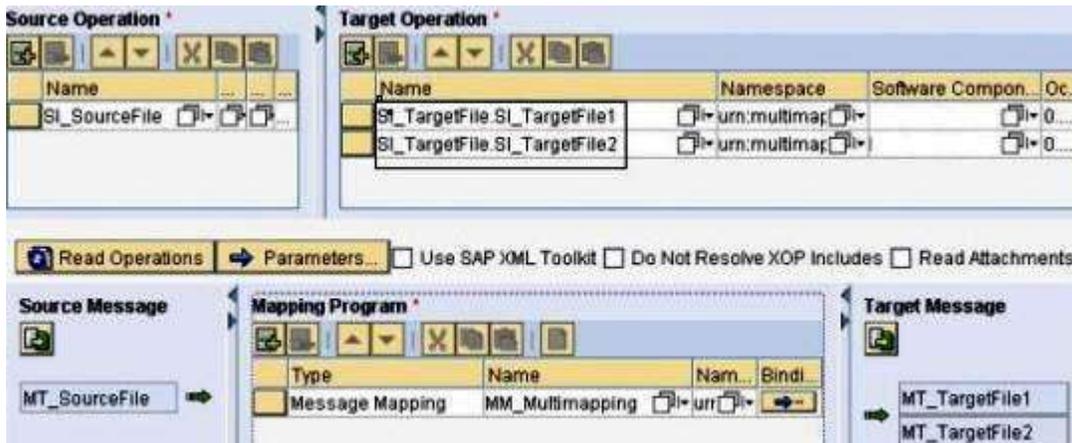


Fig9. Operation Mapping Definition

Once all the changes are saved and activated in Enterprise Services Repository, the next step will be to do configuration in Integration Directory. For this example, only one Receiver FILE communication channel will be used to send the output files to the target folder.

The below figure depicts how the Interface Determination is configured for this particular example of multi-mapping.

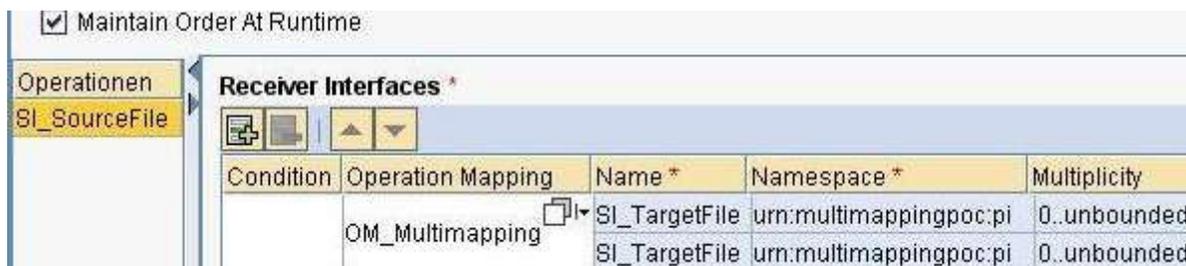


Fig10. Interface Determination

The number of Receiver Interfaces shown in Interface Determination will be equal to the number of interfaces on the target-end of the Operation Mapping. Accordingly, the above picture shows two entries for the service interface, SI_TargetFile, one for Operation SI_TargetFile1 and one for Operation SI_TargetFile2. The entries will reflect only the name of the parent Service Interface and not for the included Operations. Once the desired Operation Mapping is selected, the entries for receiver Service Interface will be populated automatically.

Configuration of all other ID objects viz., Sender Channel & Agreement, Receiver Determination, Receiver Channel & Agreement is done as per the normal procedure.

Once all the objects are created, saved and activated, the interface can be tested by sending an appropriate message through the sender communication channel.

If the source message satisfies the mapping validations, then equivalent number of target files will be generated, one each for each record in source file.

Multiple Inbound Service Interfaces

In the multi-mapping scenario, when the target message(s) needs to be delivered to different locations using same/ different protocols we need to create the relevant number of Service Interfaces; reusing the same Service Interface is not possible in this case.

In this case we will send the output files to two different target directories. To achieve this functionality, create two different Service Interfaces. These Service Interfaces will have the corresponding Message Types assigned to them. In this case, the Service Interface can be created with any Interface Pattern and will not have a restriction as it was in [Example 1](#).

For the scope of this example, no change is required to be made in the Message mapping and it will remain the same as it was shown in [Example 1](#).

Operation Mapping will now have both the Service Interfaces included as target interfaces. The minimum and maximum occurrences of the involved source and target Service Interfaces should be adjusted properly in the Operation Mapping. The below figure will describe how the Operation Mapping is configured:

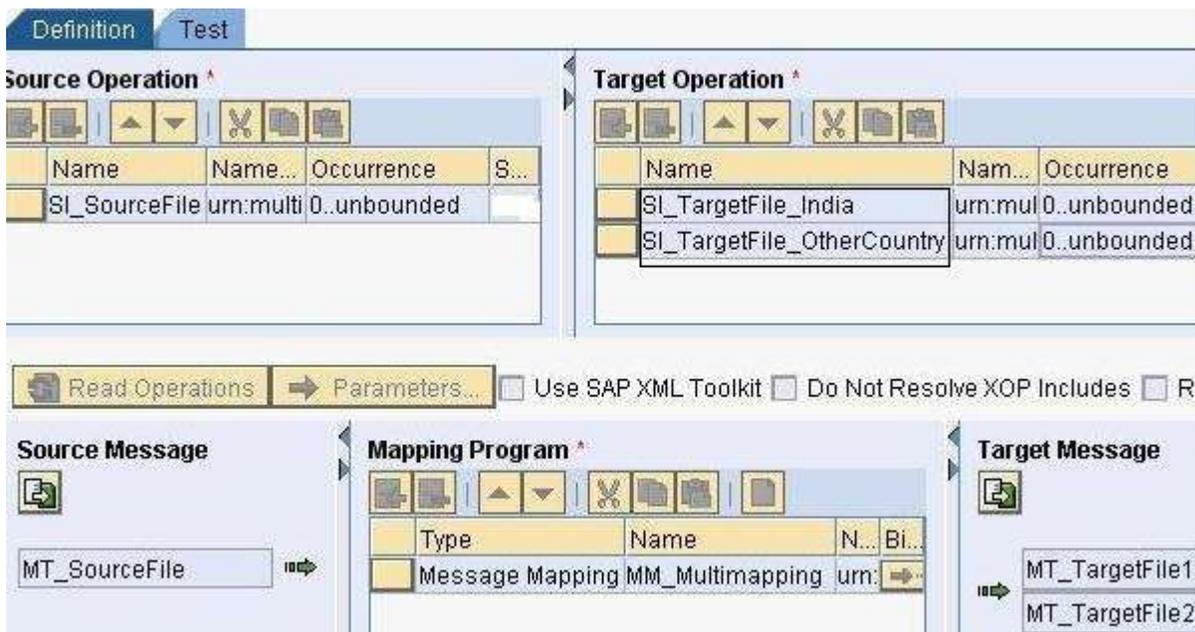


Fig11. Operation Mapping

When different Service Interfaces are used to send files to different target locations, we have to create different Receiver Agreements. Accordingly, each Receiver Agreement will refer to a different Service Interface and also to a different communication channel. For the scope of this example, two receiver FILE communication channels are created to deliver Target-File1 and Target-File2 to respective locations.

Interface Determination in this case will show the entries for both the receiver Service Interfaces involved in the mapping program. During runtime, any or both of the target messages will be created as per the mapping validations and will be routed to appropriate receiver communication channels.



Fig12. Interface Determination

In any case of multi-mapping, if more than one target messages are created then SXMB_MONI will collectively refer these messages as InterfaceCollection under the Receiver Interface column. Also the namespace-entry for this Receiver Interface will be <http://sap.com/xi/XI/System>. However when only one target message is created, the name of the inbound Service Interface will be seen in SXMB_MONI along with the corresponding namespace. This behavior can be seen in the below figure of SXMB_MONI:

XML Messages					
Status	Sender Component	Sender Inter.	Receiver Component	Receiver Namespace	Receiver Interface
	BusComp_MMSEnd	SI_SourceFile	BusComp_MMReceive	http://sap.com/xi/XI/System	InterfaceCollection
	BusComp_MMSEnd	SI_SourceFile	BusComp_MMReceive	urn:multimappingpoc:pi	SI_TargetFile_India

Fig13. Message Processing in SXMB_MONI

Related Content

[Developing Multi-Mappings for Message Splits on help.sap.com](#)

[Service Interface on help.sap.com](#)

[Blog: Using Service Interfaces? Now Reuse One!](#)

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.