

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.

Applies To:

The second part of this XI beginners guide continues walking through settings, configurations, mappings, etc. I refer you to "[A Beginner's Guide to SAP XI Settings, Part I](#)" to understand few fundamentals, SAP R/3 settings, XI settings, etc. It is designed as a learning journey to help all aspiring XI experts and it will also be useful to some Java experts who want to learn a few SAP R/3 transactions involving connectivity establishments, configurations, partner profiles, transactions and port declarations, IDoc postings and generations, etc. All intermediate and advanced users may want to ignore this but I will surely make an attempt to post few more good articles/scenarios on advanced topics involving BPM, PROXIES, and more.

Summary

As part of interface development with any business system integration, we encounter the necessity of creating master data/transaction data in the SAP R/3 environment, or visa versa, using FILE/IDoc/RFC calls. This exercise is designed to take such inputs and post master data into the SAP R/3 environment.

By: Srinivas Vankamamidi

Company: Intelligroup Inc.

Date: 31 Oct 2005

Table of Contents

Applies To:.....	2
Summary	2
Table of Contents	2
Scenario	3
Business Scenario: Creating a Customer Master Data in SAP R/3 from a Legacy Source.....	3
XI Settings.....	4
Scenario Design	5
TIP 1	8
TIP 2	9
TIP 3	9
TIP 4.....	10
TIP 6.....	11
TIP 7.....	11
Exercises	12
Author Bio.....	15

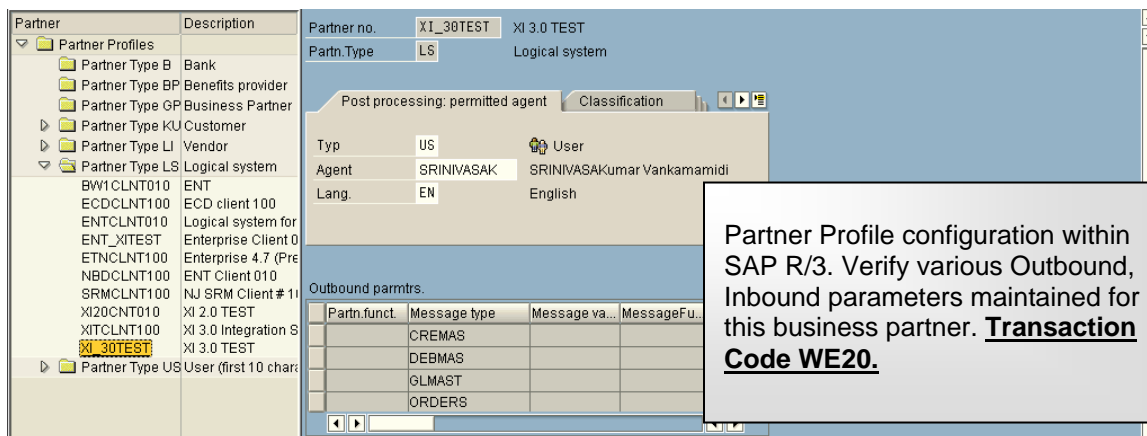
Scenario

Business Scenario: Creating a Customer Master Data in SAP R/3 from a Legacy Source

Note: Please refer to “[A Beginner's Guide to SAP XI Settings, Part I](#)” to understand various components of XI.

I. SAP R/3 Settings requirements:

Partner Profiles: Use the Transaction WE20 to maintain the partners. Their “TYPE” identifies these Partners and respective FUNCTIONS. For instance, Partner Type LS represents LOGICAL SYSTEM and Type LI represents SUPPLIERS/VENDORS, Type KU represents CUSTOMERS/DEBTORS and etc., for our scenario purpose we need to create a Partner with Type LS. Before doing this we need to create a Logical System R/3 using transaction BD54 or SALE. Once after this is done, you will then create a logical partner using the transaction WE20 and assign with certain Outbound and/or inbound parameters as needed.



Our exercise is to post a Master IDoc into R/3, and hence we need to maintain the inbound parameters.

Please use the message type DEBMAS.

Use the Process code DEBM or if you have any created Z* in your system for posting the customer master, use it here.

Partner Type: Determines which partner Master data the interface reads.

Partner Function: Identifies the business partner function like SD Players, Procurement players etc.,

Message Type: Normally assigned to the SAP Documents and are unique.

Process Code: Interface determines the Code to be used to convert the IDoc into SAP Application document.

XI Settings

1. Refer to the “[A Beginner's Guide to XI Settings, Part I](#)” to understand the settings that are needed to carry out this scenario successfully.

One point to remind you that, the type of R/3 connection we are going to choose for our purpose is of type 3 – R/3 Connection. Refer to the screen shot for additional information.

Click this push button to test the connection.

Provide LOG-ON data with Client, User & P/w.

RFC Destination definition with source IP, system number

Make sure you test the connectivity.

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

2. Refer to the "XI Beginners Guide" once again to understand the necessity of maintaining the PORTS for communicating to the receiver/senders from and to XI. If this not maintained, we get the errors like "No Recipient Exists"/"No Sender Identified," etc., and it 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.

Above two are important settings and are required to be ensured and tested before we design our scenario in XI.

Scenario Design

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

1. Start XI and type/access SXMB_IFR to launch your integration builder or type "http://<server>:<port>/rep in the Internet explorer address bar, then hit enter. This will launch the 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. OR Use the same components, which we have created while reading the ORDERS data from R/3 as seen in "XI Beginner's Guide".

3. Click on Integration Repository, import all your SLD (system landscape directory) data from ENVIRONMENT menu option.

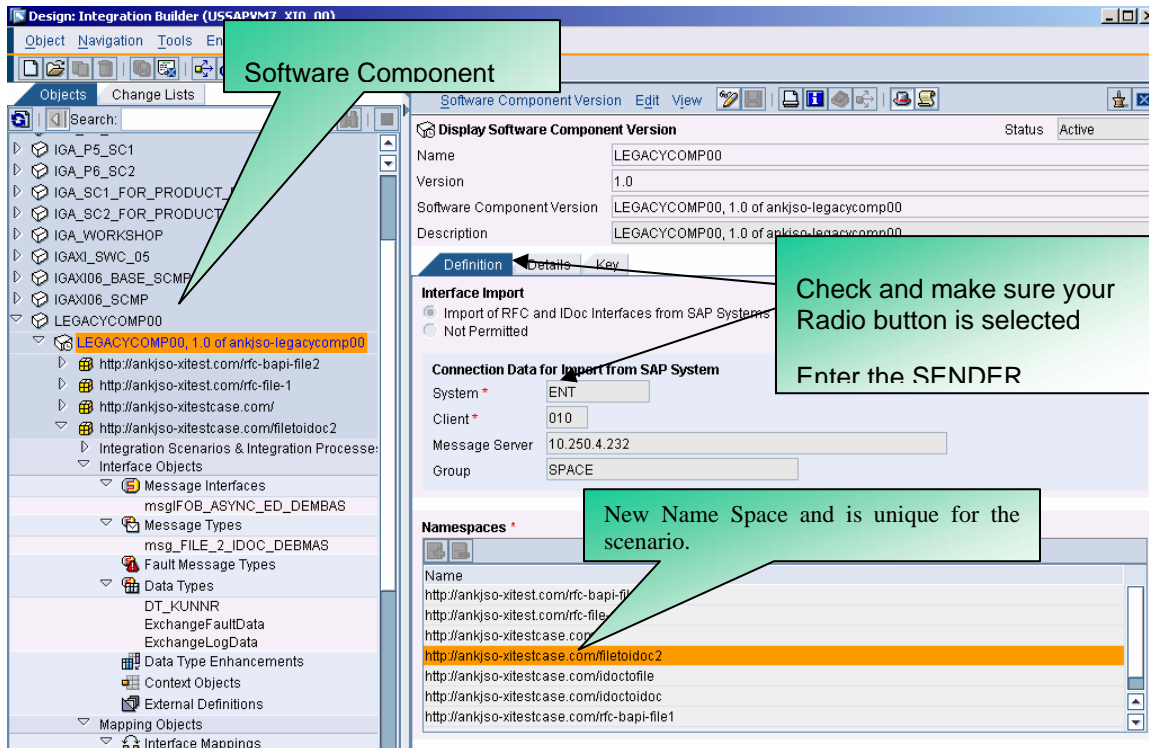
Note: Refer to Part I to get quick insight and tips for a better understanding.

4. So, we have our SWC and corresponding version available for us to begin with. Double click on SWCV then bring it the Edit mode. Now as specified in TIP 2.1, enter the required info after switching to "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 " + " 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. Enter something meaningful like "http://xitestcase.com/debmas 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.

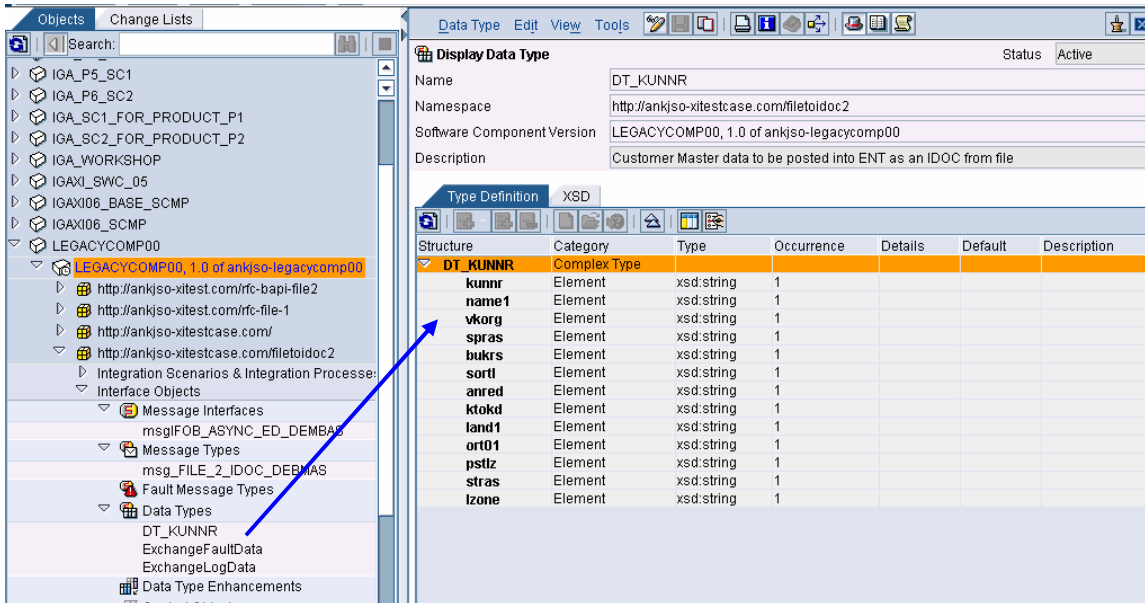
7. Expand the Name Space by clicking on Triangle pointing rightwards.



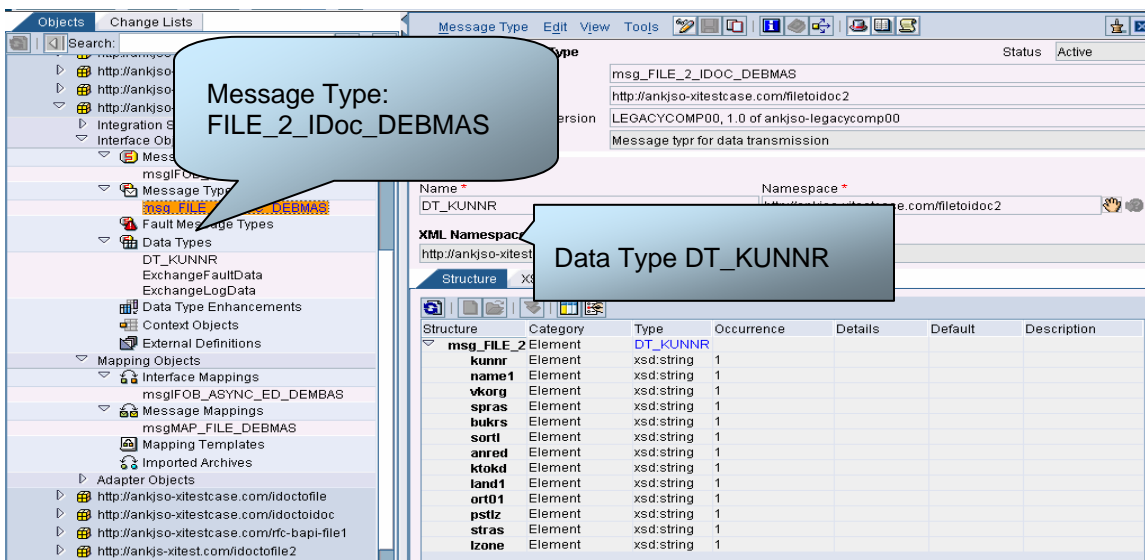
8. Create the Data Type and build the structure into which the data from R/3 IDoc shall be received. After expanding the NAME SPACE, expand the INTERFACE OBJECTS. Then right click on DATA TYPES to start creating.

For this purpose we define two data types DT_KUNNR and we need not create Receiver Data type, as it is an IDoc.

Refer to the following screen shot for additional information.

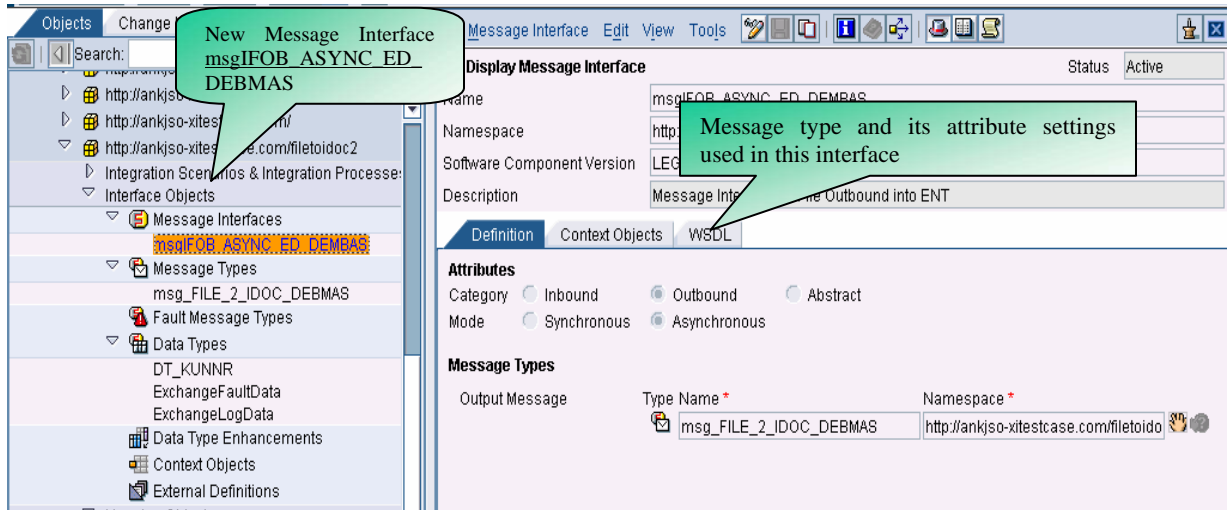


9. Create a Message type and assign the data type DT_KUNNR to it. Make sure you drag and drop the data type on the OPEN HAND icon shown on the right hand side next to NAME SPACE box. Or, you can drop down the list and select the one, which you have created.



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 and outbound (into R/3 from XI).

Drag and Drop the FILE_2_IDoc_DEBMAS created in the previous step 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 resulting window. Or simply type the details.



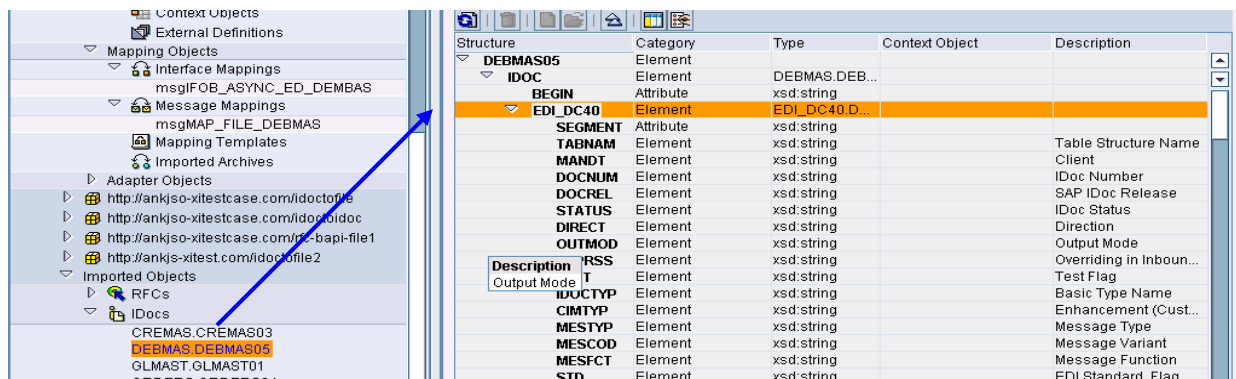
TIP 1

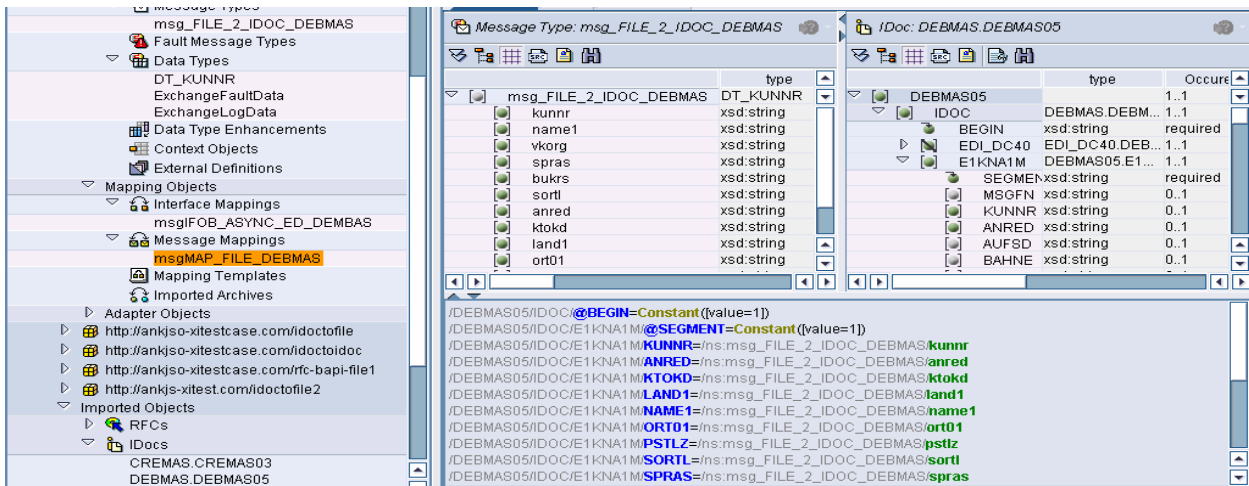
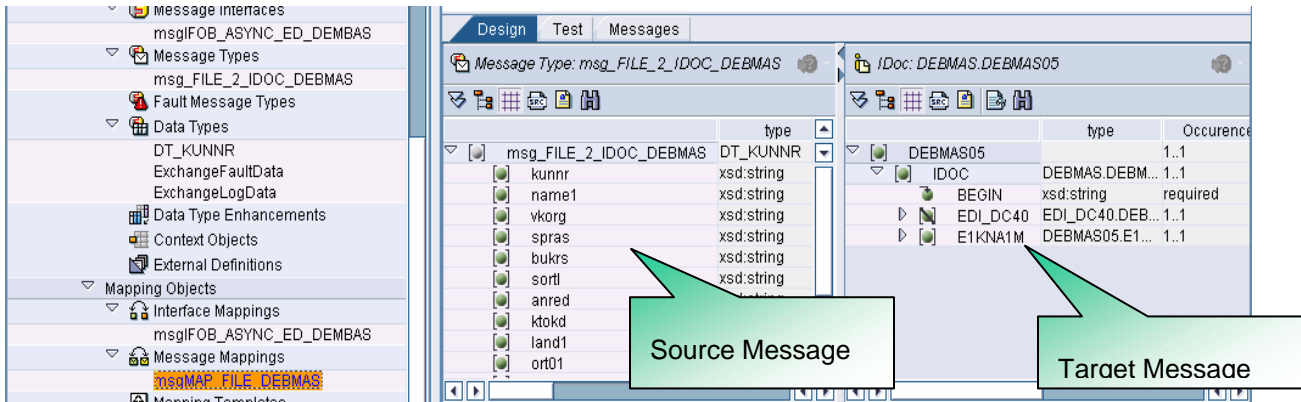
IDoc is a Standard Object and we need not create the Receiver data type, message type for handling in our scenario. While designing our SWCV, SWC, we have seen the Usage Dependencies. This will enable us to use the **IDoctype.Message type** while Message mapping. Refer to the XI Beginners Guide for details. Also, the screen shot shown under Message mapping.

11. At this stage we are done with definitions and now its time to map the messages and interface created thus far. Do the Consistency Check (F7) before going further, if needed.

To begin with, click on the triangle pointing rightwards on MAPPING OBJECTS. Right click on MESSAGE MAPPING and create a new map "msgMAP_FILE_DEBMAS". This will result a graphical window where in you drag and drop the source (msgFILE_2_IDoc_DEBMAS) and the target message DEBMAS.DEBMAS05 from imported Objects. Refer to the screen shot shown below.

This is needed to map the SOURCE field to the TARGET field.





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

TIP 2

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.

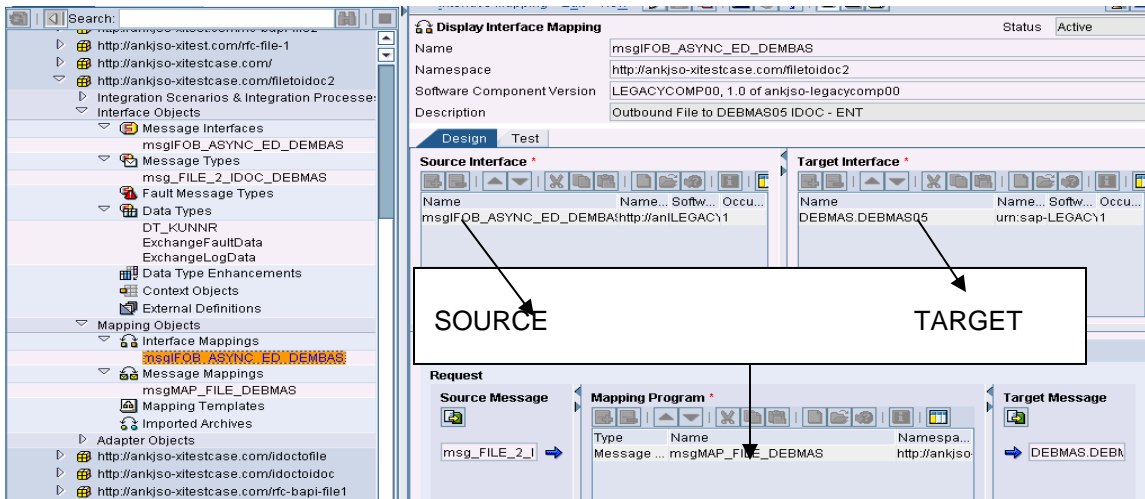
TIP 3

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

12. Now, the last step in the Integration Repository is to create the Interface Mapping. In this step map the Source (msgIFOB_ASYNC_ED_DEBMAS) to the Target Interface (DEBMAS.DEBMAS.05). Once the objects are identified and filled into respective fields, click on "Read Interfaces" push button on the right hand side in the bottom portion of the work area. This replicates the Source and Targets into Request Tab Window. Identify the Message mapping program created in the above step and assign here.

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

At the end, save the work, activate the Change List.

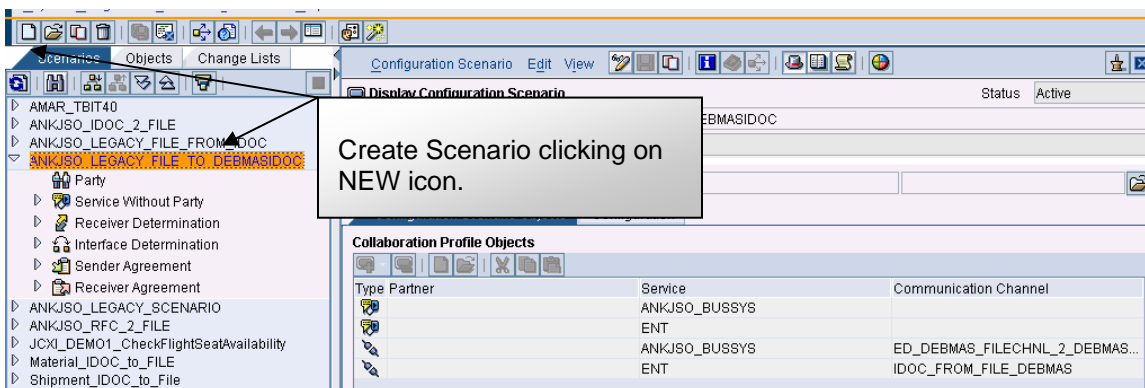


TIP 4

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 (will be more clear if I say Receiver and Sender Agreement). But in our case, the Idea is to Send the IDoc into R/3 and hence, we don't need a sender configuration and hence no senders agreement.

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.



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 use the graphical editor, as you will not know what is happening in the background.

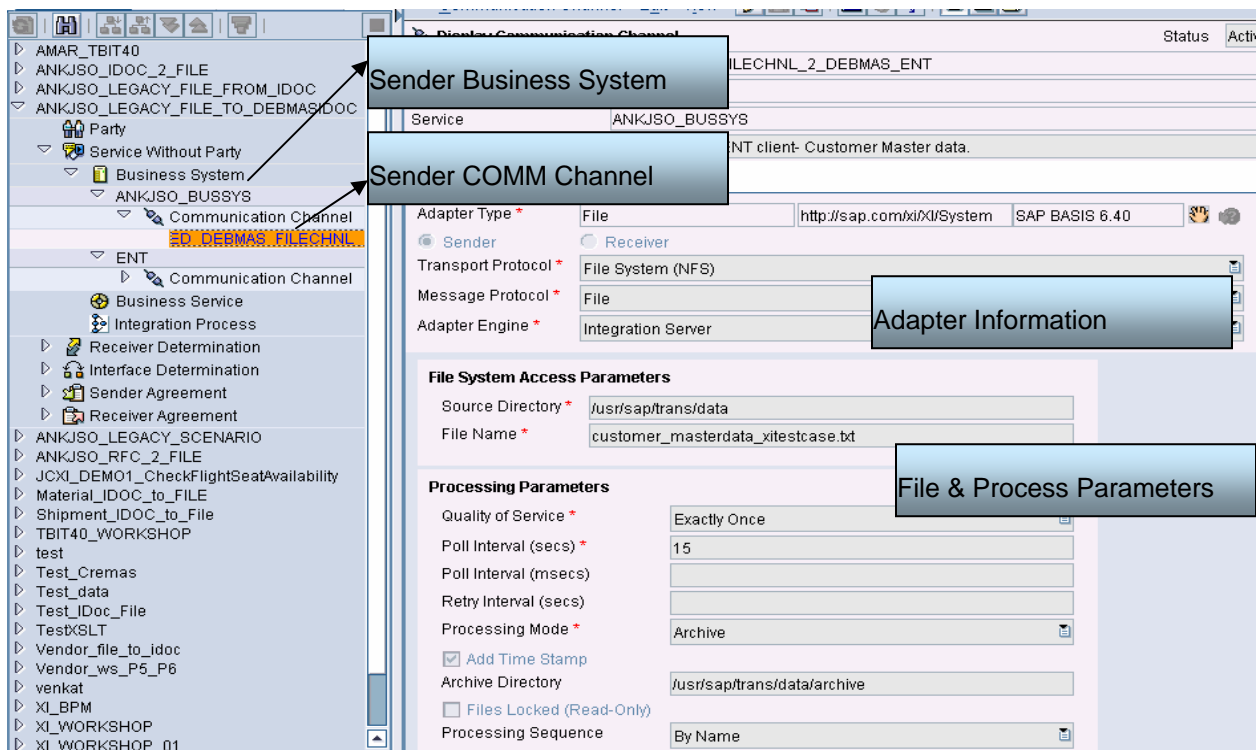
TIP 5

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 6

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 Sender System assignment is needed at this stage. For the receiver Business system, we now create the communication channel. We know that we are mapping the File to IDoc, 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).



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 process the file. Don't forget to choose the SENDER radio button as FILE WILL BE SENT by this channel when executed.

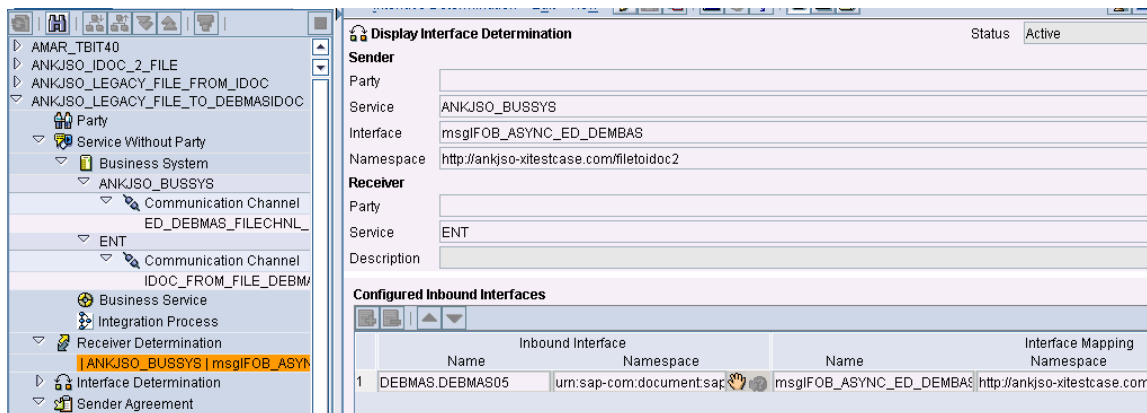
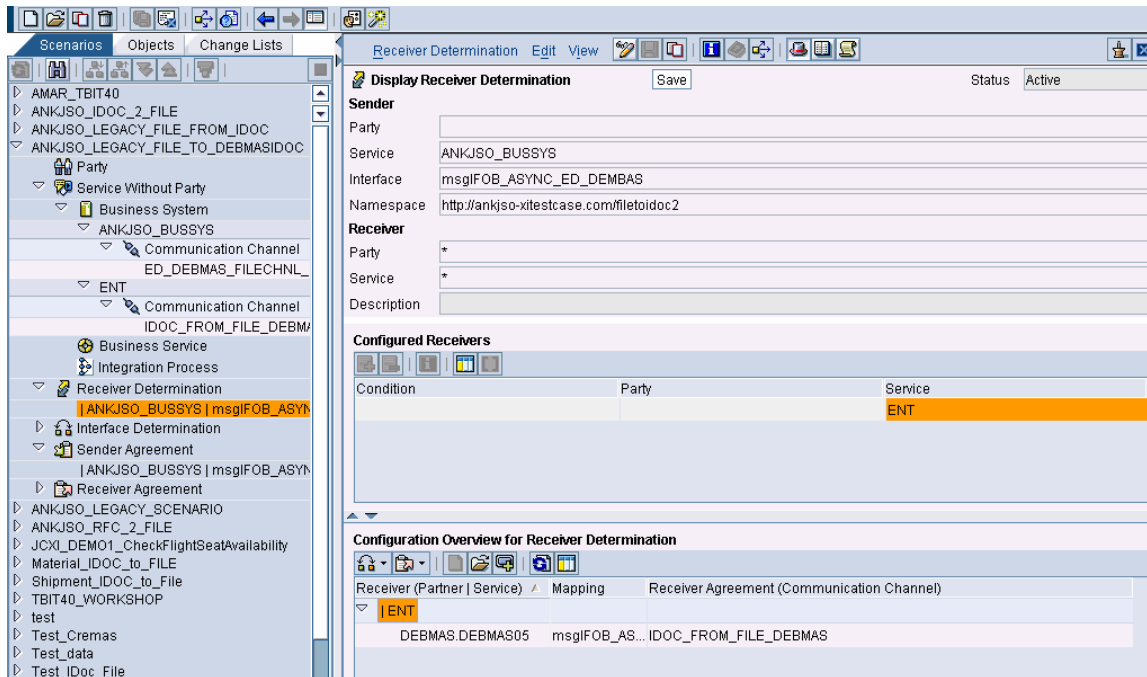
TIP 7

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 Business System (R/3) and corresponding Communication channel (IDoc).

19. Create the Sender and Receiver Agreements and refer to the Beginners Guide Part 1 for all information on this part of XI Settings and configuration.

20. Create the RECEIVER DETERMINATION.



Save all changes. Activate the Changes after checking the Consistency (F7).

Exercises

1. List the Configuration steps followed in the Integration Repository
2. List the sequence of steps followed in Integration directory.
3. List the types of Adapters
4. List the Adapters we used in our scenario and describe
5. Where does the IDoc Comm Channel reside?
6. Where does the FILE Comm channel reside?

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



Our test file:

```

First page Previous page Next page Last page
Directory: E:\usr\sap\trans\data
Name: customer_masterdata_xitestcase.txt

<?xml version="1.0" encoding="UTF-8"?>
<ns:msg_FILE_2_IDOC_DEBMAS xmlns:ns="http://ankjso-xitestcase.com/filetoidoc2">
  <kunnr>A10204</kunnr>
  <name1>XI TEST CASE-changes test</name1>
  <vkorg>USA</vkorg>
  <spras>EN</spras>
  <bukrs>US01</bukrs>
  <sort1>XI TEST CASE - Srini</sort1>
  <anred>COMPANY</anred>
  <ktokd>0001</ktokd>
  <lnd1>US</lnd1>
  <ort01>EDISON</ort01>
  <pstlz>08837</pstlz>
  <stras>499 Thornall Street</stras>
  <lzone>0000000001</lzone>
</ns:msg_FILE_2_IDOC_DEBMAS>

```

Execute the scenario and look in the RWB for monitoring and results verification.

Start Reset Show Additional Criteria Messages from Archive

Status: All From (Date) 7/25/2005 From (Time) 21:40:23 To (Date) 7/26/2005

Sender Party Service Receiver Party Service Interface Interface Namespace

Configure Table Columns

Messages from Database

Update Details Referencing Messages Referenced Message Repeat Cancel Error Log

Select	End-End Information	Display Message Details Status	Start	Exit	Sender Party	Sender Service	Sender Interface
<input checked="" type="checkbox"/>		Successful	25.07.2005 22:39:30	25.07.2005 22:39:30		ANKJSO_BUSSYS	msgIFOB_ASYNC_ED_DEMBAS http://ankjso-xitestcase.com/filetoidoc
<input type="checkbox"/>		Successful	25.07.2005 22:16:00	25.07.2005 22:16:00		ANKJSO_BUSSYS	msgIFOB_ASYNC_ED_DEMBAS http://ankjso-xitestcase.com/filetoidoc

Total: 2 Messages 10 messages displayed per page; this is page 1 of 1 page(s)

Display Detailed

Audit Log for Message: f6963d00-fd5c-11d9-af27-000c2991667b

Time Stamp	Status	Description
2005-07-25 22:39:30	Success	Channel ED_DEBMAS_FILECHNL_2_DEBMAS_ENT: Send binary file "E:\usr\sap\trans\data\customer_masterdata_xitestcase.txt". Size 513 with QoS ED
2005-07-25 22:39:30	Success	Application attempting to send an XI message asynchronously using connection AFW.
2005-07-25 22:39:30	Success	Trying to put the message into the send queue.
2005-07-25 22:39:30	Success	Message successfully put into the queue.
2005-07-25 22:39:30	Success	The application sent the message asynchronously using connection AFW. Returning to application.
2005-07-25 22:39:30	Success	The message was successfully retrieved from the send queue.
2005-07-25 22:39:30	Success	File "E:\usr\sap\trans\data\customer_masterdata_xitestcase.txt" archived after processing
2005-07-25 22:39:30	Success	The message status set to DLNO.
2005-07-25 22:39:30	Success	The message was successfully transmitted to endpoint http://USSAPVM7:8000/sap/xi/engine?type=entry using connection AFW.
2005-07-25 22:39:30	Success	The message status set to DLVD.

Total: 11 Entries 10 messages displayed per page; this is page 1 of 2 page(s)

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

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

SOAPAction:"http://sap.com/xi/XI/Message/30"
Content-Type:Multipart/Related; boundary=SAP_Self3671-fd5d-11d9-a895-000c2991667b

--SAP_Self3671-fd5d-11d9-a895-000c2991667b_END
content-type: text/xml; charset=utf-8
content-length: 1702
content-id: <soap-Self3670fd5d11d9a56d000c2991667b8sap.com>
content-disposition: attachment; filename="soap-Self3670fd5d11d9a56d000c2991667b8s

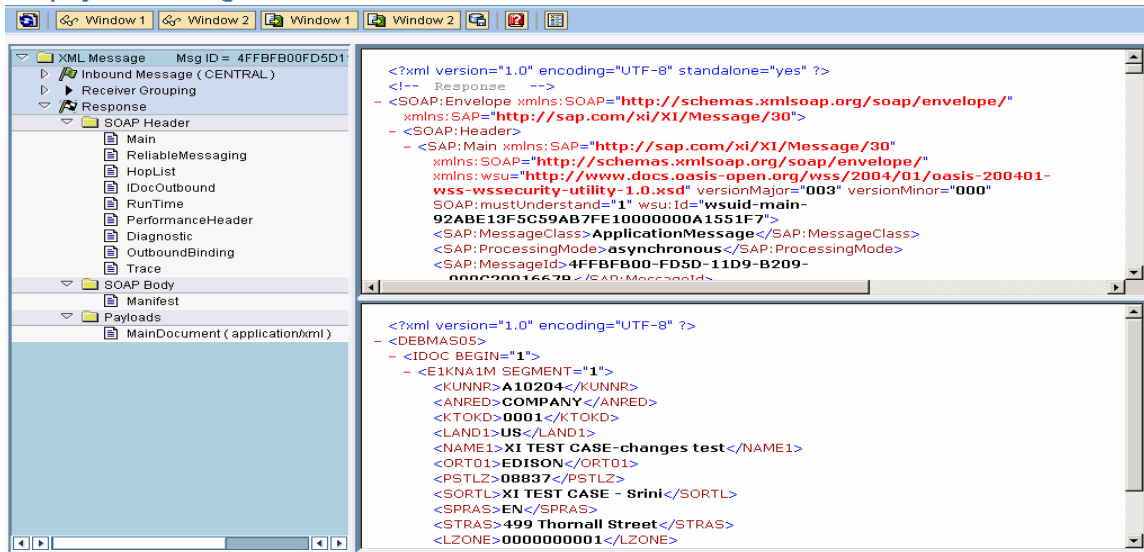
<SOAP:Envelope xmlns:SOAP='http://schemas.xmlsoap.org/soap/envelope/'><SOAP:Header
--SAP_Self3671-fd5d-11d9-a895-000c2991667b_END
content-type: application/xml
content-length: 513
content-description: MainDocument
content-id: <payload-502220a1fd5d11d9c778000c2991667b8sap.com>
content-disposition: attachment; filename="payload-502220a1fd5d11d9c778000c2991667

<?xml version="1.0" encoding="UTF-8"?>

<ns:msg_FILE_2_IDOC_DEBMAS xmlns:ns="http://ankjso-xitestcase.com/filetoidoc2">
  <KUNNR>A10204</KUNNR>
  <NAME1>XI TEST CASE-changes test</NAME1>
  <VKORG>USA</VKORG>
  <SPRAS>EN</SPRAS>
  <BUKRS>US01</BUKRS>
  <SORTL>XI TEST CASE - Srini</SORTL>
  <ANRED>COMPANY</ANRED>
  <KTOKD>0001</KTOKD>
  <LAND1>US</LAND1>
  <ORT01>EDISON</ORT01>
  <PSTLZ>08837</PSTLZ>
  <STRAS>499 Thornall Street</STRAS>
  <LZONE>000000001</LZONE>
</ns:msg_FILE_2_IDOC_DEBMAS>
```

XML Payload details:

Display XML Message Versions



The screenshot shows the SAP Message Display Tool interface. On the left, a tree view displays the message structure: XML Message (Msg ID = 4FFBF00FD5D1) -> Inbound Message (CENTRAL) -> Receiver Grouping -> Response -> SOAP Header -> Main -> ReliableMessaging -> HopList -> IDocOutbound -> RunTime -> PerformanceHeader -> Diagnostic -> OutboundBinding -> Trace -> SOAP Body -> Manifest -> Payloads -> MainDocument (application/xml).

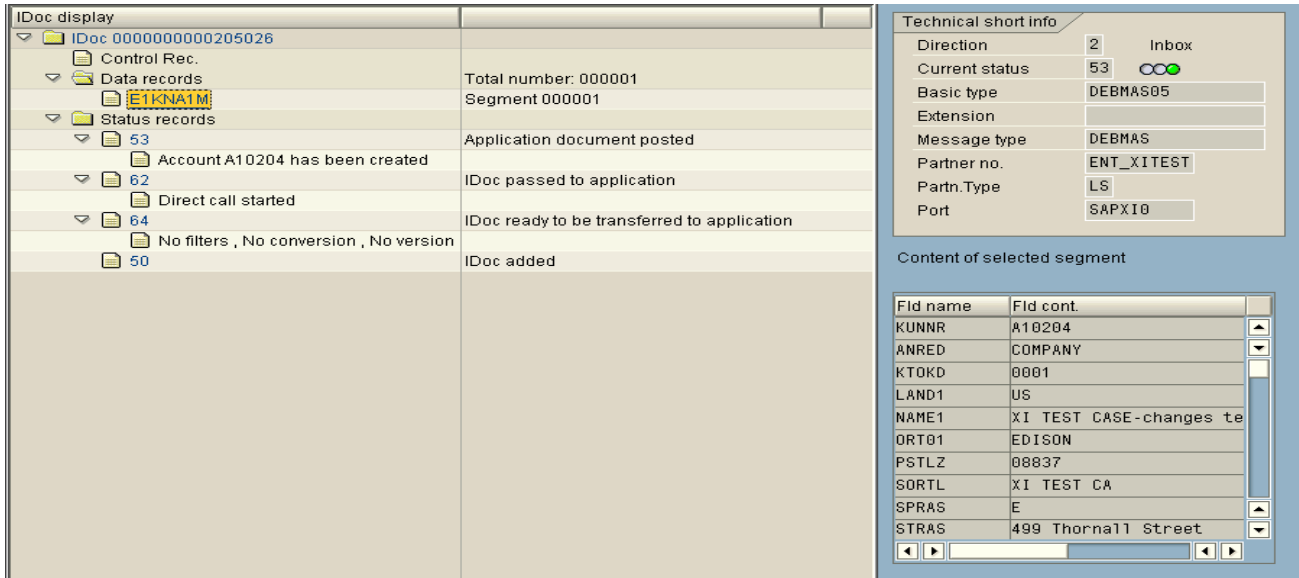
The main display area shows the XML payload details for the SOAP Header and the MainDocument. The SOAP Header XML is as follows:

```
<?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.w3.org/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd" versionMajor="003" versionMinor="000"
SOAP:mustUnderstand="1" wsu:Id="wsuid-main-
92ABE13F5C59AB7FE10000000A1551F7">
      <SAP:MessageClass>ApplicationMessage</SAP:MessageClass>
      <SAP:ProcessingMode>asynchronous</SAP:ProcessingMode>
      <SAP:MessageId>4FFBF00-FD5D-11D9-B209-
000c2991667b</SAP:MessageId>
```

The MainDocument XML is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<DEBMAS05>
  <IDOC BEGIN="1">
    <EIKNAIM SEGMENT="1">
      <KUNNR>A10204</KUNNR>
      <ANRED>COMPANY</ANRED>
      <KTOKD>0001</KTOKD>
      <LAND1>US</LAND1>
      <NAME1>XI TEST CASE-changes test</NAME1>
      <ORT01>EDISON</ORT01>
      <PSTLZ>08837</PSTLZ>
      <SORTL>XI TEST CASE - Srini</SORTL>
      <SPRAS>EN</SPRAS>
      <STRAS>499 Thornall Street</STRAS>
      <LZONE>000000001</LZONE>
```

In R/3:



The screenshot shows the SAP XI IDoc display interface. On the left, a tree view shows the IDoc structure for IDoc 0000000000205026, including Control Rec., Data records, and Status records. The Status records section is expanded, showing a sequence of events: 'Account A10204 has been created', 'Direct call started', 'IDoc ready to be transferred to application', and 'IDoc added'. On the right, the 'Technical short info' panel displays details such as Direction (2), Current status (53), Basic type (DEBMAS05), and Partner no. (ENT_XITEST). Below this, the 'Content of selected segment' table lists fields like KUNNR, ANRED, and STRAS with their corresponding values.

Fld name	Fld cont.
KUNNR	A10204
ANRED	COMPANY
KTOKD	0001
LAND1	US
NAME1	XI TEST CASE-changes te
ORT01	EDISON
PSTLZ	08837
SORTL	XI TEST CA
SPRAS	E
STRAS	499 Thornall Street

Display the customer in R/3:



The screenshot shows the SAP R/3 customer display interface. The 'Name' section displays the company name 'XI TEST CASE-changes test'. The 'Search terms' section shows the search term 'XI TEST CA'. The 'Street address' section displays the address '499 Thornall Street', postal code '08837', city 'EDISON', country 'US', and time zone 'CST'.

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.