SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & 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.

A Beginner's Guide to SAP XI Settings, Part I

SAP DEVELOPER NETWORK

Applies To:

SAP Exhange Infrastructure

This beginner's document will walk you though various settings, configurations, and mappings.

This document is intended for those who want to understand the "what and how-to" of things in XI. It is also intended for some of the Java experts who want to read and check through some of the SAP R/3 transactions involving establishing connectivity, partner and port profiles and declarations, IDoc postings and IDoc generations, and various other set-up and configuration activities related to XI.

Therefore all intermediate and advanced users beware. You may want to ignore this particular document and visit a future attempt with scenarios on advanced topics.

One final piece of advice to the readers: This document includes a number of tips to help readers discern critical points and to enable the reader to have a compendium of quick reference materials. These might be useful for those interested in preparing themselves for certification or attending some interviews.

So we begin….

Summary

Our business scenario describes a real-time business environment. In our scenario we will imagine using SAP R/3 for ERP applications and business needs that will then be used for various integrations. Examples are using Hyperian systems for corporate financial reporting, LIMS for auality assurance and EH&S applications, VERTEX as the tax software, mainframes, Solumina, I2, ICH, CRM, SCM, SRM, BIW, SEM, etc., for various business needs. Apart from this, we use EDI for business partners' communications involving order information, delivery information, shipping and billing information, and invoices. This scenario evolves into a complex project needing huge amount of development work and maintenance.

It will grow more difficult to manage as we move forward.To cut this short, all the interfaces developed in SAP R/3 with various outbound/inbound environments can be replaced with XI in a most adaptable way within the permitted industry standard.

By: Srinivas Vankamamidi

Company: Intelligroup Inc.,

Date: 08 Jul 2005

Table of Contents

Scenario

Business scenario: Purchase order created in SAP R/3 sent as an IDoc into XI for FILE integration.

Introduction to SAP XI Components

XI has the following components and they will be used in designing, configuring, and monitoring an interface object.

1. SLD : Contains the Component Repository and Landscape Directory. Three of its important parts will be used extensively for any development. They are: SOFTWARE CATALOG (define the products, software components and versions, usage dependencies, if any), TECHNICAL SYSTEM (technical system of third-party type - for example purpose and its attributes), BUSINESS SYSTEM (to attach to the scenario that invokes all the definitions of a technical landscape).

2. Integration Repository: Desgin and develop the interface - for example, mappings, interfaces, and components. This is built on Java and follows the J2EE standards. These components will be deployed in configuration for generating a scenario.

3. Integration Directory: Develop the business scenario to execute the interface between two systems (sending and receiving parties).

4. Integration Server: It is the central and core part of XI used for processing the messages. It receives the message from Sender and applies the **L**ogical routing, **T**echnical Routing, **M**apping and sends to the Receiver through communication channels developed in the integration directory. It has the following parts in it, used for specific purposes.

   4.1. Integration Engine

   4.2. BPE

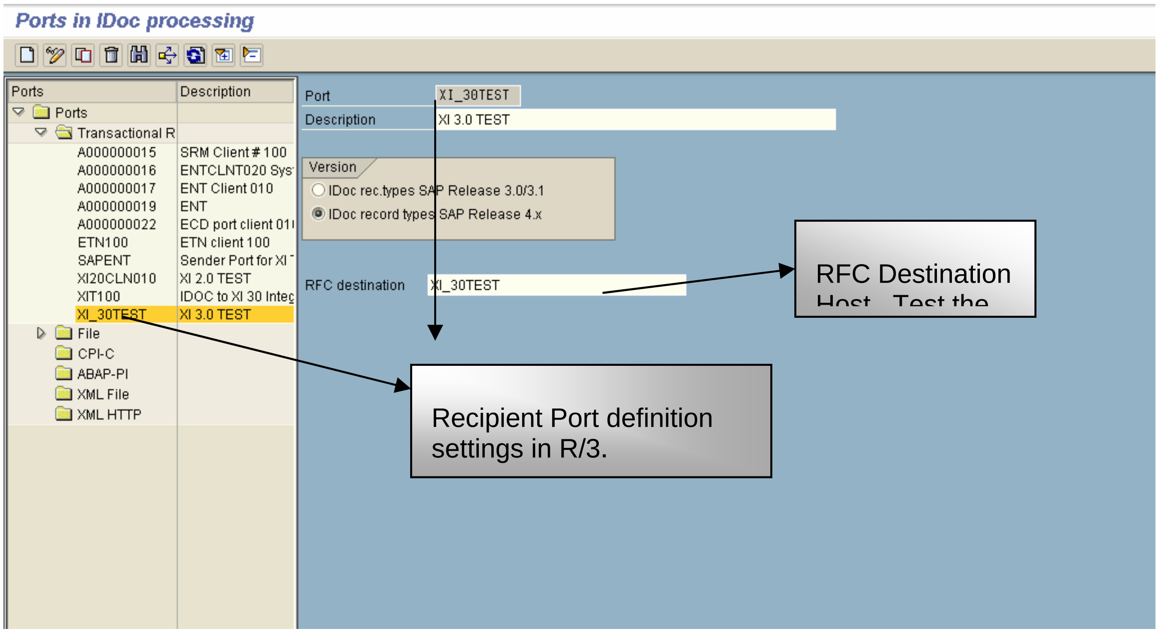
   4.3. Adapter Engine

Each of these plays a vital role in any development and they are required to be maintained (for now please understand it this way) for every object we design, develop, and deploy.

SAP R/3 Settings for Processing this Scenario

To begin with, let's understand some basic steps that will take us through some of the crucial transactions in SAP R/3 that are prerequisites to perform interfacing with XI.

Communication Ports Use the transaction WE21 to maintain a port and assign a proper RFC destination.



Ports in IDoc processing

Ports	Description
Ports	
Transactional R	
A000000015	SRM Client # 100
A000000016	ENTCLNT020 Sys
A000000017	ENT Client 010
A000000019	ENT
A000000022	ECD port client 01
ETN100	ETN client 100
SAPENT	Sender Port for XI
XI20CLN010	XI 2.0 TEST
XIT100	IDOC to XI 30 Integ
XI_30TEST	XI 3.0 TEST
File	
CPI-C	
ABAP-PI	
XML File	
XML HTTP	

Port XI_30TEST

Description XI 3.0 TEST

Version

○ IDoc rec.types SAP Release 3.0/3.1

● IDoc record types SAP Release 4.x

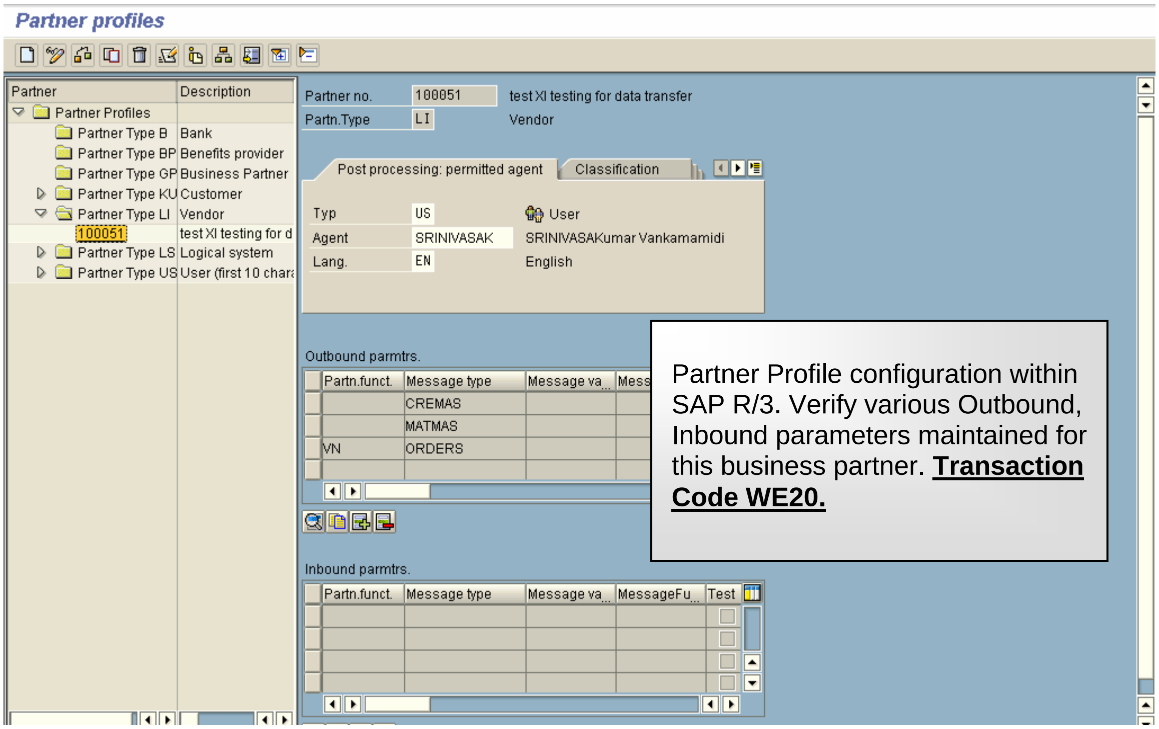
RFC destination XI_30TEST

RFC Destination
Host   Test the

Recipient Port definition settings in R/3.

Partner Profiles

Use the transaction WE20 to maintain the partners. These partners are identified by their TYPE, FUNCTION. For instance, partner type LS represents LOGICAL SYSTEM and type LI represents SUPPLIERS/VENDORS; type KU represents CUSTOMERS/DEBTORS, etc. For the purpose of our scenario we need to create a partner with type LI. Before doing this we need to create a VENDOR in the Vendor Master within R/3 using transaction XK01 (central) or MK01 (purchasing). To create this you will need the information such as company code, purchasing organization, recon account, currency, payment terms (not a required field), etc. Once this is done, you will then create a logical partner and assign it with certain outbound and/or inbound parameters as needed.



Partner profiles

Partner	Description
Partner Profiles	
Partner Type B	Bank
Partner Type BP	Benefits provider
Partner Type GP	Business Partner
Partner Type KU	Customer
Partner Type LI	Vendor
100051	test XI testing for d
Partner Type LS	Logical system
Partner Type US	User (first 10 chara

Partner no. 100051 test XI testing for data transfer
Partn.Type LI Vendor

Post processing: permitted agent | Classification

Typ US User
Agent SRINIVASAK SRINIVASAKumar Vankamamidi
Lang. EN English

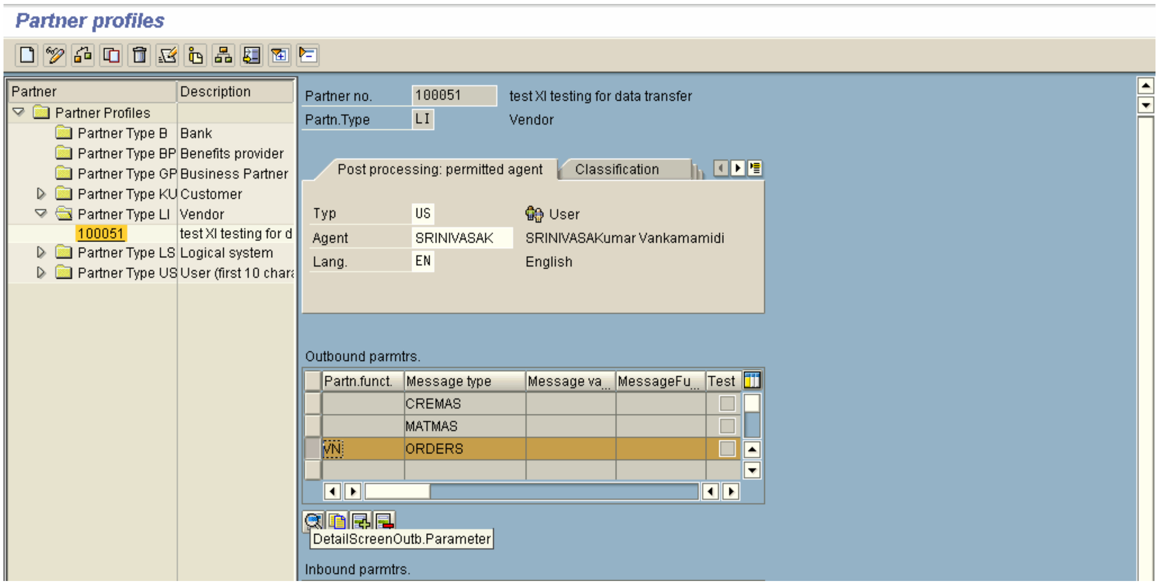
Outbound parmtrs.

Partn.funct.	Message type	Message va..	Mess
	CREMAS		
	MATMAS		
VN	ORDERS		

Inbound parmtrs.

Partn.funct.	Message type	Message va..	MessageFu..	Test
				□
				□
				□
				□

Partner Profile configuration within SAP R/3. Verify various Outbound, Inbound parameters maintained for this business partner. **Transaction Code WE20.**



Partner profiles

Partner	Description
Partner Profiles	
Partner Type B	Bank
Partner Type BP	Benefits provider
Partner Type GP	Business Partner
Partner Type KU	Customer
Partner Type LI	Vendor
100051	test XI testing for d
Partner Type LS	Logical system
Partner Type US	User (first 10 chara

Partner no. 100051 test XI testing for data transfer
Partn.Type LI Vendor

Post processing: permitted agent | Classification

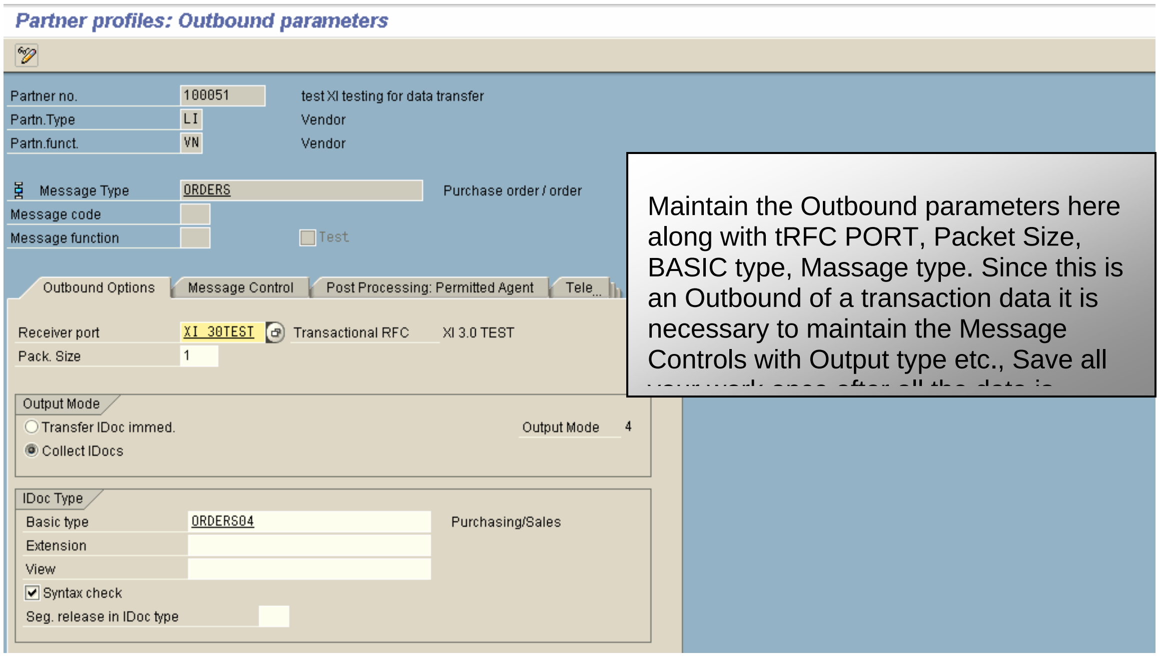
Typ US User
Agent SRINIVASAK SRINIVASAKumar Vankamamidi
Lang. EN English

Outbound parmtrs.

Partn.funct.	Message type	Message va..	MessageFu..	Test
	CREMAS			□
	MATMAS			□
VN	ORDERS			□

DetailScreenOutb.Parameter

Inbound parmtrs.



Partner profiles: Outbound parameters

Partner no.	100051	test XI testing for data transfer
Partn.Type	LI	Vendor
Partn.funct.	VN	Vendor
Message Type	ORDERS	Purchase order / order
Message code		
Message function		□ Test

Receiver port: XI_30TEST Transactional RFC XI 3.0 TEST

Pack. Size: 1

Output Mode

○ Transfer IDoc immed.

● Collect IDocs

Output Mode 4

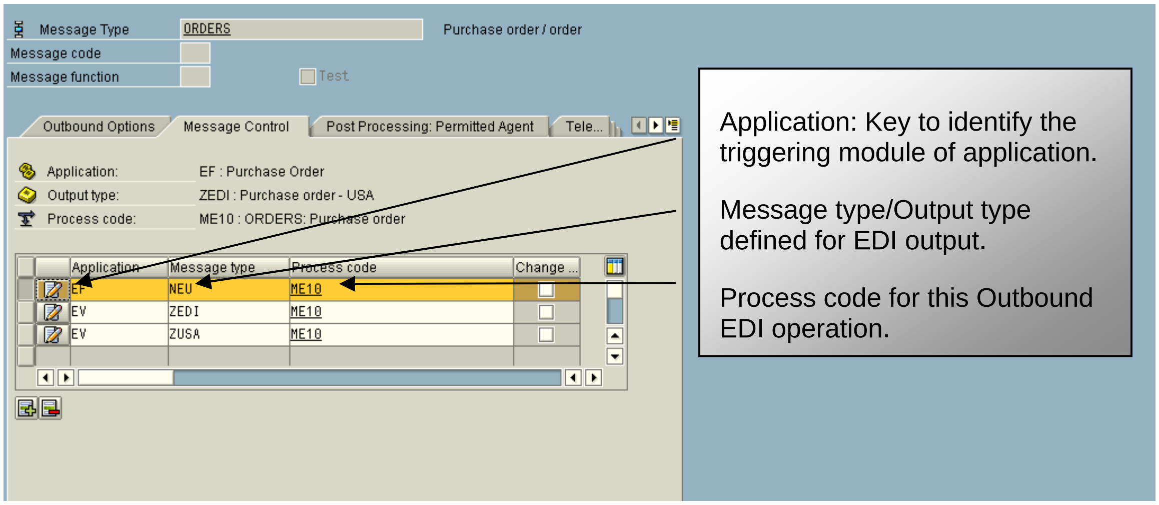
IDoc Type

Basic type	ORDERS04	Purchasing/Sales
Extension		
View		

☑ Syntax check

Seg. release in IDoc type

Maintain the Outbound parameters here along with tRFC PORT, Packet Size, BASIC type, Massage type. Since this is an Outbound of a transaction data it is necessary to maintain the Message Controls with Output type etc., Save all your work once after all the data is



Message Type	ORDERS	Purchase order / order
Message code		
Message function		□ Test

Application: EF : Purchase Order

Output type: ZEDI : Purchase order - USA

Process code: ME10 : ORDERS: Purchase order

Application	Message type	Process code	Change ...
EF	NEU	ME10	□
EV	ZEDI	ME10	□
EV	ZUSA	ME10	□

Application: Key to identify the triggering module of application.

Message type/Output type defined for EDI output.

Process code for this Outbound EDI operation.

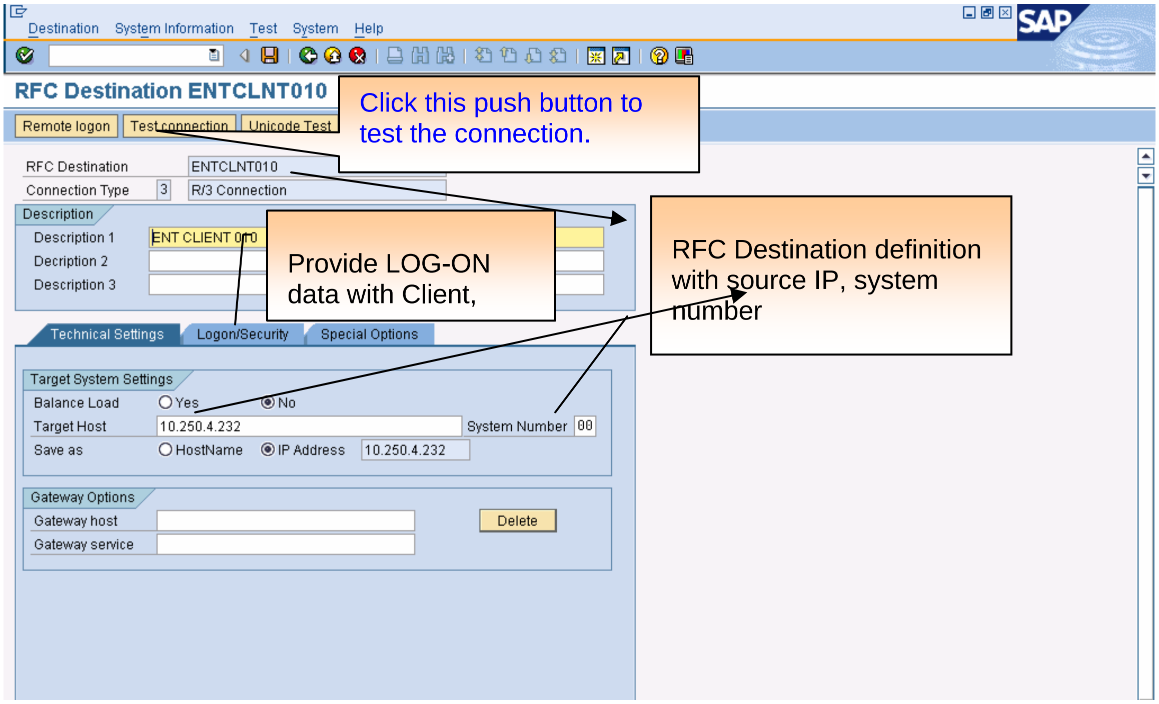
We are almost done with R/3 settings. Before going to XI, do the following:

Test the RFC destination using SM59. Write down the information like material number, plant, storage location, item category, etc. needed for creating a purchase order in R/3.

Open the IDoc type ORDERS04 using transaction WE30 and check all the SEGEMENTS assigned in it. Double-click on each of the segments and note what fields you wish to capture into the file or as per business needs. This will be easy if it is known at this stag; otherwise, it will be a bit cumbersome while in the designing stage in XI.

SAP XI Settings

Create an RFC destination and test the connection. Use transaction SM59 for doing this. Make sure you use the correct IP address of the host and system number along with the proper User ID and password.



RFC Destination ENTCLNT010

Click this push button to test the connection.

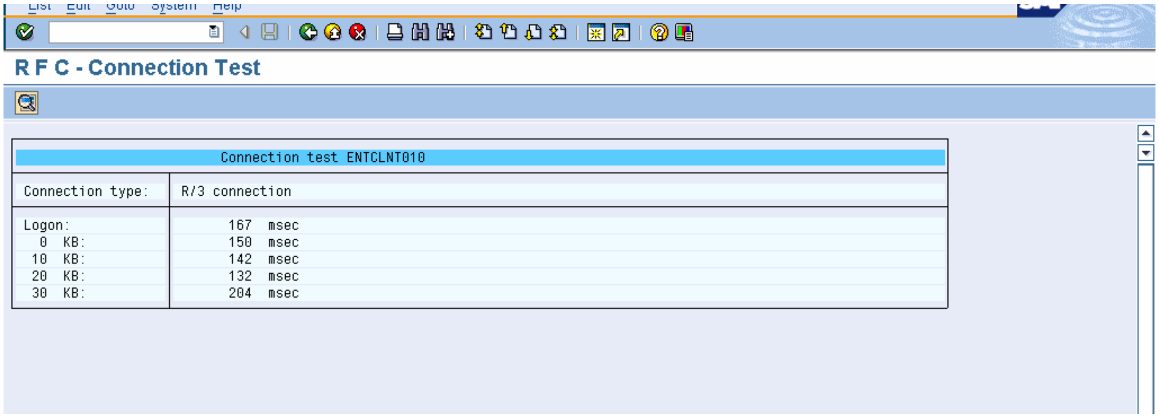
Provide LOG-ON data with Client,

RFC Destination definition with source IP, system number

RFC Destination ENTCLNT010
Connection Type 3 R/3 Connection
Description 1 ENT CLIENT 010
Decription 2
Description 3

Balance Load ○ Yes ● No
Target Host 10.250.4.232 System Number 00
Save as ○ HostName ● IP Address 10.250.4.232

Gateway host
Gateway service

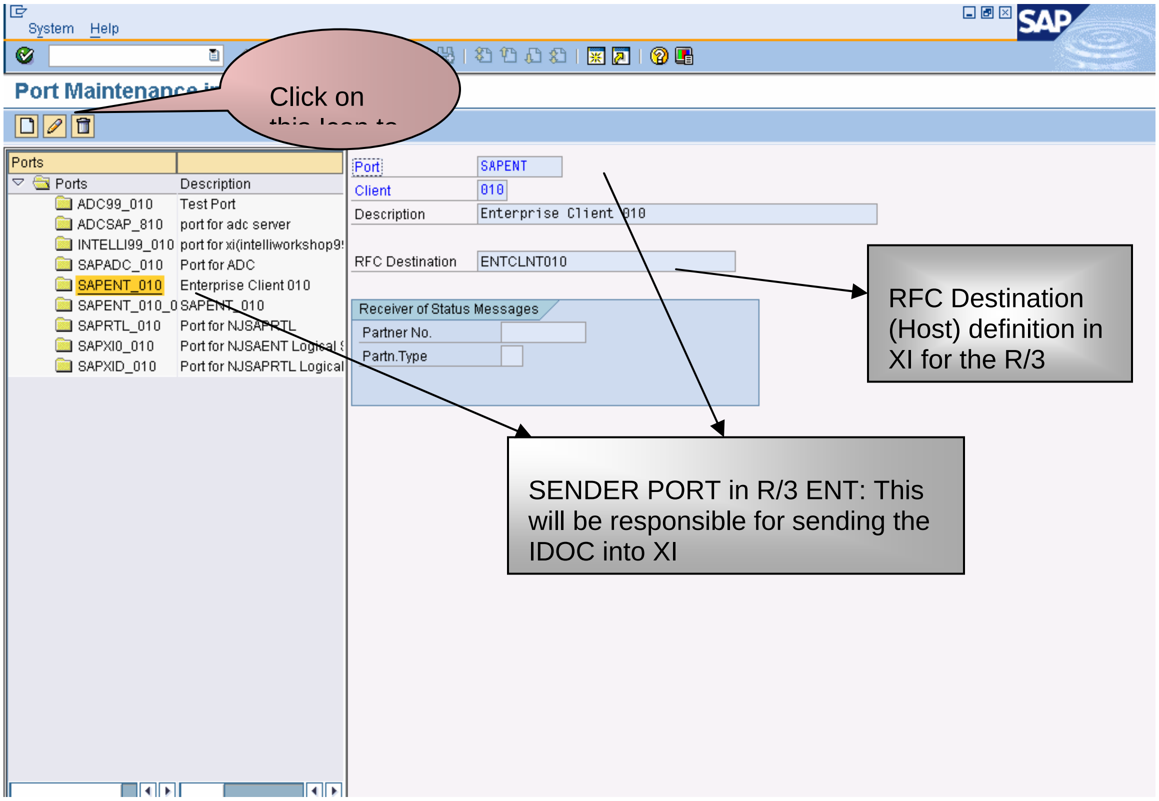


R F C - Connection Test

```
                    Connection test ENTCLNT010

Connection type:     R/3 connection

Logon:                     167  msec
   0  KB:                  150  msec
  10  KB:                  142  msec
  20  KB:                  132  msec
  30  KB:                  204  msec
```

Create a PORT for Communicating and Sender Identification

If we don’t maintain it, and it is not in sync with data maintained in R/3, it will cause issues and we will not have a reason why the IDoc is not getting transmitted. We might see issues like “No Recipient Exists”/”No Sender Identified” etc., and will be very annoying. To do this in XI, use transaction IDX1. For all those who are familiar with SAP R/3, this looks like WE21.


System
Help
Port Maintenance i
Click on
this Icon to
RFC Destination (Host) definition in XI for the R/3
SENDER PORT in R/3 ENT: This will be responsible for sending the IDOC into XI

Ports	Description
ADC99_010	Test Port
ADCSAP_810	port for adc server
INTELLI99_010	port for xi(intelliworkshop9!
SAPADC_010	Port for ADC
SAPENT_010	Enterprise Client 010
SAPENT_010_0	SAPENT_010
SAPRTL_010	Port for NJSAPRTL
SAPXI0_010	Port for NJSAENT Logical !
SAPXID_010	Port for NJSAPRTL Logical

Port SAPENT
Client 010
Description Enterprise Client 010
RFC Destination ENTCLNT010

Receiver of Status Messages
Partner No.
Partn.Type

There are a few more steps to follow, but we will do it at a later stage.

Now, let’s see how to design our scenario.

Below are some tips before we begin.

Tips

Tip 1

You need MESSAGE CONTROL maintained while data processing between two systems (mostly while sending the transaction data like PO, SO, delivery, billing, shipping, PGI, etc.) that are controlled by using an OUTPUT TYPE for data transmission (EDI/PRINT/FAX etc., either outbound or inbound). We will see this in detail at a later stage, but for now understand that we are using the ORDERS message type for sending the IDoc from R/3, which uses an output type, process code, etc. defined in the partner profile.

Tip 2

Your partner number should pre-exist as a logical system when the partner type is LS. If is not, create a logical system either from transaction SALE or BD54. If you have no authorizations to create the logical system (normally not) then check the entries in table TBDLS (using transaction SE16, enter this table) then display all entries (hit F8). See if there are any test systems for test purpose or any one that you think can be used for your purpose. Use this for IDoc communication. *In the current case, the partner type is LI, so let’s not worry about it but is needed to know about it.*

1. Start XI and type/access SXMB_IFR to launch your integration builder, or type "http://<server>:<port>/rep in your browser's address bar, then hit enter. This will launch Integration Builder.

2. Create the technical landscape involving SWCV, SWC, PRODUCT, business systems, and technical system. Save all your work and go back to Integration Builder.

Tip 3

Create the SWCV with usage dependency (installation time) and add the SAP BASIS component to it. This will preinstall a few components that you can use while in the integration repository.

Tip 4

You need only one business system for your RECEIVER to be created and subsequently one technical system. There are a couple of crucial things to be maintained here, like logical system, integration server, SWCV etc., which are essentials to know.

Tip 5

List everything for your convenience.

Tip 6

Since sender is SAP R/3, you don't need a sender configuration and sender agreement (done in Integration Directory). So, IDoc outflow from R/3 needs less set up. Of course, you should be doing this in a proper way or else you will be wondering, "what had had happened to my IDoc?"

3. Click on Integration Repository and import all your stuff from SLD (system landscape directory) from the ENVIRONMENT menu option. All the SAP folks know this is a kind of standard tool and menu bar as we see it in the normal SAP screen. When you do this, the system will prompt you with all the available SWCVs from SLD and have the chosen one imported into the Integration Repository.

4. So, we have our SWC and corresponding version available for us to begin with. Double-click on SWCV then bring it TO the edit mode. Now as specified in tip 2.1, enter the required info after switching to the "IMPORT RFC/IDOC............" radio button displayed in the work area. This is essential, as this will then be used for importing IDocs from the source system.

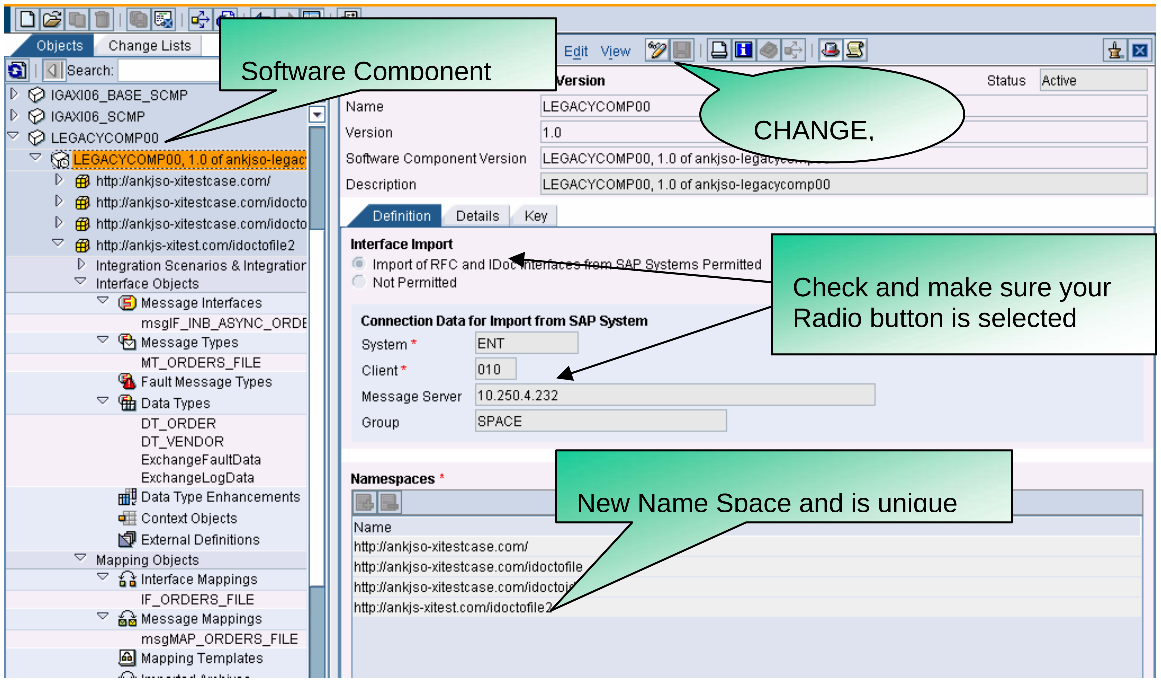
5. Click on the " + " icon at the bottom part of work area appearing just below the server info where it has been entered as stated in the step-4 above. This will give you the option to enter the NAME SPACE. It is unique identifier along with SWCV and Object Name it becomes a Key. (<--TIP 7). Enter something meaningful like "http://xitestcase.com/orderdata or urn:.......... the way you like it. But, give a meaningful name so that it will not be hard to type it if you need to.

6. Save the work, then activate it.

Tip 7

Release your change list once at this stage as it adds all the components required for designing the integration and the pre-acquired components from USAGE DEPENDENCIES (refer to TIP-3). This is one essential and you are partly done with your work at this stage.

7. Expand the Name Space by clicking on the triangle pointing rightwards.



IGAXI06_BASE_SCMP
IGAXI06_SCMP
LEGACYCOMP00
LEGACYCOMP00, 1.0 of ankjso-legac
http://ankjso-xitestcase.com/
http://ankjso-xitestcase.com/idocto
http://ankjso-xitestcase.com/idocto
http://ankjs-xitest.com/idoctofile2
Integration Scenarios & Integratior
Interface Objects
Message Interfaces
msgIF_INB_ASYNC_ORDE
Message Types
MT_ORDERS_FILE
Fault Message Types
Data Types
DT_ORDER
DT_VENDOR
ExchangeFaultData
ExchangeLogData
Data Type Enhancements
Context Objects
External Definitions
Mapping Objects
Interface Mappings
IF_ORDERS_FILE
Message Mappings
msgMAP_ORDERS_FILE
Mapping Templates

Version

Status: Active

Name: LEGACYCOMP00

Version: 1.0

Software Component Version: LEGACYCOMP00, 1.0 of ankjso-legacy

Description: LEGACYCOMP00, 1.0 of ankjso-legacycomp00

Interface Import

● Import of RFC and IDoc Interfaces from SAP Systems Permitted

○ Not Permitted

Connection Data for Import from SAP System

System *: ENT

Client *: 010

Message Server: 10.250.4.232

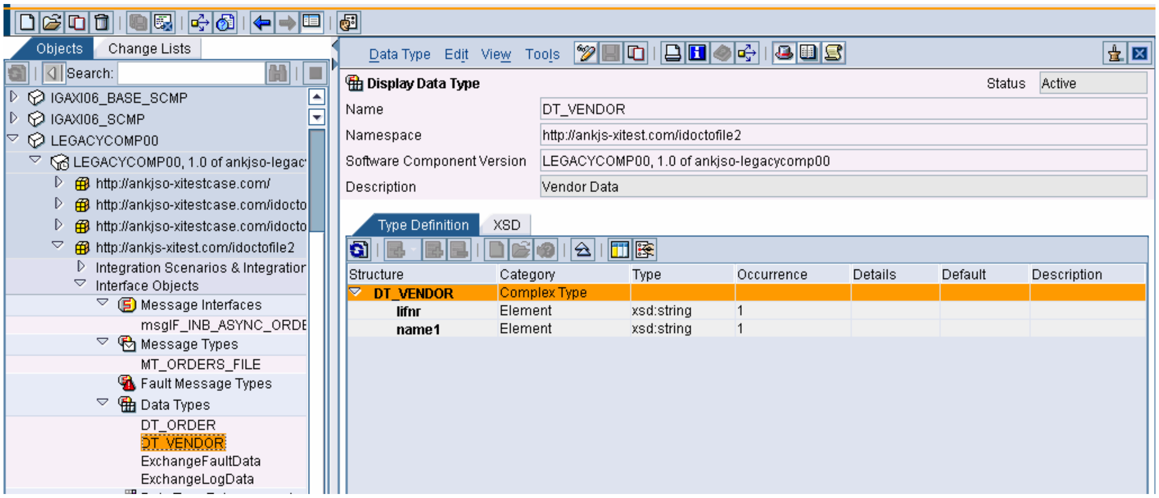
Group: SPACE

Namespaces *

Name
http://ankjso-xitestcase.com/
http://ankjso-xitestcase.com/idoctofile
http://ankjso-xitestcase.com/idoctoi
http://ankjs-xitest.com/idoctofile2

8. Create the Data Type and build the structure into which the data from R/3 IDoc shall be received. After expanding the NAME SPACE, expand the INTERFACE OBJECTS. Then right click on DATA TYPES to start creating.

For this purpose we define two data types DT_VENDOR and DT_ORDER. Use the first in second. (You can create single order data type if you want to).



Display Data Type

Status: Active

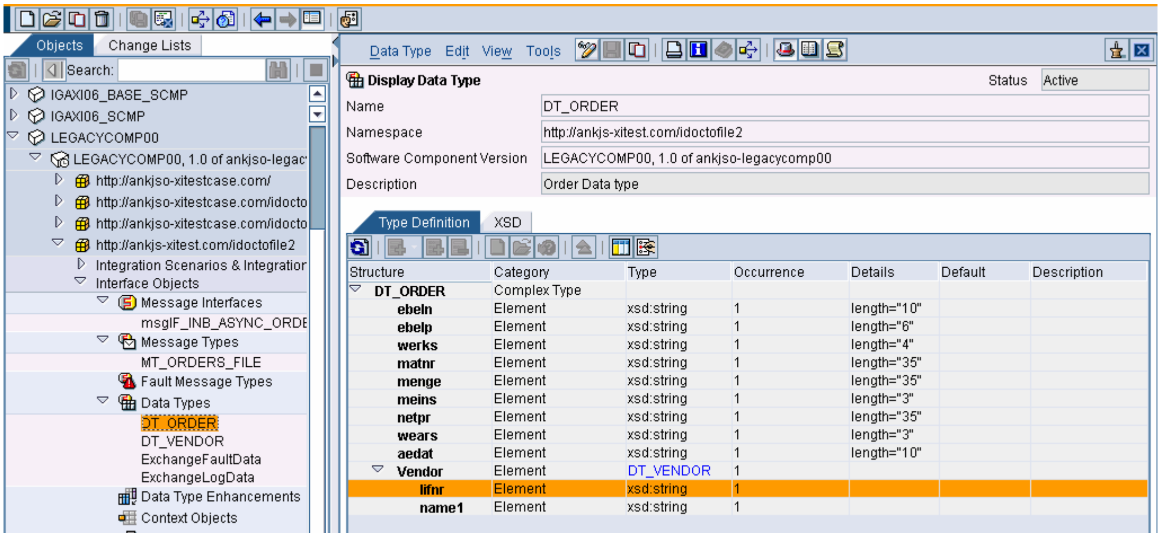
Name: DT_VENDOR

Namespace: http://ankjs-xitest.com/idoctofile2

Software Component Version: LEGACYCOMP00, 1.0 of ankjso-legacycomp00

Description: Vendor Data

Structure	Category	Type	Occurrence	Details	Default	Description
DT_VENDOR	Complex Type					
lifnr	Element	xsd:string	1			
name1	Element	xsd:string	1			



Display Data Type

Status: Active

Name: DT_ORDER

Namespace: http://ankjs-xitest.com/idoctofile2

Software Component Version: LEGACYCOMP00, 1.0 of ankjso-legacycomp00

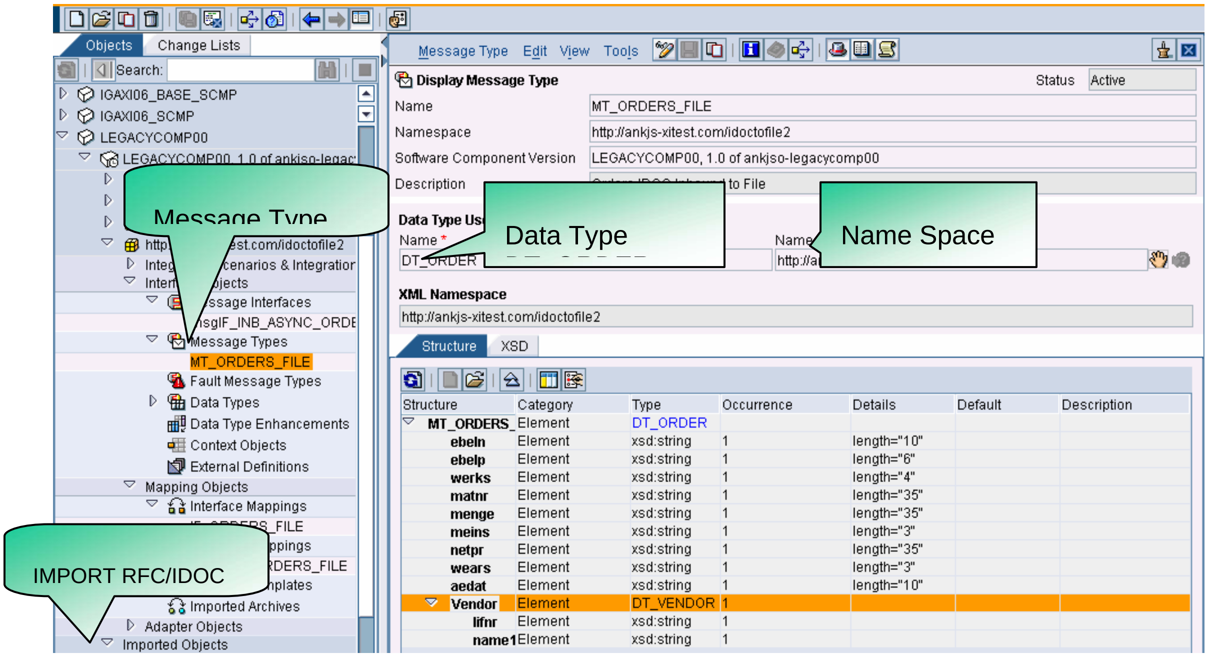
Description: Order Data type

Structure	Category	Type	Occurrence	Details	Default	Description
DT_ORDER	Complex Type					
ebeln	Element	xsd:string	1	length="10"		
ebelp	Element	xsd:string	1	length="6"		
werks	Element	xsd:string	1	length="4"		
matnr	Element	xsd:string	1	length="35"		
menge	Element	xsd:string	1	length="35"		
meins	Element	xsd:string	1	length="3"		
netpr	Element	xsd:string	1	length="35"		
wears	Element	xsd:string	1	length="3"		
aedat	Element	xsd:string	1	length="10"		
Vendor	Element	DT_VENDOR	1			
lifnr	Element	xsd:string	1			
name1	Element	xsd:string	1			

Tip 8

You can use other options too like using the EXTERNAL DEFINITIONS but you need to generate the XSD file then import it into XI. This will ease most of the work. This we will see at a later stage.

9. Create a message type and assign the data type DT_ORDER to it. Make sure you drag and drop the data type on the OPEN HAND icon shown on the right hand side next to NAME SPACE box. Or, you can drop down the list and select the one that you have created.



Display Message Type

Status: Active

Name: MT_ORDERS_FILE

Namespace: http://ankjs-xitest.com/idoctofile2

Software Component Version: LEGACYCOMP00, 1.0 of ankjso-legacycomp00

Description: to File

Data Type Us

Name *: DT_ORDER

Name: http://a

XML Namespace

http://ankjs-xitest.com/idoctofile2

Structure | XSD

Structure	Category	Type	Occurrence	Details	Default	Description
MT_ORDERS_	Element	DT_ORDER				
ebeln	Element	xsd:string	1	length="10"		
ebelp	Element	xsd:string	1	length="6"		
werks	Element	xsd:string	1	length="4"		
matnr	Element	xsd:string	1	length="35"		
menge	Element	xsd:string	1	length="35"		
meins	Element	xsd:string	1	length="3"		
netpr	Element	xsd:string	1	length="35"		
wears	Element	xsd:string	1	length="3"		
aedat	Element	xsd:string	1	length="10"		
Vendor	Element	DT_VENDOR	1			
lifnr	Element	xsd:string	1			
name1	Element	xsd:string	1			

Tip 9

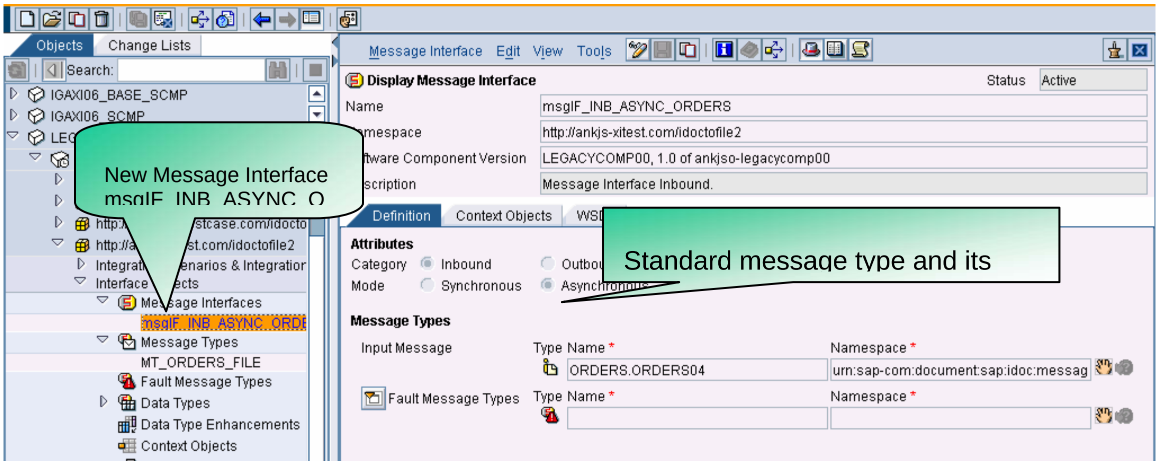
Recommend generating the DETAILS while creating the data types for the scenario. This is to avoid any disparities while sending the FILE generated from IDoc into the final destination system…..just in case.

Tip 10

At this stage, right-click on the IMPORTED OBJECTS and import the IDocs from R/3. From the resulting window select the ORDERS.ORDERS04 (message type.IDOC type from R/3) and then click Continue to import. Once this is done, EXIT from the screen. This will be your SOURCE MESSAGE TYPE that will be used in mapping the message types for data handling. This is most important step.

10. Create the message interface. To do this, right-click on the MESSAGE INTERFACE title and create the new object. Here we will make sure that the attributes are set to ASYNCHRONOUS (will see this later) and INBOUND (into XI from R/3).

Drag and drop the ORDERS.ORDERS04 from Imported Objects→IDOC on to the hand icon on the right-hand side next to name space field. Or drop down the list then select the message type from the resulting window. Or simply type the details.



Display Message Interface

Status: Active

Name: msgIF_INB_ASYNC_ORDERS

amespace: http://ankjs-xitest.com/idoctofile2

ftware Component Version: LEGACYCOMP00, 1.0 of ankjso-legacycomp00

scription: Message Interface Inbound.

Attributes

Category ● Inbound ○ Outbou

Mode ○ Synchronous ● Asynchronous

Message Types

Input Message — Type Name *: ORDERS.ORDERS04 — Namespace *: urn:sap-com:document:sap:idoc:messag

Fault Message Types — Type Name *: — Namespace *:

11. At this stage we are done with definitions and now its time to map the messages and interface created thus far.

To begin with, click on the triangle pointing rightwards on MAPPING OBJECTS. Right-click on MESSAGE MAPPING and create a new map. This will result in a graphical window wherein you drag and drop the source (ORDERS.ORDERS04 from IMPORTED OBJECTS) and the target message MT_ORDERS_FILE created as stated above.

This is needed to map the SOURCE filed to the TARGET field. In our case the source is the R/3 IDoc coming into XI from SAP R/3 and the target is the FILE stored at root directory that will then be accessed by any script to load in to the destination system.

By doing this, we are creating a 1-1 kind of relationshop between the source and target.

Tip 10

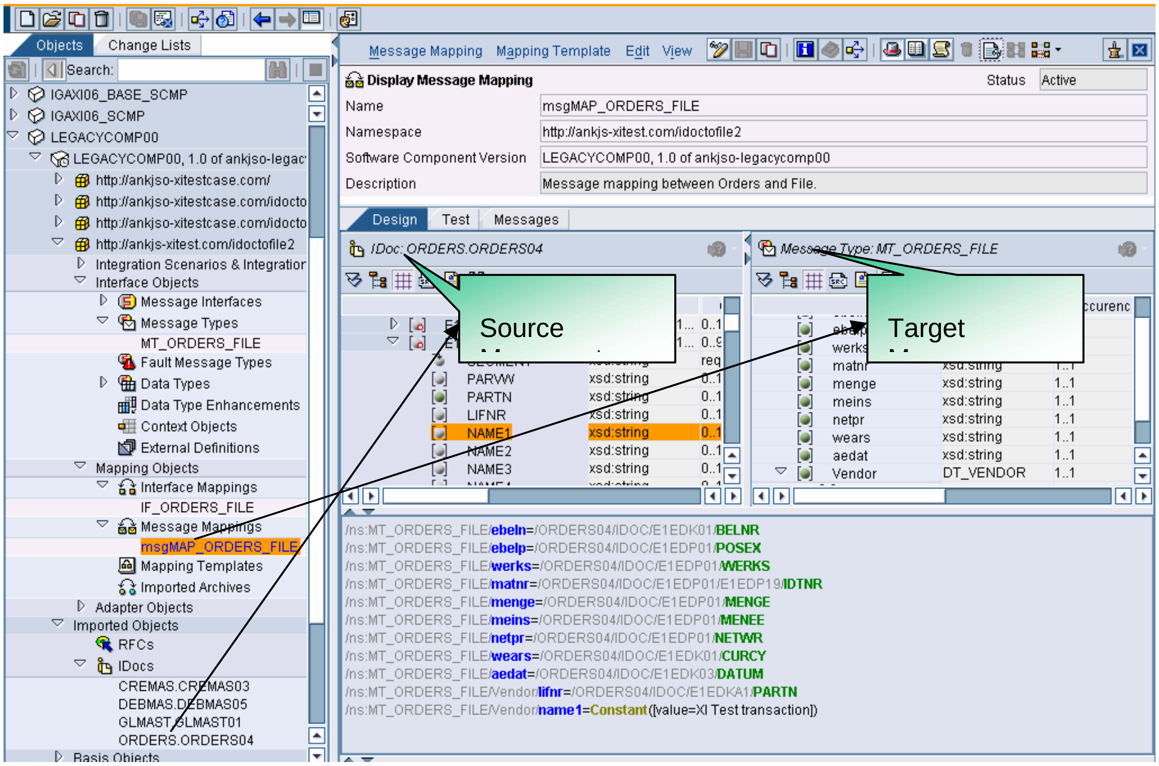
Referring to TIP9, add the external definition by clicking F4 on TARGET filed on right hand side then select the IMPORTED MESSAGE TYPE. This will add the entire structure for easing the mapping function.

Tip 11

Generate simple functions for performing various string operations, minor conversions, assigning the date (CURRENT DATE) to any of the target fields. This may be recommended some times.

Tip 12

Assign CONSTANT with value 1 to BEGIN if you see it in the TARGET field list at this stage.



Display Message Mapping

Status: Active

Name: msgMAP_ORDERS_FILE

Namespace: http://ankjs-xitest.com/idoctofile2

Software Component Version: LEGACYCOMP00, 1.0 of ankjso-legacycomp00

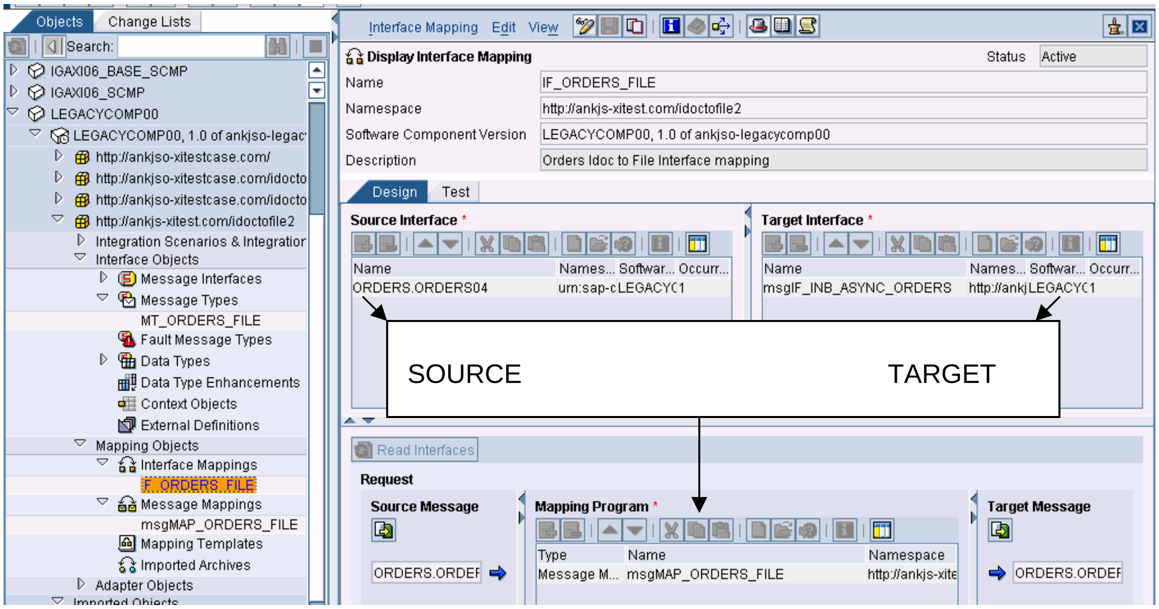
Description: Message mapping between Orders and File.

```
/ns:MT_ORDERS_FILE/ebeln=/ORDERS04/IDOC/E1EDK01/BELNR
/ns:MT_ORDERS_FILE/ebelp=/ORDERS04/IDOC/E1EDP01/POSEX
/ns:MT_ORDERS_FILE/werks=/ORDERS04/IDOC/E1EDP01/WERKS
/ns:MT_ORDERS_FILE/matnr=/ORDERS04/IDOC/E1EDP01/E1EDP19/IDTNR
/ns:MT_ORDERS_FILE/menge=/ORDERS04/IDOC/E1EDP01/MENGE
/ns:MT_ORDERS_FILE/meins=/ORDERS04/IDOC/E1EDP01/MENEE
/ns:MT_ORDERS_FILE/netpr=/ORDERS04/IDOC/E1EDP01/NETWR
/ns:MT_ORDERS_FILE/wears=/ORDERS04/IDOC/E1EDK01/CURCY
/ns:MT_ORDERS_FILE/aedat=/ORDERS04/IDOC/E1EDK03/DATUM
/ns:MT_ORDERS_FILE/Vendor/lifnr=/ORDERS04/IDOC/E1EDKA1/PARTN
/ns:MT_ORDERS_FILE/Vendor/name1=Constant([value=XI Test transaction])
```

12. Now, the last step in the Integration Repository is to create the Interface Mapping. In this step map the source (ORDERS.ORDERS04) to the target interface (msgIF_INB_ASYNC_ORDERS). Once the objects are identified and filled into respective fields, click on the "Read Interfaces" push button on the right-hand side in the bottom portion of the work area. This replicates the source and targets into the Request Tab Window. Identify the message mapping program created in the above step and assign it here.

We are done with SOURCE, TARGET, and LINK for data handling.

At the end, save the work and activate the change list.



Display Interface Mapping Status: Active

Name: IF_ORDERS_FILE
Namespace: http://ankjs-xitest.com/idoctofile2
Software Component Version: LEGACYCOMP00, 1.0 of ankjso-legacycomp00
Description: Orders Idoc to File Interface mapping

Source Interface *

Name	Names...	Softwar...	Occurr...
ORDERS.ORDERS04	urn:sap-c	LEGACYC	1

Target Interface *

Name	Names...	Softwar...	Occurr...
msgIF_INB_ASYNC_ORDERS	http://ankj	LEGACYC	1

Source Message: ORDERS.ORDEF

Mapping Program *

Type	Name	Namespace
Message M...	msgMAP_ORDERS_FILE	http://ankjs-xite

Target Message: ORDERS.ORDEF

Tip 13

Time to check the object consistency. Make sure to hit “**F7**” at this stage just before activating.

14. Minimize the Integration Repository and log into Integration Directory. This is the place where we create the business scenario, communication medium, collaboration agreements between partners (more clear if I say receiver and sender agreement). But in our case, the idea is to get the IDoc from R/3 and hence we don’t need a sender configuration or sender agreement.

Tip 14

No senders agreement for inbound IDoc flow from R/3 to XI. We need identify the sender only.

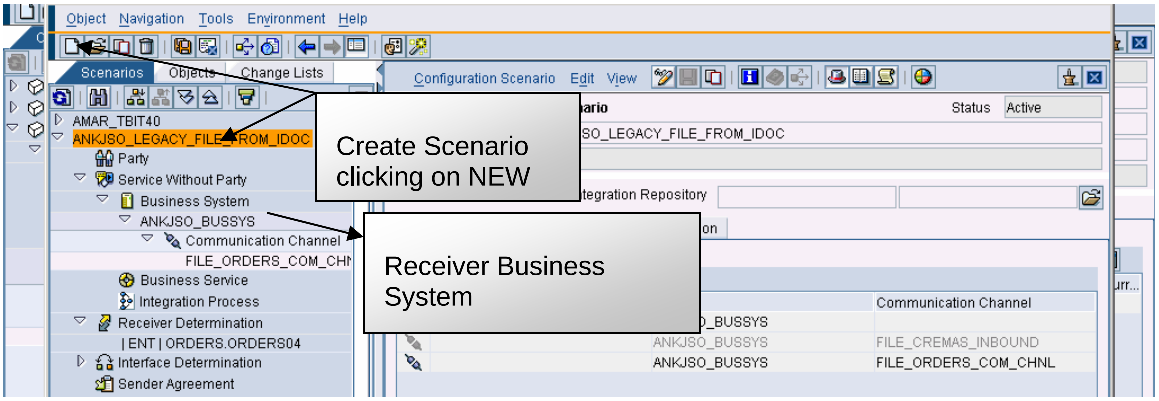
Tip 15

In this case, *we don’t need a communication channel for the SENDER to communicate to XI*. This will be taken care by RFC. Refer to the settings we maintained in the beginning of this column.

Tip 16

Since sender is an R/3 system and medium is IDoc, it resides in the ABAP Stack. In fact it resides outside the adapter framework in the Integration Server. So, monitoring from the adapter framework is not possible while INBOUND into XI. This can be monitored using IDoc monitoring transaction IDX5.

15. Let's begin with Configuration. As the first step create the Business scenario and to do this, keep the cursor anywhere (normally on the last scenario in the list) and click on NEW SCENARIO icon from tool bar. Enter a unique name that will be easy to remember.



Create Scenario clicking on NEW

Receiver Business System

AMAR_TBIT40
ANKJSO_LEGACY_FILE_FROM_IDOC
Party
Service Without Party
Business System
ANKJSO_BUSSYS
Communication Channel
FILE_ORDERS_COM_CHN
Business Service
Integration Process
Receiver Determination
| ENT | ORDERS.ORDERS04
Interface Determination
Sender Agreement

Status: Active
SO_LEGACY_FILE_FROM_IDOC

	Communication Channel
_BUSSYS	
ANKJSO_BUSSYS	FILE_CREMAS_INBOUND
ANKJSO_BUSSYS	FILE_ORDERS_COM_CHNL

16. Assign the business systems to the scenario. To do this, drop down the services without party then right-click on the business systems to assign. I would advise not using the graphical editor as you will not know what is happening in the background.

Tip 17

When you see the object not appearing the list of business systems, there could be few reasons why it disappears from the list. Make sure that SLD cache is cleared and if nothing works, try to locate the object manually navigating in the object list.

Tip 18

Make sure that you uncheck the "Create Communication Channel" check box while assigning the business system.

17. We have assigned the Receiver Business System and no sender assignment is needed at this stage. For the receiver Business system, we now create the communication channel. We know that we are mapping the IDOC to a File, so the Adapter we are going to use shall be a FILE Adapter. A point to make at this stage is to understand the COMMUNICATION CHANNEL as an INSTANCE of ADAPTER. Meaning, it is going to be the run time component of ADAPTER. (In OOPS, we call this as Instantiation as the part of INHERITANCE).

Tip 19

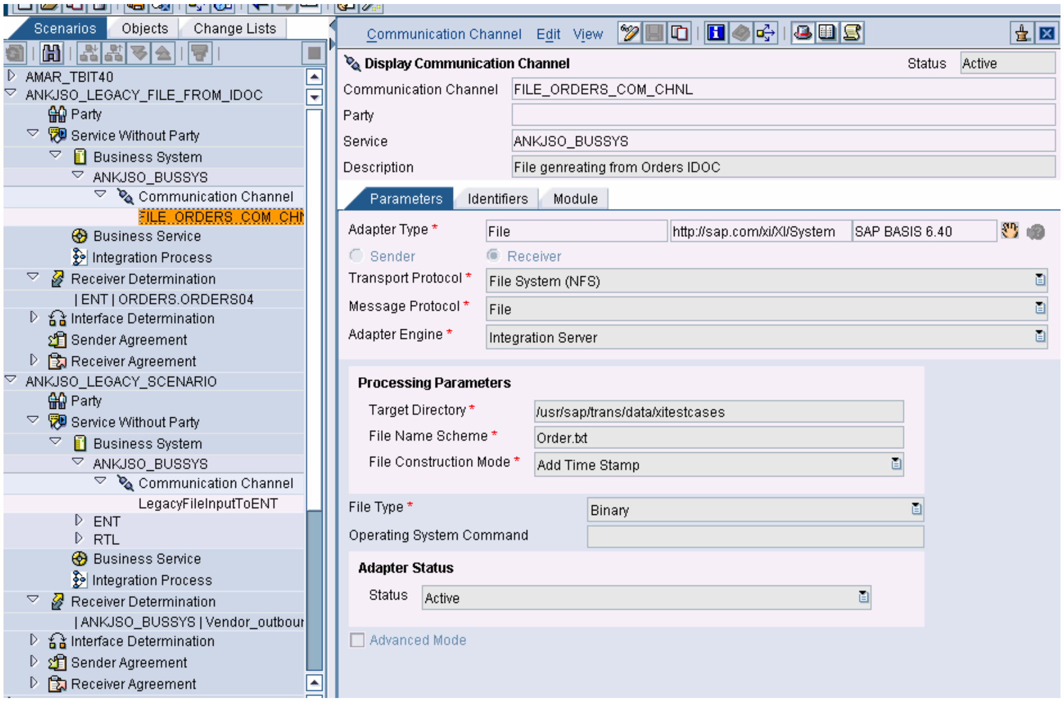
FILE ADAPTER is developed on J2EE environment and resides in the Java stack.

Tip 21

FILE Adapter/Communication channel activities can be monitored in the adapter framework. In our case, this will be seen at the final stage as IDoc will be finally saved into a file. In case of errors at this stage, we can find out by monitoring the adapter framework.

Tip 22

RUN TIME WORK BENCH is beyond the adapter framework monitoring. It keeps the complete audit log of the message while processing.



Display Communication Channel

Status: Active

Communication Channel: FILE_ORDERS_COM_CHNL

Party:

Service: ANKJSO_BUSSYS

Description: File genreating from Orders IDOC

Adapter Type *: File | http://sap.com/xi/XI/System | SAP BASIS 6.40

○ Sender ● Receiver

Transport Protocol *: File System (NFS)

Message Protocol *: File

Adapter Engine *: Integration Server

Processing Parameters

Target Directory *: /usr/sap/trans/data/xitestcases

File Name Scheme *: Order.txt

File Construction Mode *: Add Time Stamp

File Type *: Binary

Operating System Command:

Adapter Status

Status: Active

□ Advanced Mode

Check the attributes and make sure that you have specified the file location (with UNIX PATH but of course with FORWARD SLASHES, file name, time stamp if needed to make sure that we create a new file every time we receive an IDoc. Don’t forget to choose the RECEIVER radio button as FILE WILL BE RECEIVED by this channel at the end.

Tip 22

Hit “**F7**” at this stage to ensure the consistency. In case of errors, check the process log for necessary corrections. If needed, activate the change list as may be needed.

18. Create the RECEIVER DETERMINATION. (Make a note of TIPs 13-15) before going any further. We understand that RECEIVER DETERMINATION is only needed in our case and is required for identifying the sender, interface, and etc.

To do this, right-click on the receiver determination and create new.

Step 1: Locate the sender from the drop down list on SENDER field. Since this is an R/3 system, choose and click on "TIK" icon at the bottom of the window.

Step 2: Identify the interface again from the drop down list. In our case it is ORDERS.ORDERS04, locate it and repeat the same as stated above.

Step 3: Click on create, just save once in the resulting main window.

Step 4: Identify the receiver from the drop down list on SERVICE field in the CONFIGURED RECEIVERS sub screen. In our case it is the business system we assigned to the scenario in the previous steps. Locate and assign here for further processing.

Step 5: Save at this stage and notice that CONFIGURATION OVERVIEW FOR RECEIVER DETERMINATION is updated with the receiver business system we identified in the step-4.

Step 6: Expand by clicking on the triangle pointing rightwards on the receiver business system in the overview sub-screen. You will notice three text strings in red color which needs to be identified now.

Step 7: Right-click on the first text item and choose "**New Specific**". This will open a new window and is for identifying the message interface (created in the integration repository) and the corresponding Interface. This step will assign the above to two of the three text items.

Step 8: Save, close then click on refresh icon in the configuration overview sub-screen.

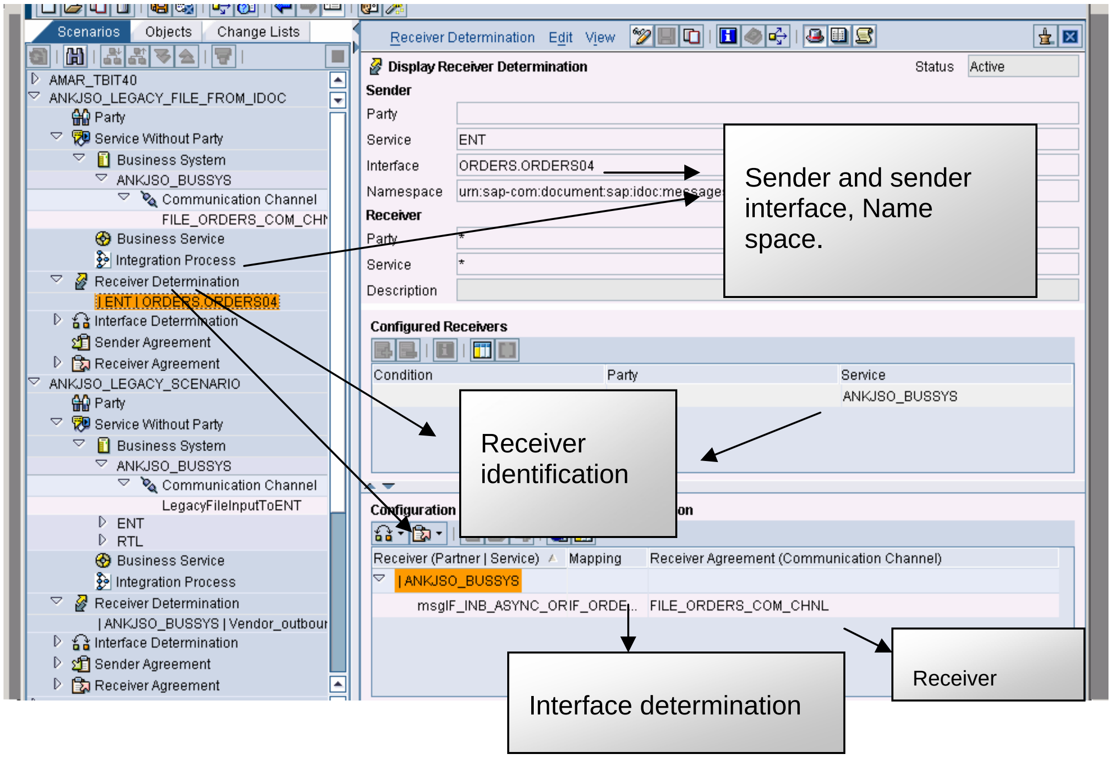
Step 9: Right-click on the third text item, then assign the receiver communication channel. Repeat what we have done in step-8.

What have we done? We have executed three crucial steps in this whole exercise:

1. Receiver Determination

2. Interface Determination

3. Receiver Agreement



Display Receiver Determination

Status: Active

Sender

Party:
Service: ENT
Interface: ORDERS.ORDERS04
Namespace: urn:sap-com:document:sap:idoc:message

Receiver

Party: *
Service: *
Description:

Configured Receivers

Condition	Party	Service
		ANKJSO_BUSSYS

Configuration

Receiver (Partner | Service)	Mapping	Receiver Agreement (Communication Channel)
| ANKJSO_BUSSYS		
msgIF_INB_ASYNC_ORIF_ORDE...		FILE_ORDERS_COM_CHNL

Sender and sender interface, Name space.

Receiver identification

Interface determination

Receiver

Save all changes. Activate the changes after checking the consistency (F7).

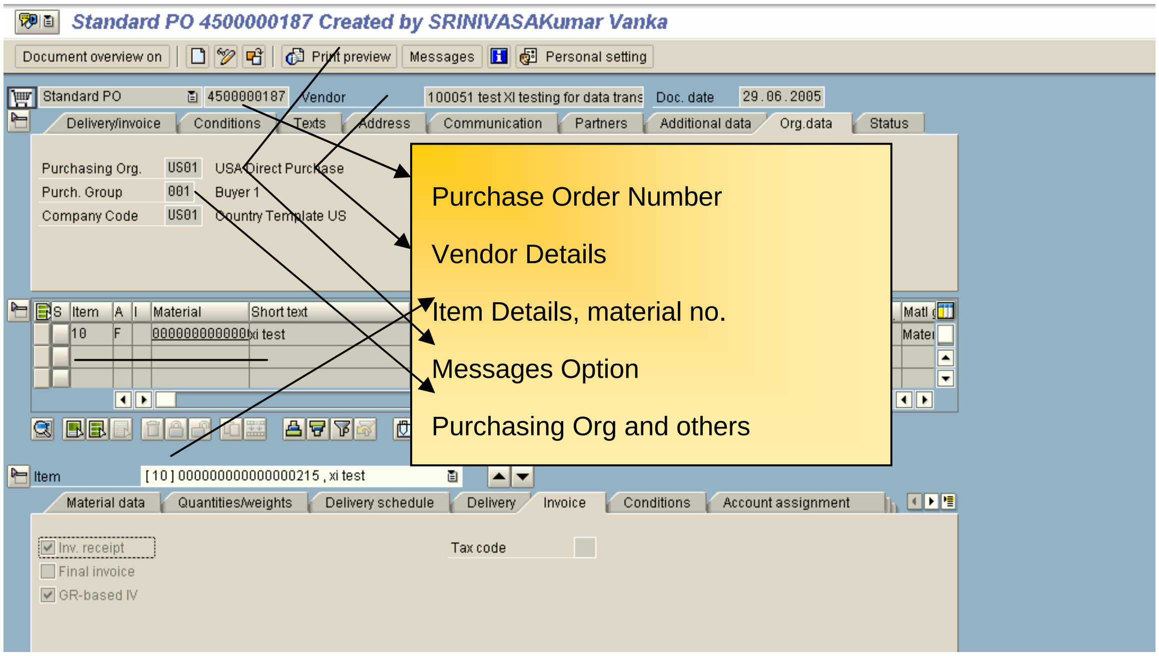
Now, we are all set to test what we have done.

How to Test

Log into SAP R/3, display the purchase order using ME23N (ME23 is its old transaction name). To create the purchase order, you have already collected the data in the first step. (For details refer to points A,B and C under step-2 in R/3 settings.)

Save the PO and click on GO TO → MESSAGES. Or choose MESSAGES pushbutton on the application toolbar. This opens another window where you key in the MESSAGE CONTROL DATA you have maintained in the outbound parameters in the PARTNER PROFILE configuration.

Once after the output is triggered and processed, click on Process Log to check the IDoc generated as the result of issuing the output. The following screen shots should look similar to what you have done.



Standard PO 4500000187 Created by SRINIVASAKumar Vanka

Standard PO 4500000187 Vendor 100051 test XI testing for data trans Doc. date 29.06.2005

Purchasing Org. US01 USA Direct Purchase
Purch. Group 001 Buyer 1
Company Code US01 Country Template US

S	Item	A	I	Material	Short text
	10		F	000000000000	xi test

Item [10] 000000000000000215 , xi test

☑ Inv. receipt
☐ Final invoice
☑ GR-based IV
Tax code

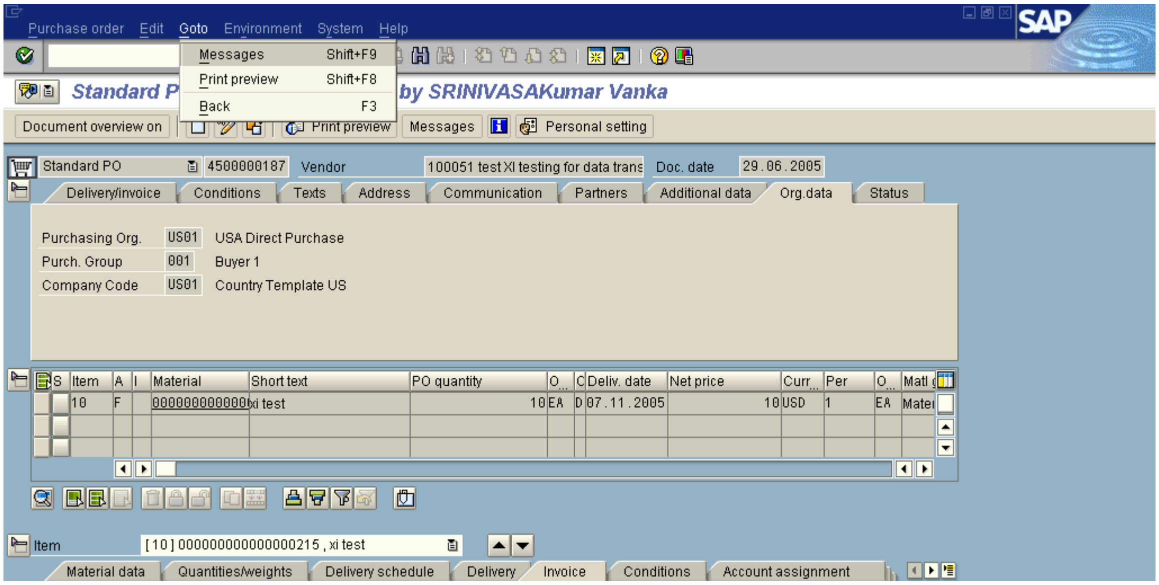
Purchase Order Number

Vendor Details

Item Details, material no.

Messages Option

Purchasing Org and others



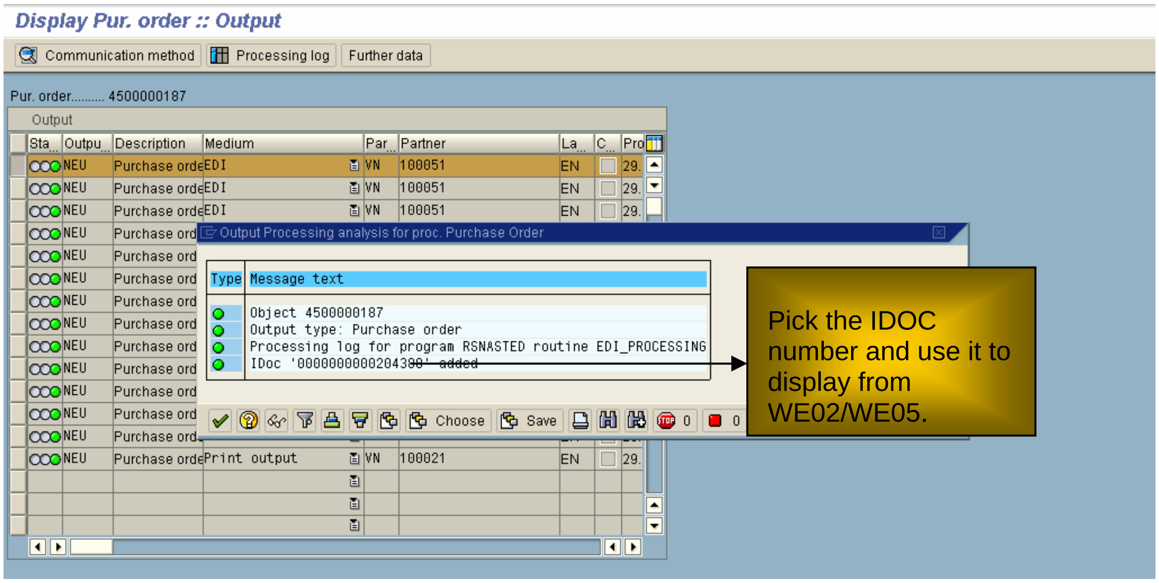
Standard PO 4500000187 Created by SRINIVASAKumar Vanka

Standard PO 4500000187 Vendor 100051 test XI testing for data trans Doc. date 29.06.2005

Purchasing Org. US01 USA Direct Purchase
Purch. Group 001 Buyer 1
Company Code US01 Country Template US

S	Item	A	I	Material	Short text	PO quantity	O...	C	Deliv. date	Net price	Curr...	Per	O...	Matl
	10		F	000000000000	xi test	10	EA	D	07.11.2005	10	USD	1	EA	Mater

Item [10] 000000000000000215 , xi test



Display Pur. order :: Output

Pur. order.......... 4500000187

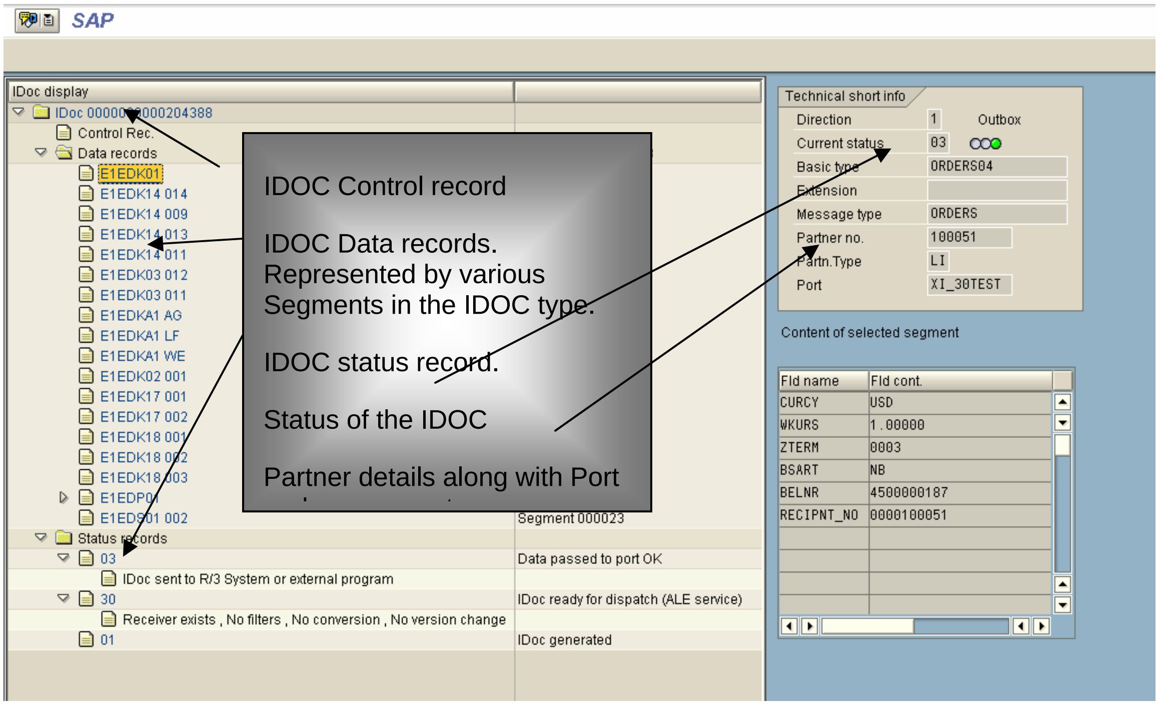
Output

Sta...	Outpu...	Description	Medium	Par...	Partner	La...	C...	Pro
○○●	NEU	Purchase orde	EDI	VN	100051	EN	□	29.
○○●	NEU	Purchase orde	EDI	VN	100051	EN	□	29.
○○●	NEU	Purchase orde	EDI	VN	100051	EN	□	29.
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase ord						
○○●	NEU	Purchase orde	Print output	VN	100021	EN	□	29.

Output Processing analysis for proc. Purchase Order

Type	Message text
●	Object 4500000187
●	Output type: Purchase order
●	Processing log for program RSNASTED routine EDI_PROCESSING
●	IDoc '0000000000204388' added

Pick the IDOC number and use it to display from WE02/WE05.



SAP

IDoc display	
IDoc 0000000000204388	
Control Rec.	
Data records	
E1EDK01	
E1EDK14 014	
E1EDK14 009	
E1EDK14 013	
E1EDK14 011	
E1EDK03 012	
E1EDK03 011	
E1EDKA1 AG	
E1EDKA1 LF	
E1EDKA1 WE	
E1EDK02 001	
E1EDK17 001	
E1EDK17 002	
E1EDK18 001	
E1EDK18 002	
E1EDK18 003	
E1EDP01	
E1EDS01 002	Segment 000023
Status records	
03	Data passed to port OK
IDoc sent to R/3 System or external program	
30	IDoc ready for dispatch (ALE service)
Receiver exists , No filters , No conversion , No version change	
01	IDoc generated

Technical short info

Direction	1	Outbox
Current status	03	○○●
Basic type	ORDERS04	
Extension		
Message type	ORDERS	
Partner no.	100051	
Partn.Type	LI	
Port	XI_30TEST	

Content of selected segment

Fld name	Fld cont.
CURCY	USD
WKURS	1.00000
ZTERM	0003
BSART	NB
BELNR	4500000187
RECIPNT_NO	0000100051

IDOC Control record

IDOC Data records. Represented by various Segments in the IDOC type.

IDOC status record.

Status of the IDOC

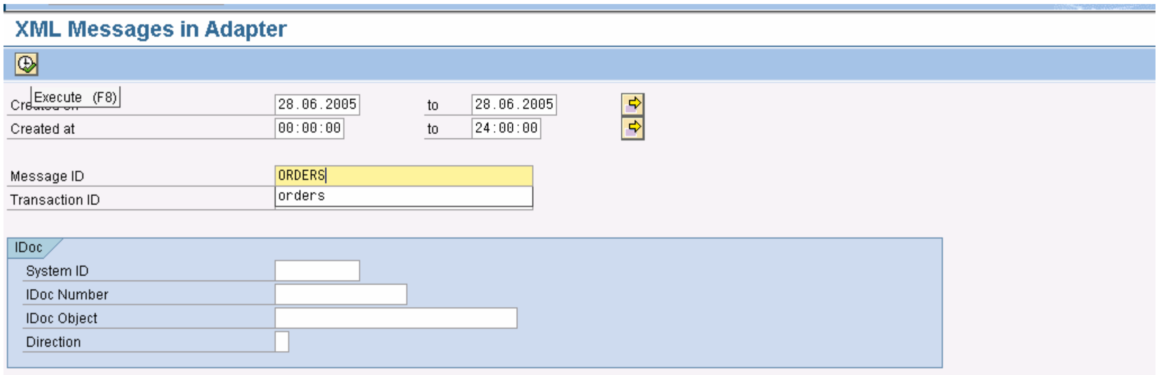
Partner details along with Port

The IDoc is generated in R/3 and the status is indicating that the IDoc has been dispatched to the sender program and has been sent to its destination. Sometimes it may be helpful to know the IDoc status codes, but just for your knowledge.

Now, time to look in XI.

Where Do You See it in SAP XI?

Incoming/processing messages in R/3 can be viewed using the monitoring tools. For our purpose we first use IDX5 to check whether any inbound IDocs in XI for the ORDERS message type. To do this, log into XI, type IDX5 then check the dates, feed the logical message type then hit F8.



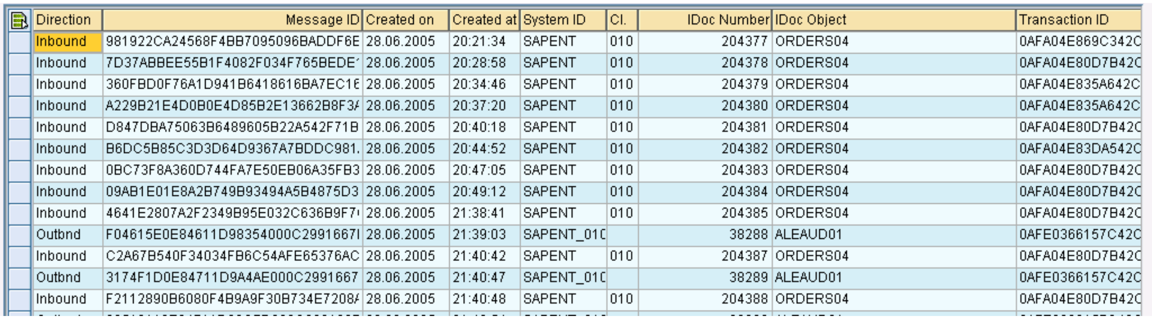
XML Messages in Adapter

Execute (F8)

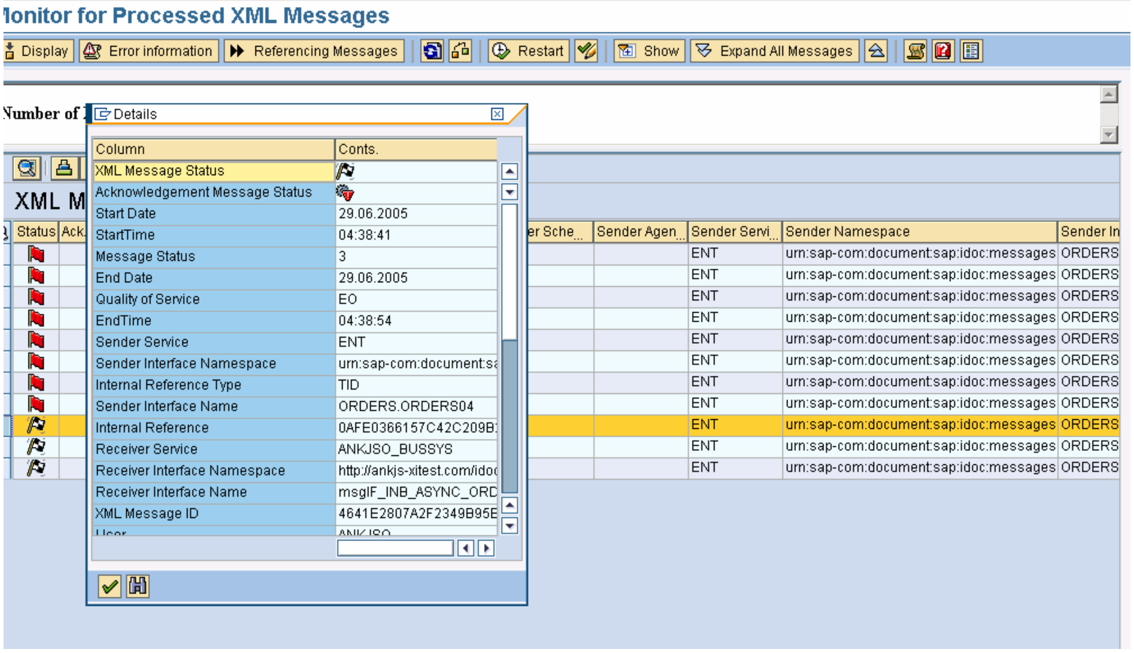
Created on	28.06.2005	to	28.06.2005
Created at	00:00:00	to	24:00:00
Message ID	ORDERS		
Transaction ID	orders		

IDoc

System ID	
IDoc Number	
IDoc Object	
Direction	



Direction	Message ID	Created on	Created at	System ID	Cl.	IDoc Number	IDoc Object	Transaction ID
Inbound	981922CA24568F4BB7095096BADDF6E	28.06.2005	20:21:34	SAPENT	010	204377	ORDERS04	0AFA04E869C342C
Inbound	7D37ABBEE55B1F4082F034F765BEDE	28.06.2005	20:28:58	SAPENT	010	204378	ORDERS04	0AFA04E80D7B42C
Inbound	360FBD0F76A1D941B6418616BA7EC1E	28.06.2005	20:34:46	SAPENT	010	204379	ORDERS04	0AFA04E835A642C
Inbound	A229B21E4D0B0E4D85B2E13662B8F3A	28.06.2005	20:37:20	SAPENT	010	204380	ORDERS04	0AFA04E835A642C
Inbound	D847DBA75063B6489605B22A542F71B	28.06.2005	20:40:18	SAPENT	010	204381	ORDERS04	0AFA04E80D7B42C
Inbound	B6DC5B85C3D3D64D9367A7BDDC981.	28.06.2005	20:44:52	SAPENT	010	204382	ORDERS04	0AFA04E83DA542C
Inbound	0BC73F8A360D744FA7E50EB06A35FB3	28.06.2005	20:47:05	SAPENT	010	204383	ORDERS04	0AFA04E80D7B42C
Inbound	09AB1E01E8A2B749B93494A5B4875D3	28.06.2005	20:49:12	SAPENT	010	204384	ORDERS04	0AFA04E80D7B42C
Inbound	4641E2807A2F2349B95E032C636B9F7	28.06.2005	21:38:41	SAPENT	010	204385	ORDERS04	0AFA04E80D7B42C
Outbnd	F04615E0E84611D98354000C2991667I	28.06.2005	21:39:03	SAPENT_010		38288	ALEAUD01	0AFE0366157C42C
Inbound	C2A67B540F34034FB6C54AFE65376AC	28.06.2005	21:40:42	SAPENT	010	204387	ORDERS04	0AFA04E80D7B42C
Outbnd	3174F1D0E84711D9A4AE000C2991667	28.06.2005	21:40:47	SAPENT_010		38289	ALEAUD01	0AFE0366157C42C
Inbound	F2112890B6080F4B9A9F30B734E7208A	28.06.2005	21:40:48	SAPENT	010	204388	ORDERS04	0AFA04E80D7B42C

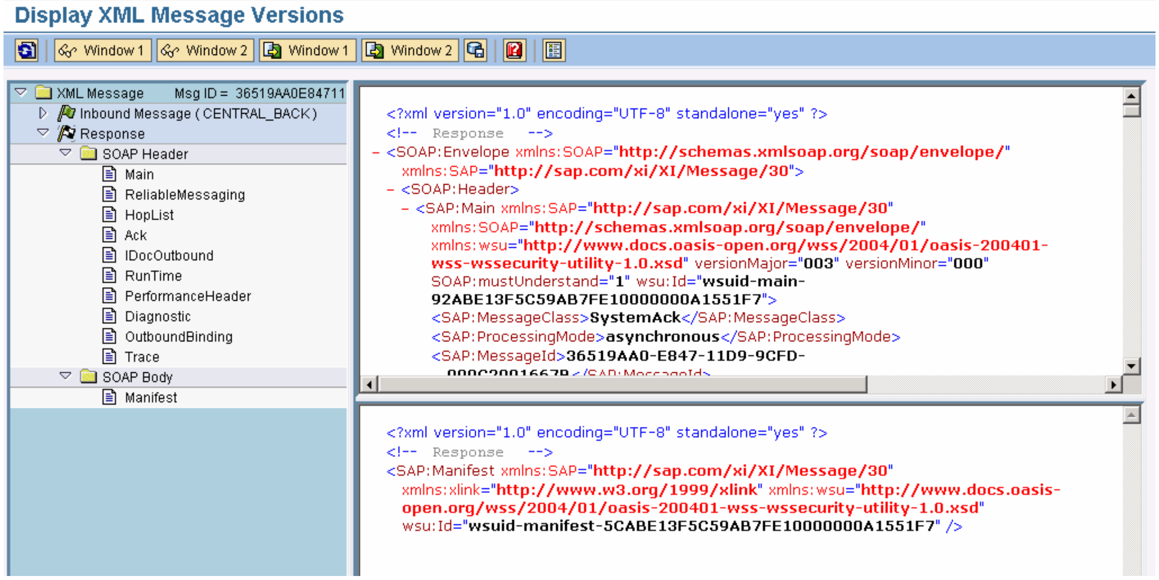


Details

Column	Conts.
XML Message Status	🏁
Acknowledgement Message Status	⚙
Start Date	29.06.2005
StartTime	04:38:41
Message Status	3
End Date	29.06.2005
Quality of Service	EO
EndTime	04:38:54
Sender Service	ENT
Sender Interface Namespace	urn:sap-com:document:sa
Internal Reference Type	TID
Sender Interface Name	ORDERS.ORDERS04
Internal Reference	0AFE0366157C42C209B
Receiver Service	ANKJSO_BUSSYS
Receiver Interface Namespace	http://ankjs-xitest.com/idoc
Receiver Interface Name	msgIF_INB_ASYNC_ORD
XML Message ID	4641E2807A2F2349B95E
User	ANKJSO

Status	Ack.	er Sche...	Sender Agen...	Sender Servi...	Sender Namespace	Sender In
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
⚑				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
🏁				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
🏁				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS
🏁				ENT	urn:sap-com:document:sap:idoc:messages	ORDERS

Select the message then double-click to display the detailed XML message.



```
▽ XML Message   Msg ID = 36519AA0E84711
  ▷ Inbound Message ( CENTRAL_BACK )
  ▽ Response
    ▽ SOAP Header
        Main
        ReliableMessaging
        HopList
        Ack
        IDocOutbound
        RunTime
        PerformanceHeader
        Diagnostic
        OutboundBinding
        Trace
    ▽ SOAP Body
        Manifest
```

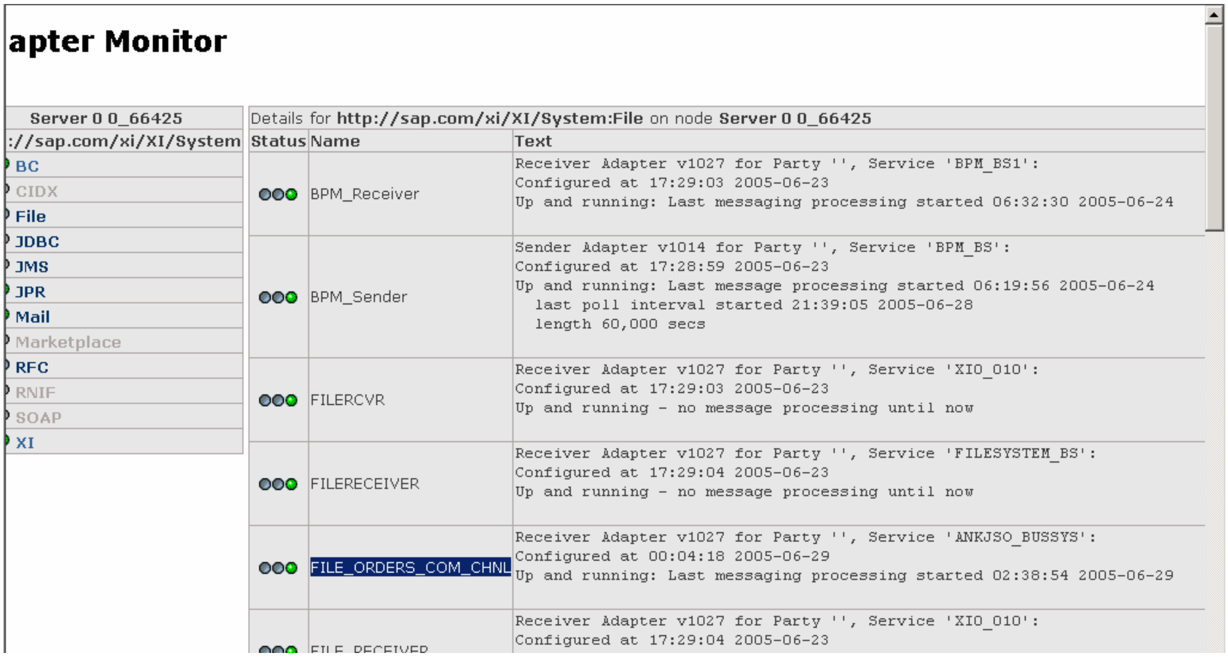
```
 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
 <!--  Response   -->
- <SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:SAP="http://sap.com/xi/XI/Message/30">
 - <SOAP:Header>
  - <SAP:Main xmlns:SAP="http://sap.com/xi/XI/Message/30"
      xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wsu="http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-
      wss-wssecurity-utility-1.0.xsd" versionMajor="003" versionMinor="000"
      SOAP:mustUnderstand="1" wsu:Id="wsuid-main-
      92ABE13F5C59AB7FE10000000A1551F7">
      <SAP:MessageClass>SystemAck</SAP:MessageClass>
      <SAP:ProcessingMode>asynchronous</SAP:ProcessingMode>
      <SAP:MessageId>36519AA0-E847-11D9-9CFD-
        000C2001667B</SAP:MessageId>
```

```
 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
 <!--  Response   -->
 <SAP:Manifest xmlns:SAP="http://sap.com/xi/XI/Message/30"
   xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:wsu="http://www.docs.oasis-
   open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
   wsu:Id="wsuid-manifest-5CABE13F5C59AB7FE10000000A1551F7" />
```

In case of failures, read the upper portion of the right-hand side work area with utmost care, as it describes why the data transmission is failed. This is applicable only when the IDoc from R/3 is dispatched to XI → destination host successfully.

File Adapter

Now, let's see the file adapter in the adapter framework and see whether the adapter picked up the XML message and saved it as a file at the specified directory.



apter Monitor

Server 0 0_66425

://sap.com/xi/XI/System

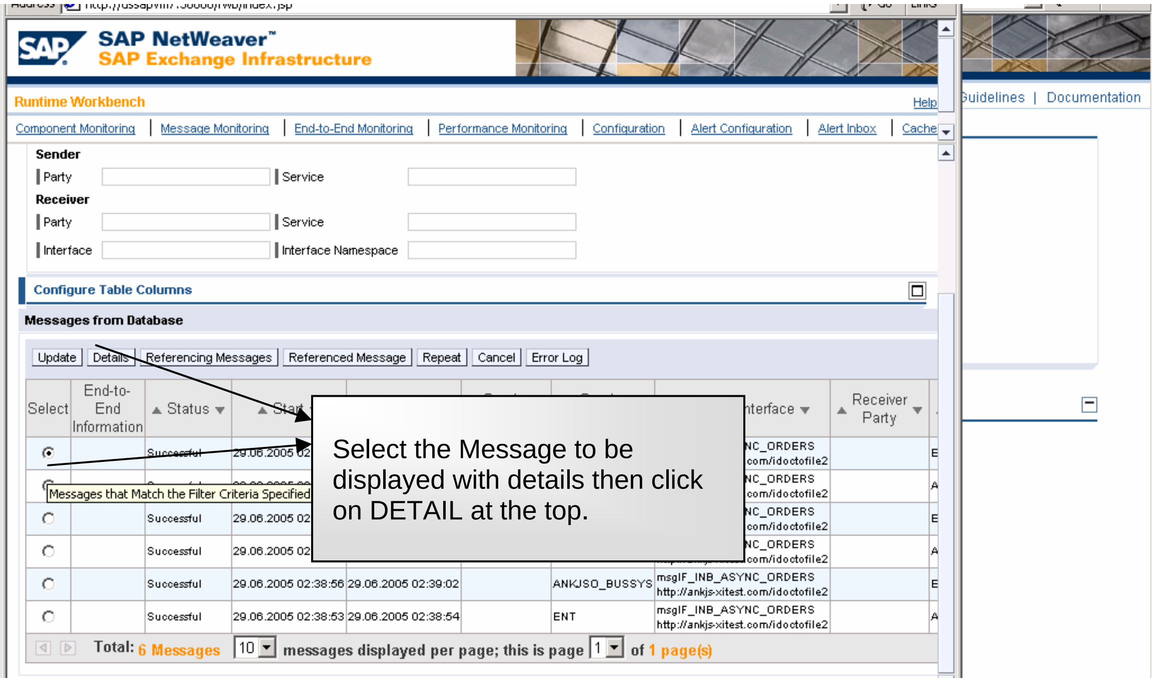
- BC
- CIDX
- File
- JDBC
- JMS
- JPR
- Mail
- Marketplace
- RFC
- RNIF
- SOAP
- XI

Details for **http://sap.com/xi/XI/System:File** on node **Server 0 0_66425**

Status	Name	Text
○○●	BPM_Receiver	Receiver Adapter v1027 for Party '', Service 'BPM_BS1': Configured at 17:29:03 2005-06-23 Up and running: Last messaging processing started 06:32:30 2005-06-24
○○●	BPM_Sender	Sender Adapter v1014 for Party '', Service 'BPM_BS': Configured at 17:28:59 2005-06-23 Up and running: Last message processing started 06:19:56 2005-06-24 last poll interval started 21:39:05 2005-06-28 length 60,000 secs
○○●	FILERCVR	Receiver Adapter v1027 for Party '', Service 'XI0_010': Configured at 17:29:03 2005-06-23 Up and running - no message processing until now
○○●	FILERECEIVER	Receiver Adapter v1027 for Party '', Service 'FILESYSTEM_BS': Configured at 17:29:04 2005-06-23 Up and running - no message processing until now
○○●	FILE_ORDERS_COM_CHNL	Receiver Adapter v1027 for Party '', Service 'ANKJSO_BUSSYS': Configured at 00:04:18 2005-06-29 Up and running: Last messaging processing started 02:38:54 2005-06-29
○○●	FILE_RECEIVER	Receiver Adapter v1027 for Party '', Service 'XI0_010': Configured at 17:29:04 2005-06-23

Check the file using transaction AL11 in XI, by locating the appropriate folder/file. Verify the contents and compare with what you were looking for.

If you have no authorizations to display the file using AL11, go to RUN TIME WORKBENCH from Integration Builder, choose message monitoring. Enter the adapter framework details then click on the DETAIL to run your query. The following screen shots depict the flow.

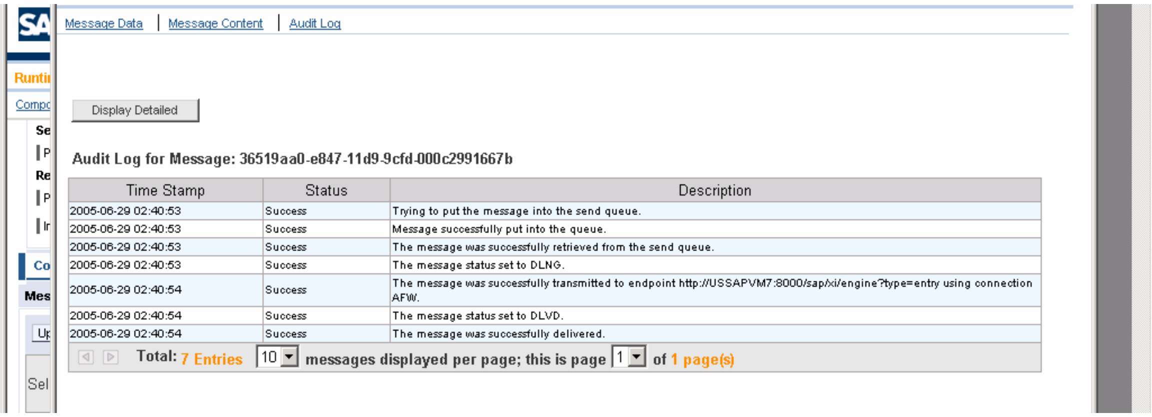


Select	End-to-End Information	Status	Start				nterface	Receiver Party	
●		Successful	29.06.2005 02				NC_ORDERS com/idoctofile2		E
○							NC_ORDERS com/idoctofile2		A
○		Successful	29.06.2005 02				NC_ORDERS com/idoctofile2		E
○		Successful	29.06.2005 02				NC_ORDERS com/idoctofile2		A
○		Successful	29.06.2005 02:38:56	29.06.2005 02:39:02		ANKJSO_BUSSYS	msgIF_INB_ASYNC_ORDERS http://ankjs-xitest.com/idoctofile2		E
○		Successful	29.06.2005 02:38:53	29.06.2005 02:38:54		ENT	msgIF_INB_ASYNC_ORDERS http://ankjs-xitest.com/idoctofile2		A

Messages that Match the Filter Criteria Specified

Total: 6 Messages 10 messages displayed per page; this is page 1 of 1 page(s)

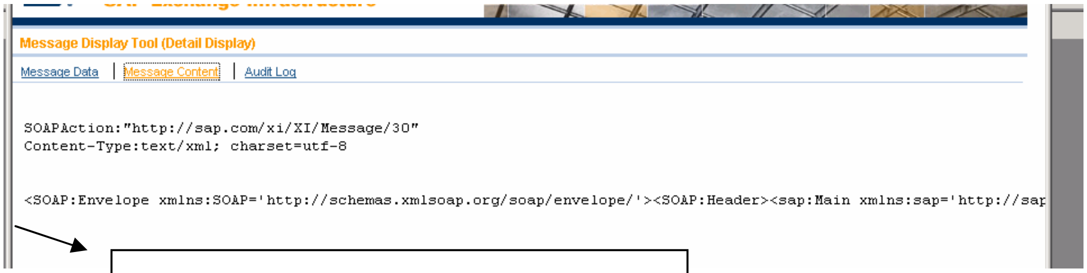
Select the Message to be displayed with details then click on DETAIL at the top.



Audit Log for Message: 36519aa0-e847-11d9-9cfd-000c2991667b

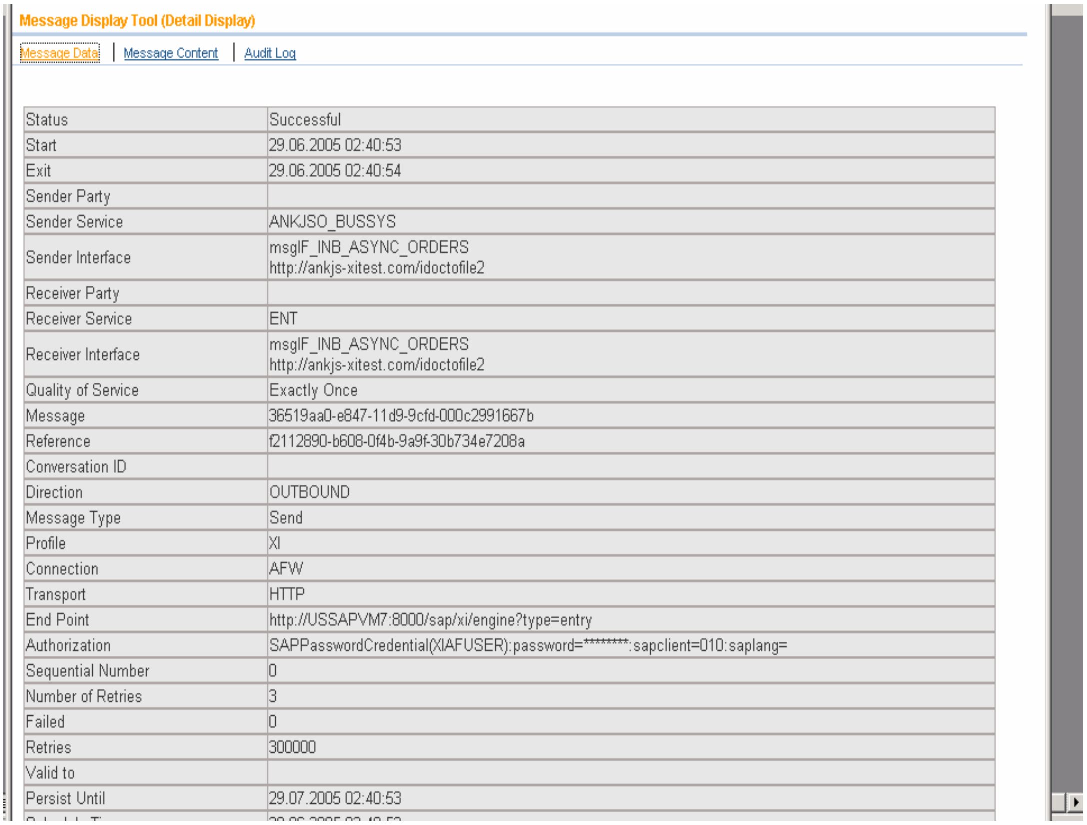
Time Stamp	Status	Description
2005-06-29 02:40:53	Success	Trying to put the message into the send queue.
2005-06-29 02:40:53	Success	Message successfully put into the queue.
2005-06-29 02:40:53	Success	The message was successfully retrieved from the send queue.
2005-06-29 02:40:53	Success	The message status set to DLNG.
2005-06-29 02:40:54	Success	The message was successfully transmitted to endpoint http://USSAPVM7:8000/sap/xi/engine?type=entry using connection AFW.
2005-06-29 02:40:54	Success	The message status set to DLVD.
2005-06-29 02:40:54	Success	The message was successfully delivered.

Total: 7 Entries 10 messages displayed per page; this is page 1 of 1 page(s)


Message Display Tool (Detail Display)
Message Data | Message Content | Audit Log

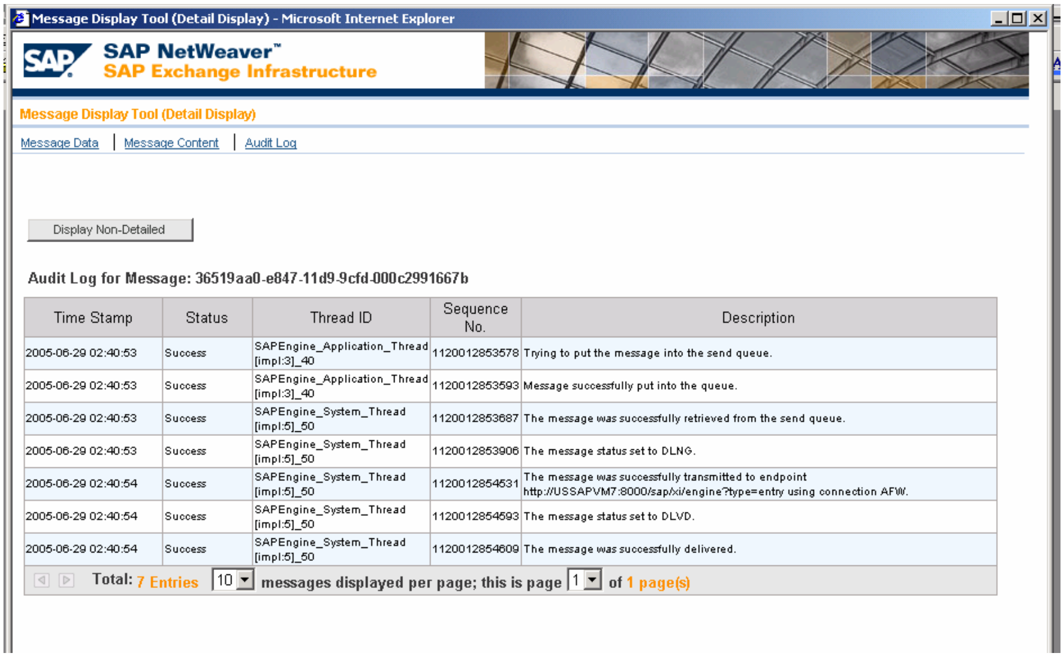
```
SOAPAction:"http://sap.com/xi/XI/Message/30"
Content-Type:text/xml; charset=utf-8

<SOAP:Envelope xmlns:SOAP='http://schemas.xmlsoap.org/soap/envelope/'><SOAP:Header><sap:Main xmlns:sap='http://sap
```

File data saved to the root directory



Message Display Tool (Detail Display)

Status	Successful
Start	29.06.2005 02:40:53
Exit	29.06.2005 02:40:54
Sender Party	
Sender Service	ANKJSO_BUSSYS
Sender Interface	msgIF_INB_ASYNC_ORDERS http://ankjs-xitest.com/idoctofile2
Receiver Party	
Receiver Service	ENT
Receiver Interface	msgIF_INB_ASYNC_ORDERS http://ankjs-xitest.com/idoctofile2
Quality of Service	Exactly Once
Message	36519aa0-e847-11d9-9cfd-000c2991667b
Reference	f2112890-b608-0f4b-9a9f-30b734e7208a
Conversation ID	
Direction	OUTBOUND
Message Type	Send
Profile	XI
Connection	AFW
Transport	HTTP
End Point	http://USSAPVM7:8000/sap/xi/engine?type=entry
Authorization	SAPPasswordCredential(XIAFUSER):password=********:sapclient=010:saplang=
Sequential Number	0
Number of Retries	3
Failed	0
Retries	300000
Valid to	
Persist Until	29.07.2005 02:40:53



Message Display Tool (Detail Display) - Microsoft Internet Explorer
SAP NetWeaver™
SAP Exchange Infrastructure

Message Display Tool (Detail Display)

Message Data
Message Content
Audit Log
Display Non-Detailed

Audit Log for Message: 36519aa0-e847-11d9-9cfd-000c2991667b

Time Stamp	Status	Thread ID	Sequence No.	Description
2005-06-29 02:40:53	Success	SAPEngine_Application_Thread [impl:3]_40	1120012853578	Trying to put the message into the send queue.
2005-06-29 02:40:53	Success	SAPEngine_Application_Thread [impl:3]_40	1120012853593	Message successfully put into the queue.
2005-06-29 02:40:53	Success	SAPEngine_System_Thread [impl:5]_50	1120012853687	The message was successfully retrieved from the send queue.
2005-06-29 02:40:53	Success	SAPEngine_System_Thread [impl:5]_50	1120012853906	The message status set to DLNG.
2005-06-29 02:40:54	Success	SAPEngine_System_Thread [impl:5]_50	1120012854531	The message was successfully transmitted to endpoint http://USSAPVM7:8000/sap/xi/engine?type=entry using connection AFW.
2005-06-29 02:40:54	Success	SAPEngine_System_Thread [impl:5]_50	1120012854593	The message status set to DLVD.
2005-06-29 02:40:54	Success	SAPEngine_System_Thread [impl:5]_50	1120012854609	The message was successfully delivered.

Total: 7 Entries 10 messages displayed per page; this is page 1 of 1 page(s)

Author Bio

With more than 13 years industry experience, Srinivas Vankamamidi has a good understanding of most of the business processes, configuration exposure, and experience developing business applications using various techniques of ABAP.

He has worked with a number of clients and handled various full cycle implementations, upgrades, critical short-term projects, production support functions, sizing and hardware plausibility checks for SAP R/3 upgrades, along with stress test analysis, performance management, EDI and ALE projects, business connectors, various integration projects, third-party application developments, ITS with huge vendor and customer integrations projects, and more.

He was involved in rolling out industry-specific needs like RRB solution development for A&D, MRP-related developments and optimization for midsize to large-sized engineering industries, Vertex, Jetforms, LIMS, Hyperion integration projects with SAP R/3, transportation management and integration with I2, specific module design with team of expertise involving warranty management for service divisions adds to his overall experience.