Value Validation of XML File Using Java Mapping

Applies to:

SAP NetWeaver™ Exchange Infrastructure/Process Integration 7.0.

Summary

This article illustrates how to validate inbound xml file using Java Mapping.

Author: Santhosh Kumar V.

Company: Wipro Technologies.

Created on: 26 June 2008

Author Bio

Santhosh Kumar is a SAP Netweaver XI/PI Consultant at Wipro Technologies.

Table of Contents

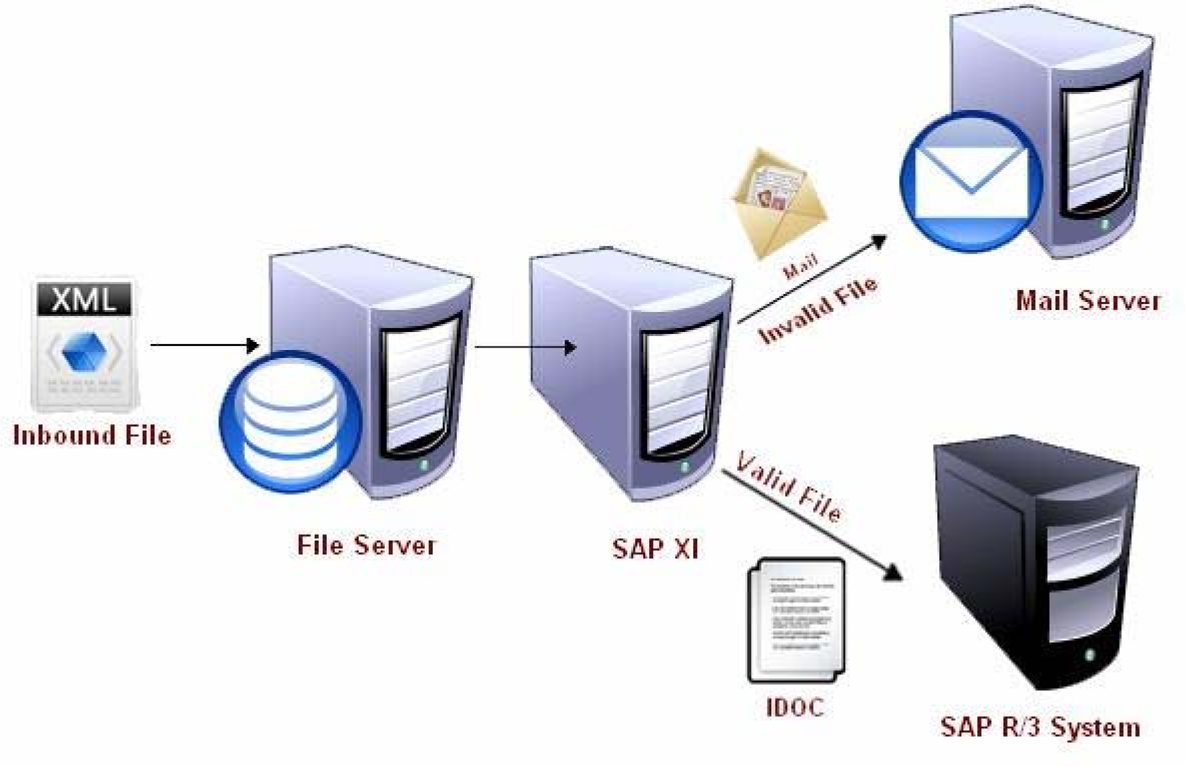
Introduction

This article illustrates how an inbound xml file can be validated and rejected by validating the values of the XML fields. A rejection mails can also triggered intimating the file rejection. The validation of the **XML file value** is done using the **Java Mapping** program that uses an Application Programming Interface (API).

An API is a set of declarations of the functions (or procedures) to support requests made by external computer programs. An API is used in this scenario with in XI by importing it in to Imported Archive.

Scenario Case

We have a File to IDoc scenario, where the input file has the field date and time. The date and time has to be validated and the file should be rejected without being transformed into an IDoc if the input file has an invalid Date and Time. Instead a rejection mail is triggered with the error message.



Scenario Using User Defined Function

Steps:

- Create a new function in the Message Mapping.
- Pass the Date field of the input file to the User Defined Function (UDF).
- Split the Date in to Year, Month and Date inside the UDF.
- Construct three IF condition statements inside the UDF checking the value of each Year, Month and Date to be within the specified range.

Yes! This will work. However a simple piece of coding will not validate the Leap year day of February and moreover some months have 30 days and others have 31 days. Where this simple coding will fail to validate. To incorporate all these the if-else condition grows inside the UDF.

Scenario Using API

We can reduce the burden of a complex coding inside the UDF to validate the date and time when the reusable coding are already available as an API.

The solution is to use an Application Programming Interface (API) that is very rich in validating not only Date but also many other input formats. We can use the API Commons Validator. You can get a brief idea about this API routin from this Help Documentation.

Steps:

- Read the XML file.
- Do an Interface Mapping that uses the Java Mapping to validate between the File sender and the Integration Process (IP).
- Receive and Send the message from the IP.
- Using a conditional receiver step determine if the receiver is going to be a IDoc or a Mail.

Scenario Depiction:

1. How an external API can be used within XI using the features of Imported Archieve.
2. How to handle file rejection with sending an email notification.

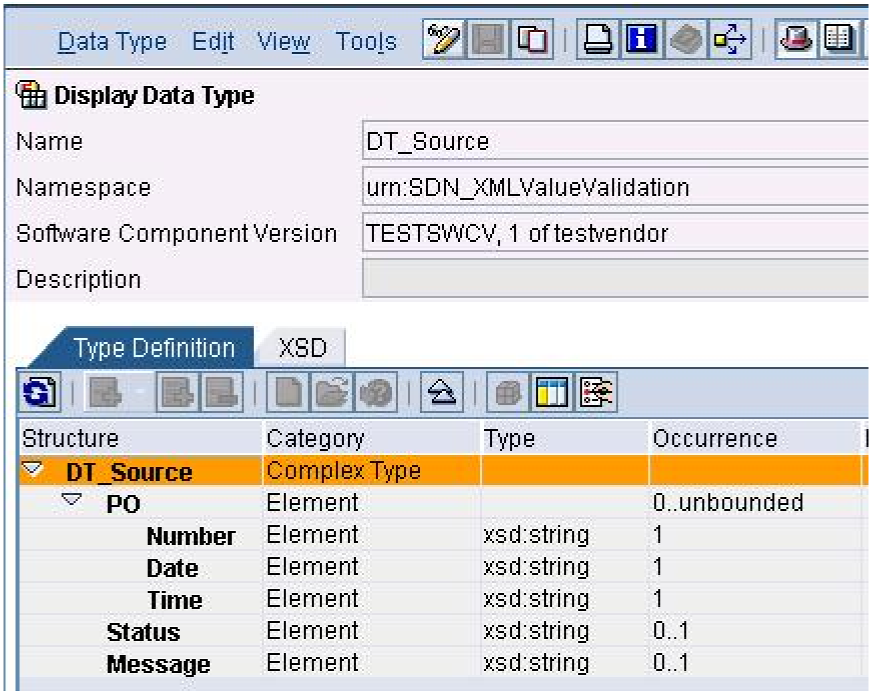
Important Points in Designing:

1. Design the source data type having a Status and Message node in addition to the required nodes. Set the occurrence of these nodes to 0….1.
2. Before Creating the Mapping Jar file place the java class file in the following folder path and archive

   \org\apache\commons\validator\routines\SDN_ValueValidation.class(& .java)

Design Objects in IR

1. Create one Source Data Type.



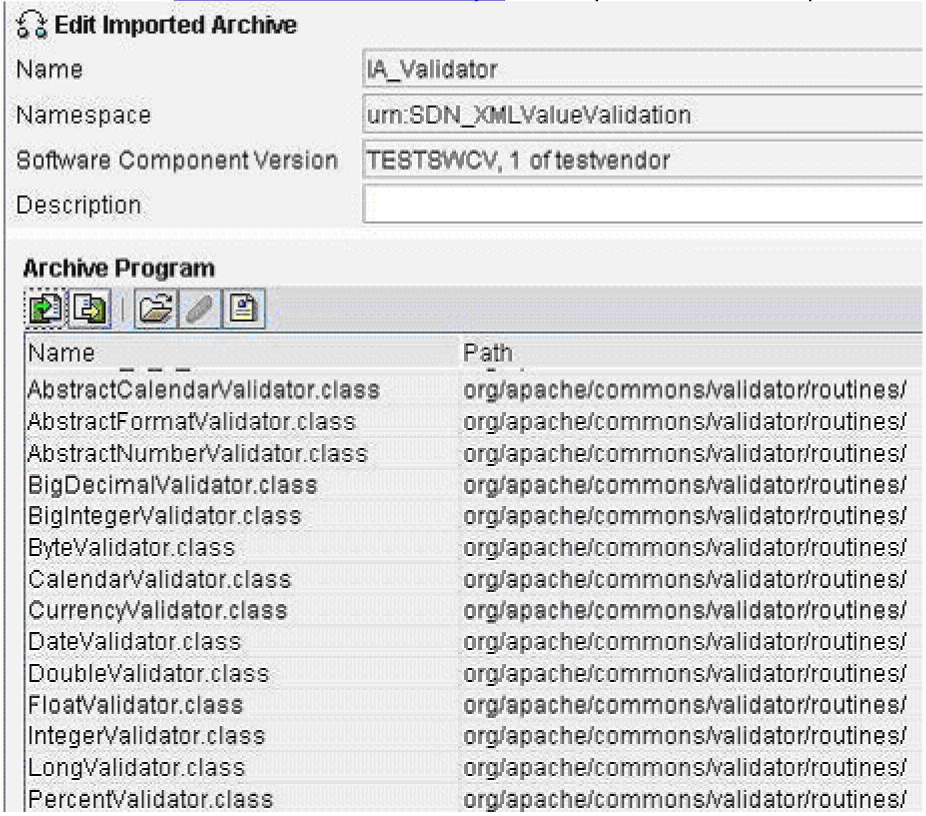
Display Data Type

Name	DT_Source
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

Structure	Category	Type	Occurrence
DT_Source	Complex Type		
PO	Element		0..unbounded
Number	Element	xsd:string	1
Date	Element	xsd:string	1
Time	Element	xsd:string	1
Status	Element	xsd:string	0..1
Message	Element	xsd:string	0..1

2. Create one Message Type and two Message Interface one as Outbound Asynchronous and the other as Abstract Asynchronous.

3. Download the commons-validator-1.3.1.jar and Import the JAR in Imported Archieve.



Edit Imported Archive

Name	IA_Validator
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

Archive Program

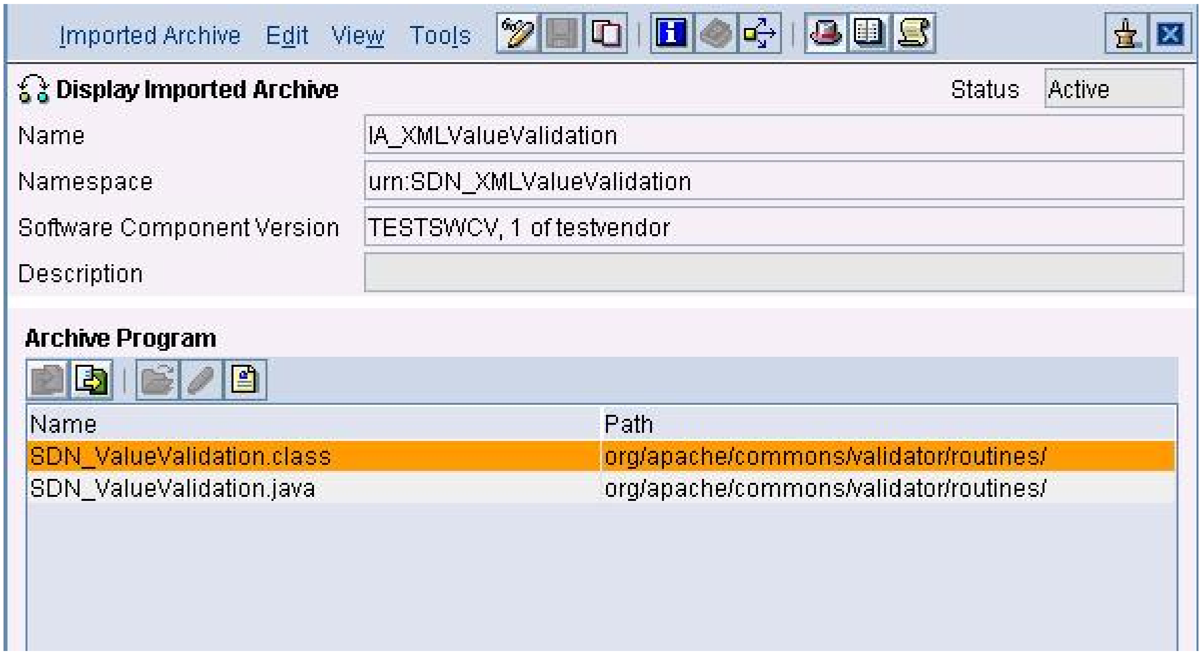
Name	Path
AbstractCalendarValidator.class	org/apache/commons/validator/routines/
AbstractFormatValidator.class	org/apache/commons/validator/routines/
AbstractNumberValidator.class	org/apache/commons/validator/routines/
BigDecimalValidator.class	org/apache/commons/validator/routines/
BigIntegerValidator.class	org/apache/commons/validator/routines/
ByteValidator.class	org/apache/commons/validator/routines/
CalendarValidator.class	org/apache/commons/validator/routines/
CurrencyValidator.class	org/apache/commons/validator/routines/
DateValidator.class	org/apache/commons/validator/routines/
DoubleValidator.class	org/apache/commons/validator/routines/
FloatValidator.class	org/apache/commons/validator/routines/
IntegerValidator.class	org/apache/commons/validator/routines/
LongValidator.class	org/apache/commons/validator/routines/
PercentValidator.class	org/apache/commons/validator/routines/

4. Compile the Java Mapping code and Import the JAR into Imported Archive.

```
package org.apache.commons.validator.routines;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Map;
import com.sap.aii.mapping.api.StreamTransformation;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Text;
public class SDN_ValueValidation implements StreamTransformation{
   private Element Status;
   private Element Message;
   private Text sts;
   private Text errs;
   private String errstatus = "0";
   private String msg = "";
   private DOMSource domS = null;

   public void setParameter(Map param)
   {
                 Map map = param;
   }
   public void execute(InputStream in, OutputStream out) throws
com.sap.aii.mapping.api.StreamTransformationException
   {
                try
                {
                       DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
                       DocumentBuilder db = dbf.newDocumentBuilder();
                       TransformerFactory tf = TransformerFactory.newInstance();
                       Transformer transform = tf.newTransformer();
                       Document doc = db.parse(in);
                       Document docout = db.newDocument();
                       DateValidator dv = new DateValidator();
                       TimeValidator tv = new TimeValidator(true,0);
                       NodeList PO = doc.getElementsByTagName("PO");
                       NodeList Number_lst = doc.getElementsByTagName("Number");
                       NodeList Date_lst = doc.getElementsByTagName("Date");
                       NodeList Time_lst = doc.getElementsByTagName("Time");
for(int i=0; i<PO.getLength(); i++)
{
String po = Number_lst.item(i).getChildNodes().item(0).getNodeValue();
String date = Date_lst.item(i).getChildNodes().item(0).getNodeValue();
String time = Time_lst.item(i).getChildNodes().item(0).getNodeValue();
                       if( dv.validate(date,"yyyy-MM-dd")== null)
                       {
                              errstatus = "1";
                              msg = msg + "Invalid Date for Purchase Order Number " +
po + "\n";
                       }
                       if( tv.validate(time,"HH:mm:ss") == null)
                       {
                              errstatus = "1";
                              msg = msg + "Invalid Time for Purchase Order Number " +
po + "\n";
                       }
}
                       Status = doc.createElement("Status");
                       Message = doc.createElement("Message");
                       sts = doc.createTextNode(errstatus);
                       errs = doc.createTextNode(msg);
```

```
                                        Status.appendChild(sts);
                                        Message.appendChild(errs);
                                        doc.getFirstChild().appendChild(Status);
                                        doc.getFirstChild().appendChild(Message);
                                        domS = new DOMSource(doc);
                                        transform.transform((domS),new StreamResult(out));
                                }catch (Exception e) {
                                System.out.print("Problem parsing the file: " + e.getMessage());
                                e.printStackTrace();
                                }
                    }
}
```



Display Imported Archive Status: Active

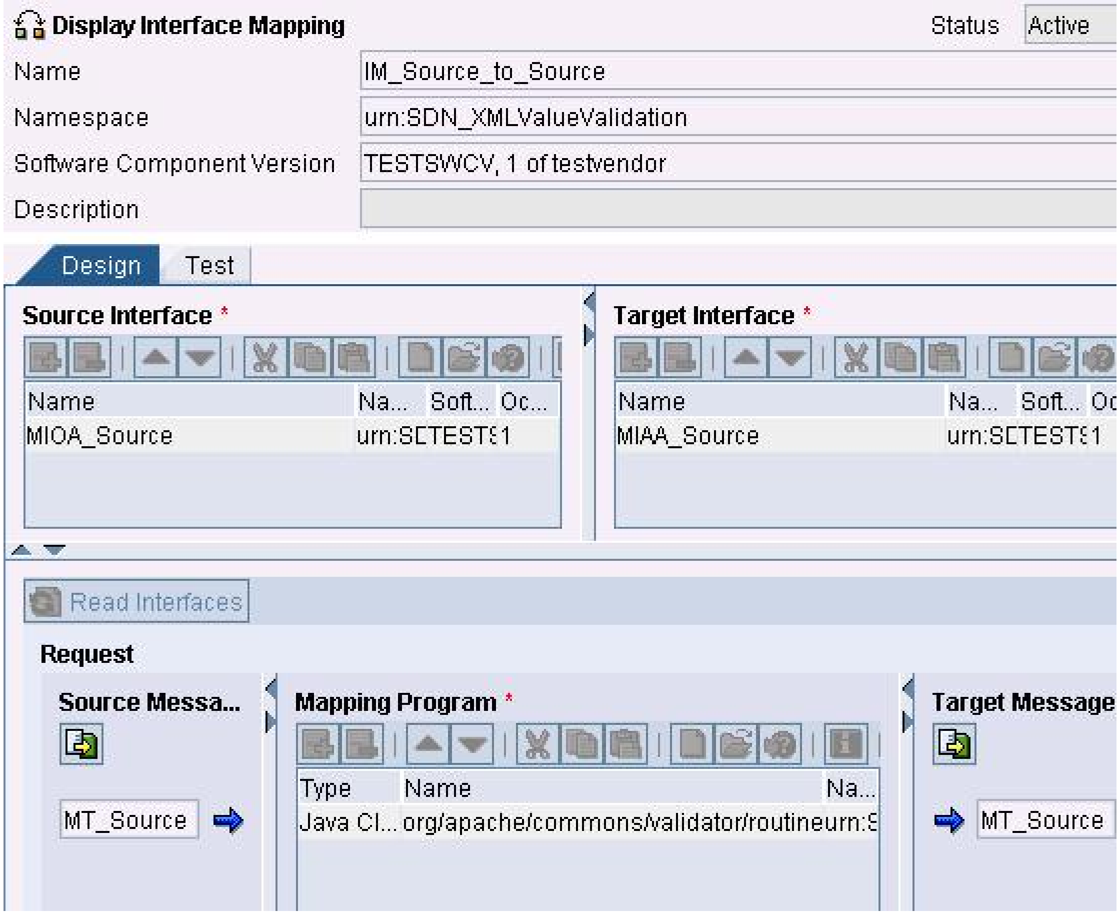
Name	IA_XMLValueValidation
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

Archive Program

Name	Path
SDN_ValueValidation.class	org/apache/commons/validator/routines/
SDN_ValueValidation.java	org/apache/commons/validator/routines/

5. Create an Interface Mapping using the following JAVA Class.

   \org\apache\commons\validator\routines\SDN_ValueValidation.class.



Display Interface Mapping Status Active

Name	IM_Source_to_Source
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

Source Interface *

Name	Na...	Soft...	Oc...
MIOA_Source	urn:SDTESTS1		

Target Interface *

Name	Na...	Soft...	Oc
MIAA_Source	urn:SDTESTS1		

Request

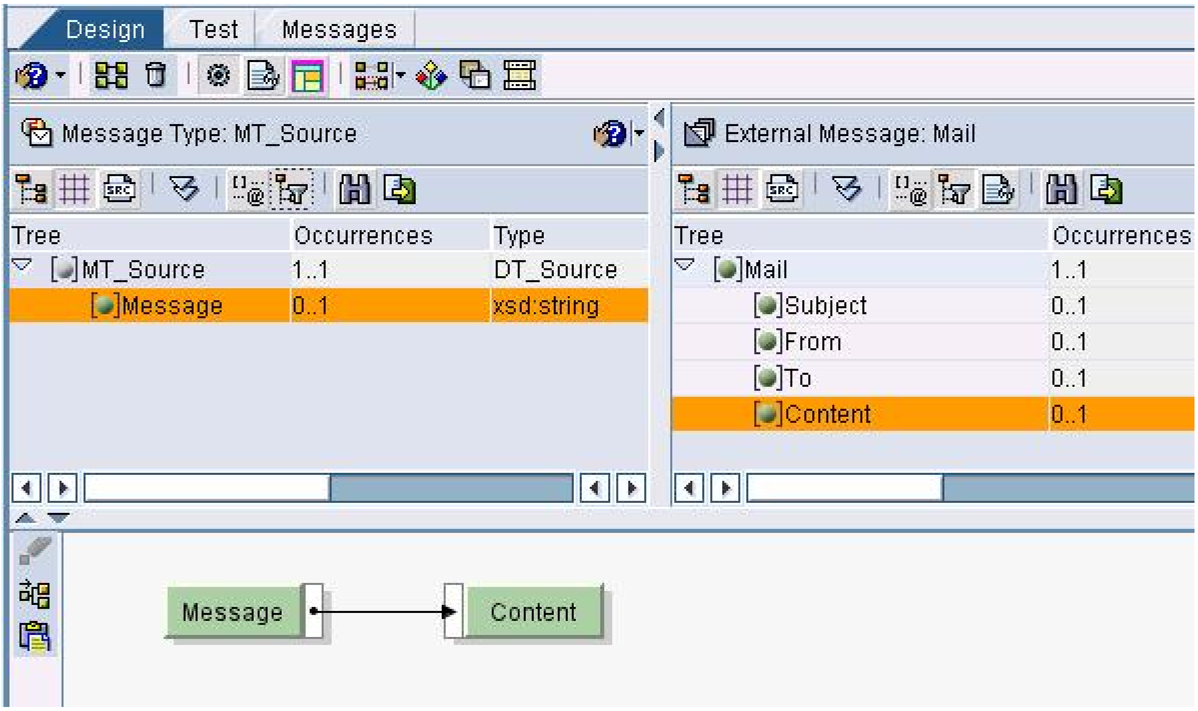
Source Messa... MT_Source

Mapping Program *

Type	Name	Na...
Java Cl...	org/apache/commons/validator/routineurn:S	

Target Message MT_Source

6. Create a Message Mapping and the Interface Mapping to trigger the Rejection Mail.

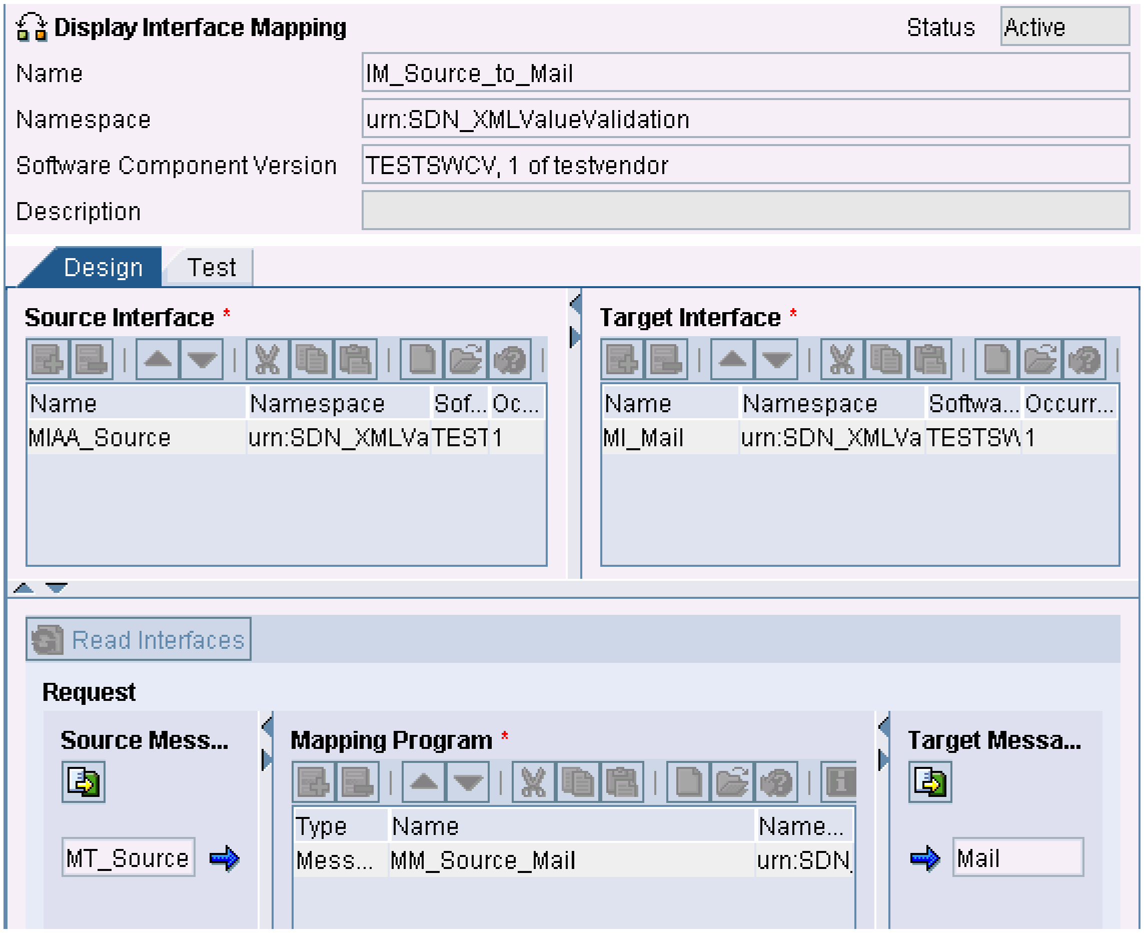


Message Type: MT_Source

Tree	Occurrences	Type
MT_Source	1..1	DT_Source
Message	0..1	xsd:string

External Message: Mail

Tree	Occurrences
Mail	1..1
Subject	0..1
From	0..1
To	0..1
Content	0..1


Display Interface Mapping

Status	Active
Name	IM_Source_to_Mail
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

Source Interface *

Name	Namespace	Sof...	Oc...
MIAA_Source	urn:SDN_XMLVa	TEST	1

Target Interface *

Name	Namespace	Softwa...	Occurr...
MI_Mail	urn:SDN_XMLVa	TESTSW	1

Request

Source Mess...

MT_Source

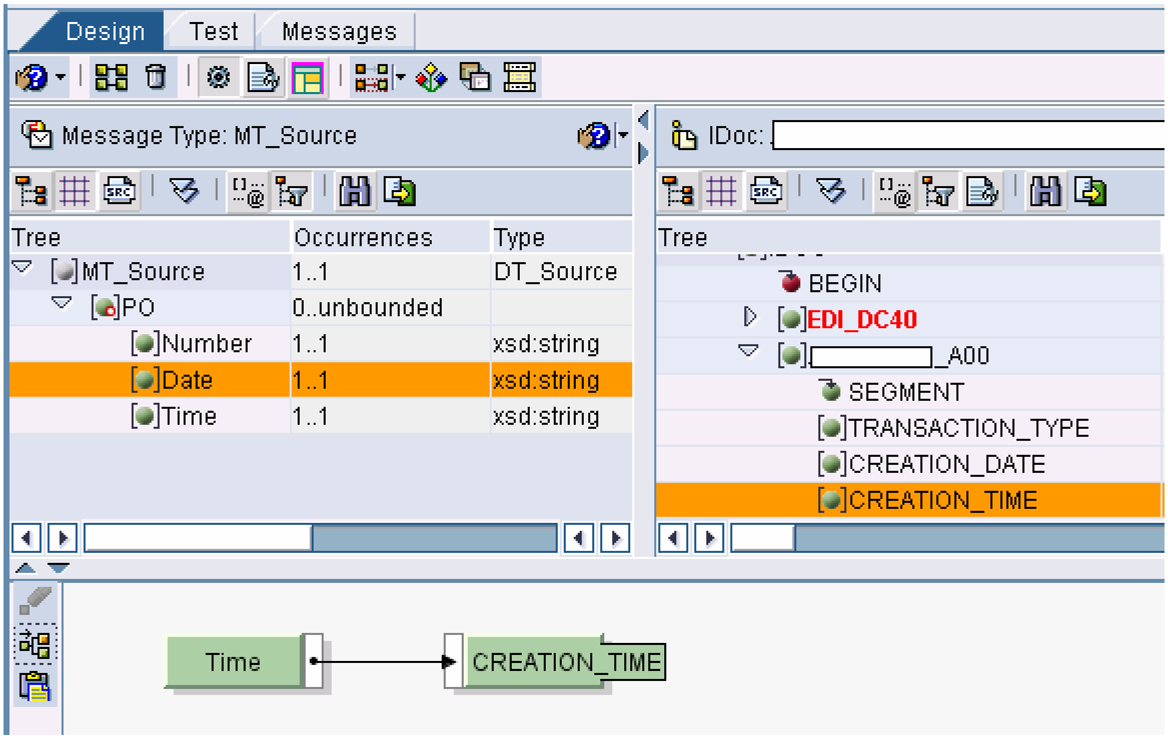
Mapping Program *

Type	Name	Name...
Mess...	MM_Source_Mail	urn:SDN

Target Messa...

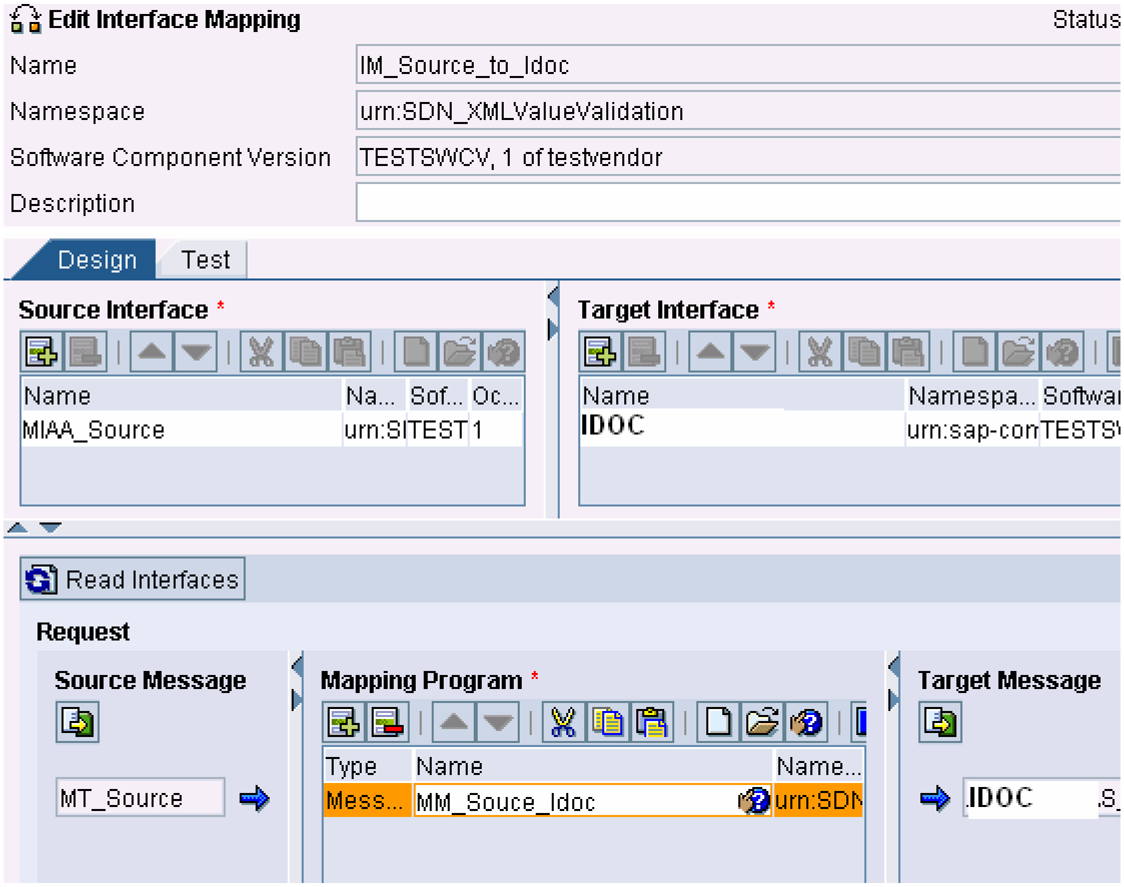
Mail

7. Create a Message Mapping and the Interface Mapping to transform File to Idoc.



Tree	Occurrences	Type
MT_Source	1..1	DT_Source
PO	0..unbounded	
Number	1..1	xsd:string
Date	1..1	xsd:string
Time	1..1	xsd:string

Tree
BEGIN
EDI_DC40
_A00
SEGMENT
TRANSACTION_TYPE
CREATION_DATE
CREATION_TIME



Edit Interface Mapping Status

Name	IM_Source_to_Idoc
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

Source Interface *

Name	Na...	Sof...	Oc...
MIAA_Source	urn:SITEST	1	

Target Interface *

Name	Namespa...	Softwa
IDOC	urn:sap-con	TESTS\

Request

Source Message

MT_Source

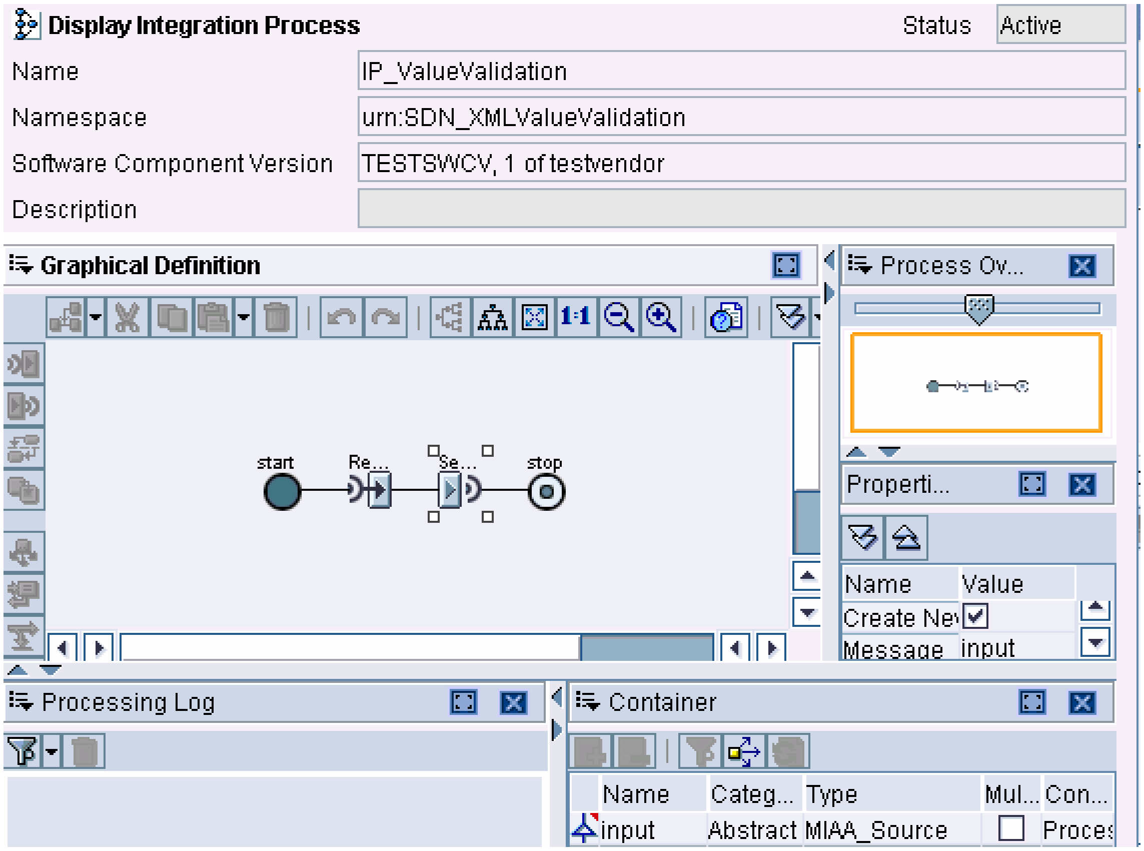
Mapping Program *

Type	Name	Name...
Mess...	MM_Souce_Idoc	urn:SDN

Target Message

IDOC S_

8. Design the Integration Process as following. Receive and Send the same source message.



Display Integration Process Status Active

Name	IP_ValueValidation
Namespace	urn:SDN_XMLValueValidation
Software Component Version	TESTSWCV, 1 of testvendor
Description	

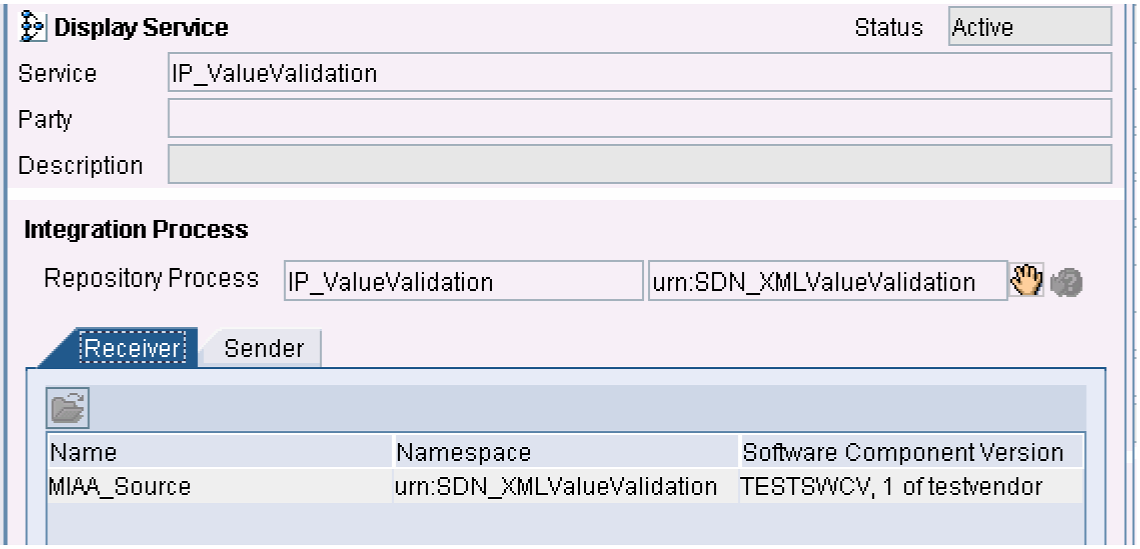
Name	Value
Create Ne	☑
Message	input

Container

Name	Categ...	Type	Mul...	Con...
input	Abstract	MIAA_Source	☐	Proces

Configuration Objects in ID:

1. Create new Integration Process.



Display Service

Status: Active

Service: IP_ValueValidation

Party:

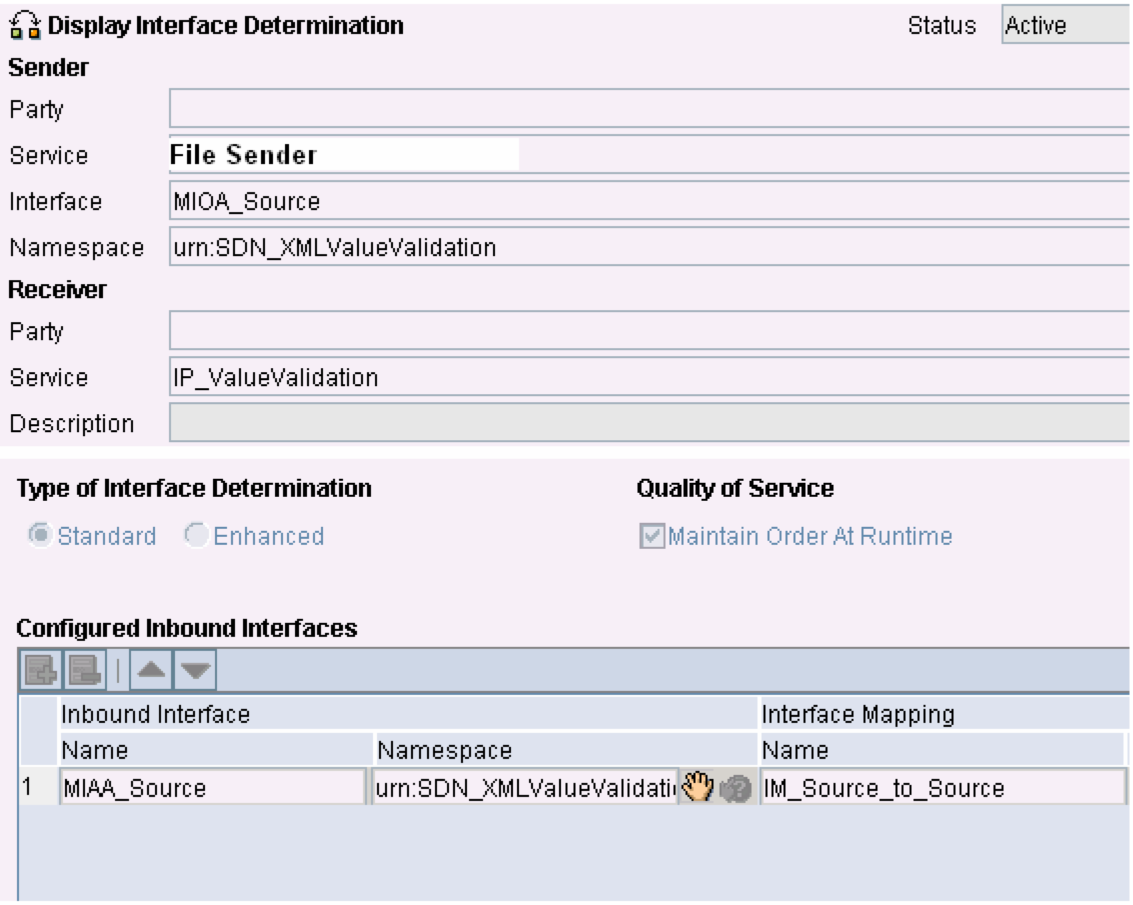
Description:

Integration Process

Repository Process: IP_ValueValidation | urn:SDN_XMLValueValidation

Name	Namespace	Software Component Version
MIAA_Source	urn:SDN_XMLValueValidation	TESTSWCV, 1 of testvendor

2. Configure the following Interface Determinations.



Display Interface Determination

Status: Active

Sender

Party:

Service: File Sender

Interface: MIOA_Source

Namespace: urn:SDN_XMLValueValidation

Receiver

Party:

Service: IP_ValueValidation

Description:

Type of Interface Determination

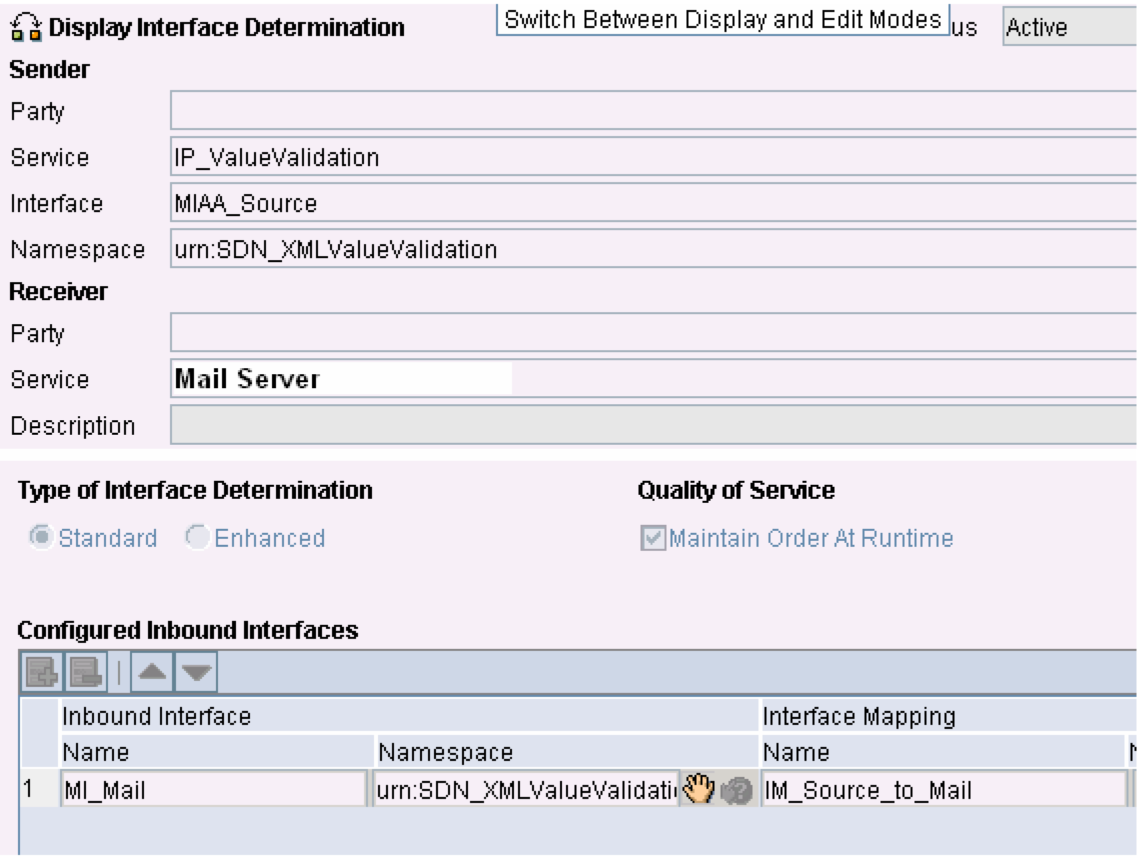
● Standard ○ Enhanced

Quality of Service

☑ Maintain Order At Runtime

Configured Inbound Interfaces

	Inbound Interface		Interface Mapping
	Name	Namespace	Name
1	MIAA_Source	urn:SDN_XMLValueValidati	IM_Source_to_Source



Display Interface Determination Switch Between Display and Edit Modes us Active

Sender

Party	
Service	IP_ValueValidation
Interface	MIAA_Source
Namespace	urn:SDN_XMLValueValidation

Receiver

Party	
Service	**Mail Server**
Description	

Type of Interface Determination

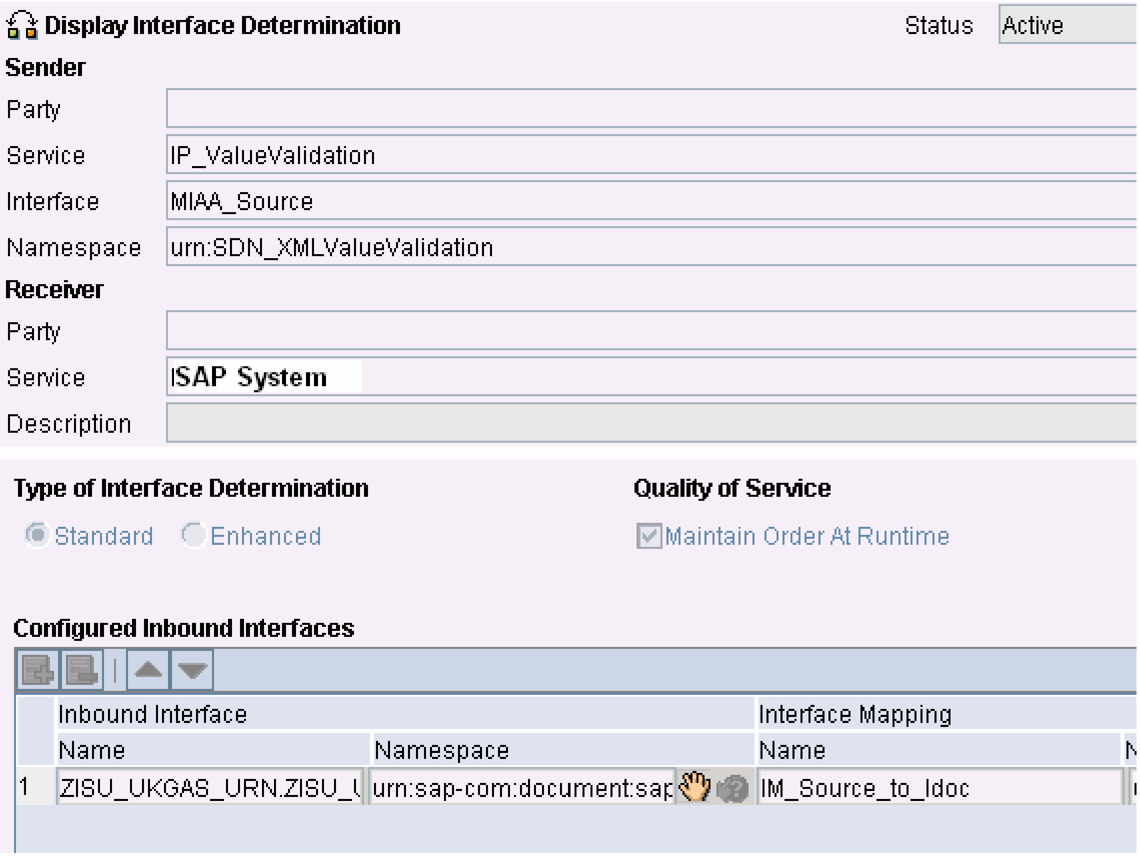
● Standard ○ Enhanced

Quality of Service

☑ Maintain Order At Runtime

Configured Inbound Interfaces

	Inbound Interface		Interface Mapping	
	Name	Namespace	Name	N
1	MI_Mail	urn:SDN_XMLValueValidati	IM_Source_to_Mail	



Display Interface Determination Status Active

Sender

Party	
Service	IP_ValueValidation
Interface	MIAA_Source
Namespace	urn:SDN_XMLValueValidation

Receiver

Party	
Service	**SAP System**
Description	

Type of Interface Determination

● Standard ○ Enhanced

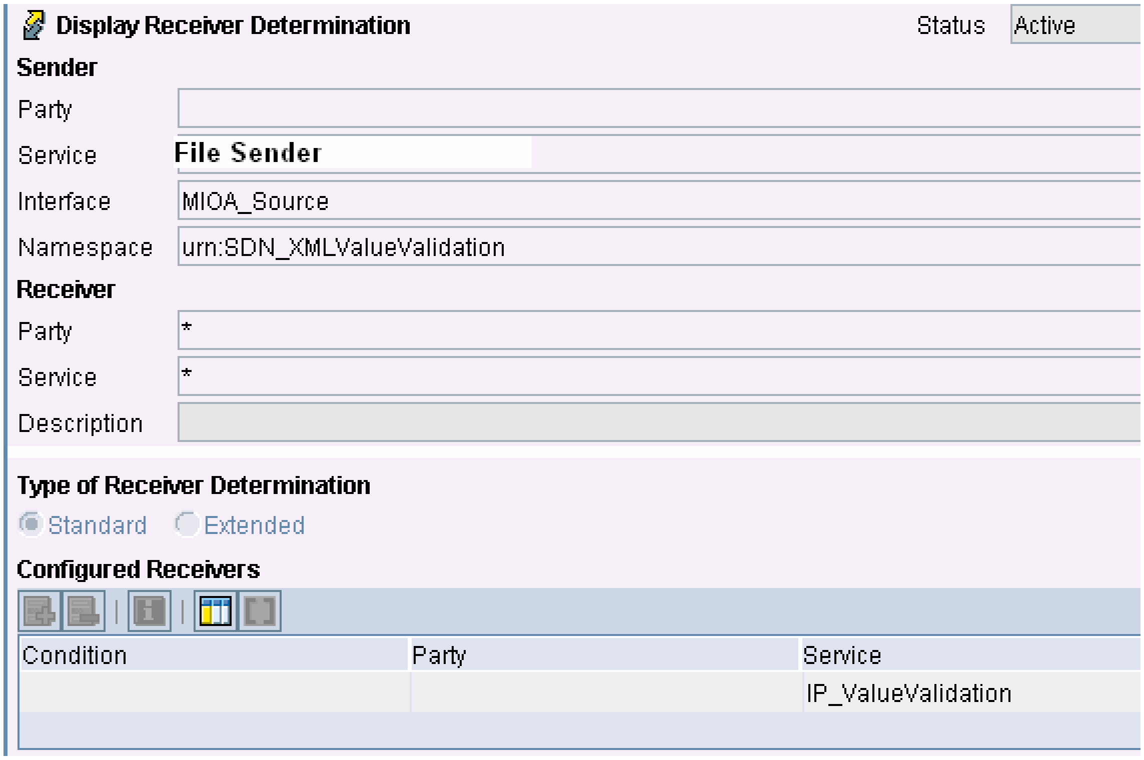
Quality of Service

☑ Maintain Order At Runtime

Configured Inbound Interfaces

	Inbound Interface		Interface Mapping	
	Name	Namespace	Name	N
1	ZISU_UKGAS_URN.ZISU_U	urn:sap-com:document:sap	IM_Source_to_Idoc	

3. Configure the following Receiver Determinations.



Display Receiver Determination Status: Active

Sender

Party	
Service	File Sender
Interface	MIOA_Source
Namespace	urn:SDN_XMLValueValidation

Receiver

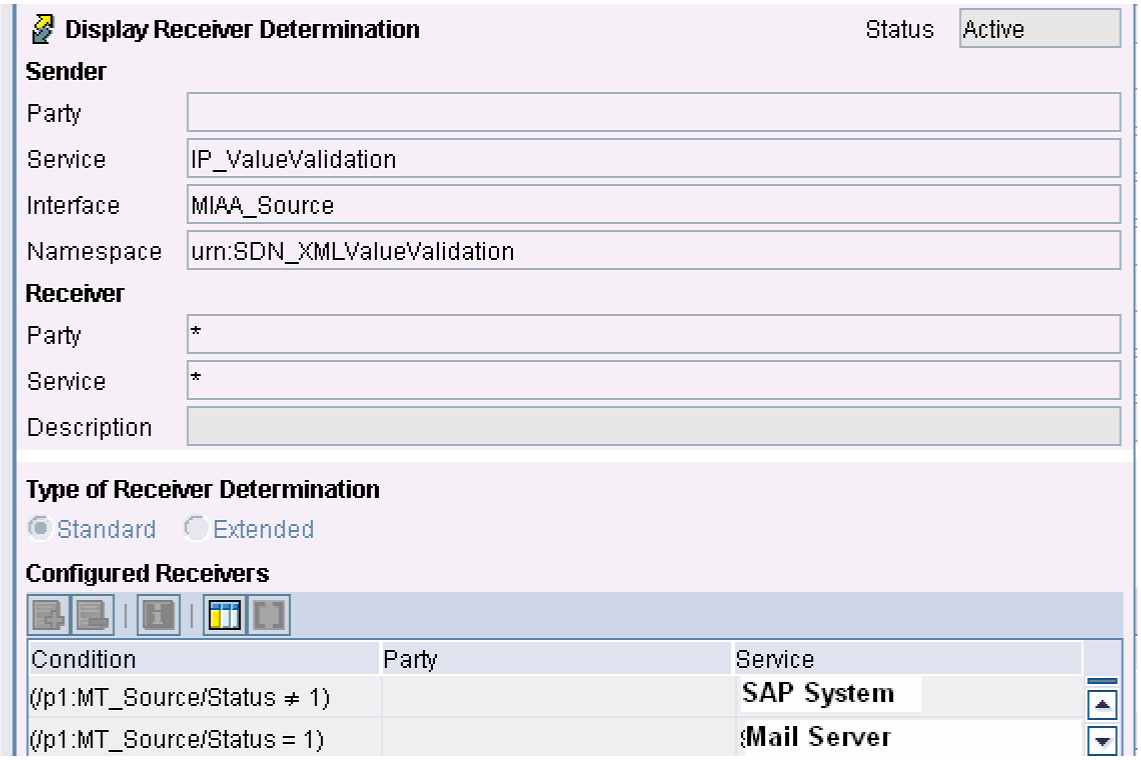
Party	*
Service	*
Description	

Type of Receiver Determination

● Standard ○ Extended

Configured Receivers

Condition	Party	Service
		IP_ValueValidation



Display Receiver Determination Status: Active

Sender

Party	
Service	IP_ValueValidation
Interface	MIAA_Source
Namespace	urn:SDN_XMLValueValidation

Receiver

Party	*
Service	*
Description	

Type of Receiver Determination

● Standard ○ Extended

Configured Receivers

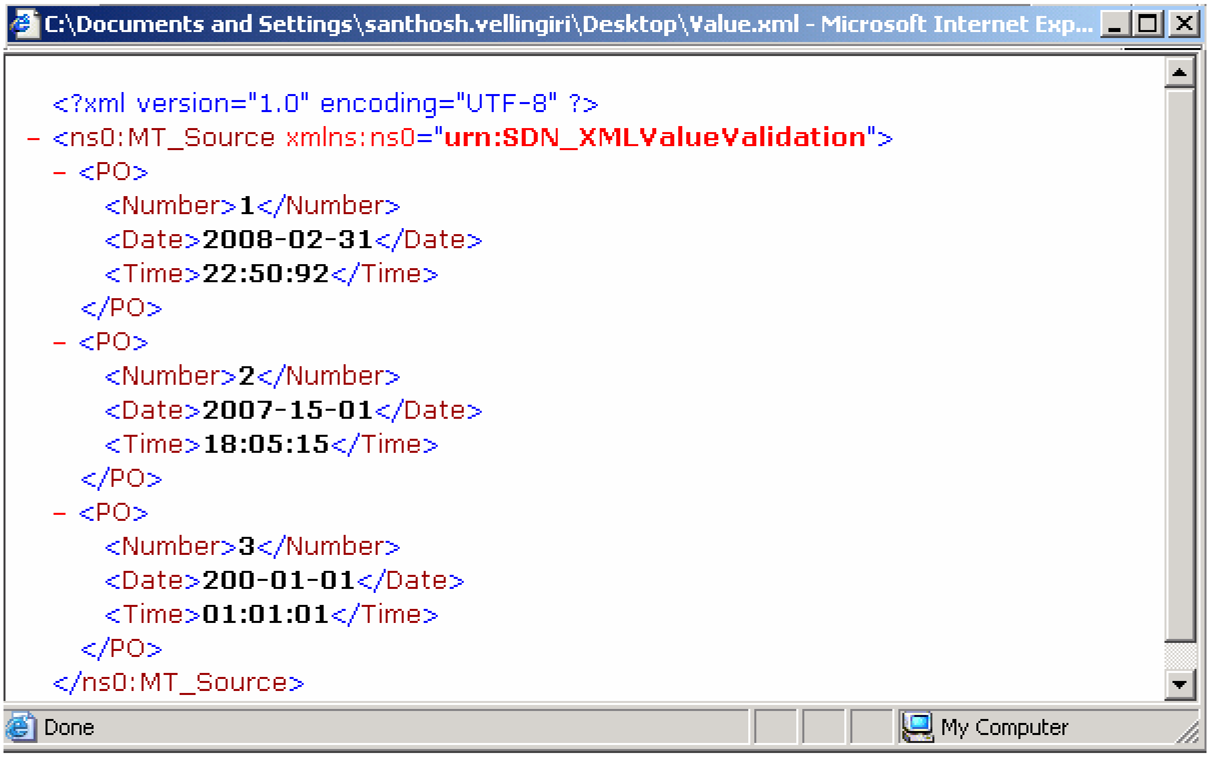
Condition	Party	Service
(/p1:MT_Source/Status ≠ 1)		SAP System
(/p1:MT_Source/Status = 1)		Mail Server

4. Create a Sender File Communication Channel, Receiver IDOC Communication Channel and a Mail Receiver Communication Channel.

5. Create a Sender Agreement and two Receiver Agreement.

Scenario Execution

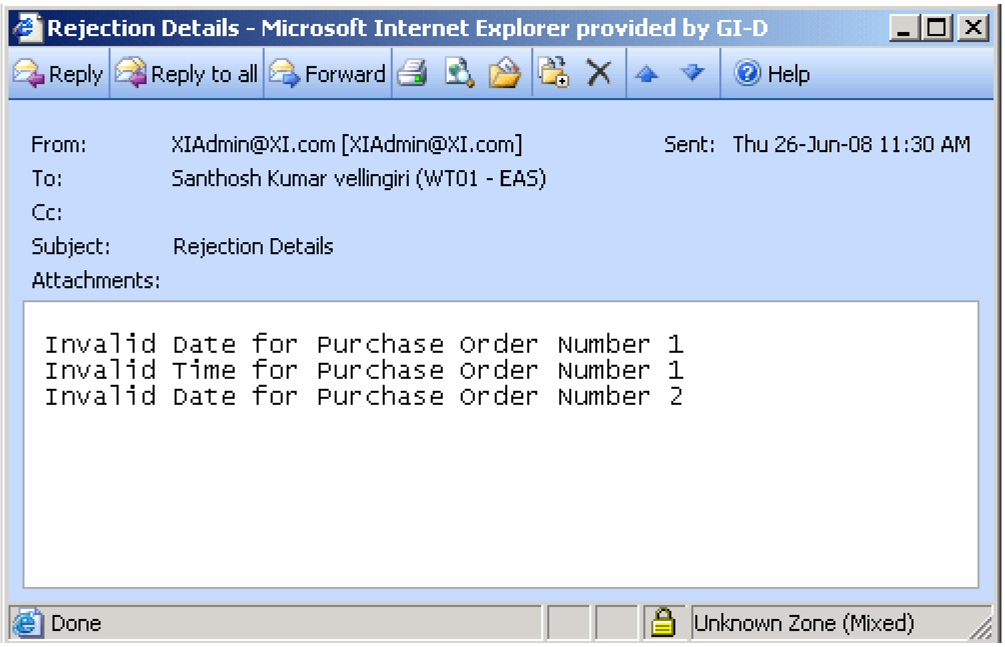
Post a File that is having an Invalid data as shown below.



C:\Documents and Settings\santhosh.vellingiri\Desktop\Value.xml - Microsoft Internet Exp...
Done
My Computer

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <ns0:MT_Source xmlns:ns0="urn:SDN_XMLValueValidation">
  - <PO>
      <Number>1</Number>
      <Date>2008-02-31</Date>
      <Time>22:50:92</Time>
    </PO>
  - <PO>
      <Number>2</Number>
      <Date>2007-15-01</Date>
      <Time>18:05:15</Time>
    </PO>
  - <PO>
      <Number>3</Number>
      <Date>200-01-01</Date>
      <Time>01:01:01</Time>
    </PO>
  </ns0:MT_Source>
```

A Rejection mails as below will knock your Mail Inbox.



Rejection Details - Microsoft Internet Explorer provided by GI-D
Reply
Reply to all
Forward
Help
Done
Unknown Zone (Mixed)

From: XIAdmin@XI.com [XIAdmin@XI.com]   Sent: Thu 26-Jun-08 11:30 AM
To: Santhosh Kumar vellingiri (WT01 - EAS)
Cc:
Subject: Rejection Details
Attachments:

```
Invalid Date for Purchase Order Number 1
Invalid Time for Purchase Order Number 1
Invalid Date for Purchase Order Number 2
```

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.