Deepak Shah
September 19, 2011 8 minute read

Exchanging IDocs with Non-SAP system over TCP/IP using PI as Middleware

3 14 11,131

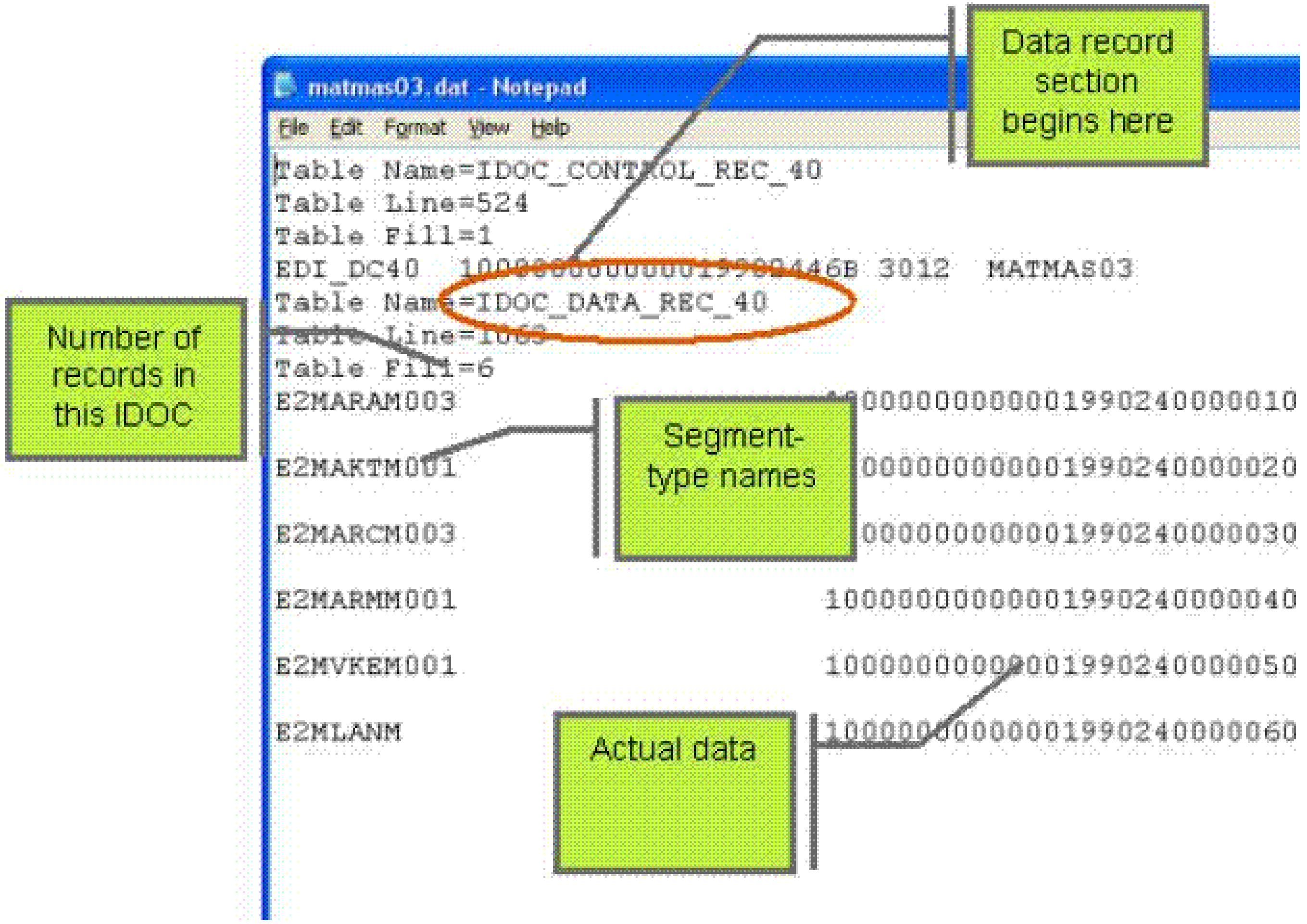
This blog outlines steps required for exchanging IDocs with non-SAP systems over TCP/IP using PI as Middleware. This blog will cover following scenarios.

A non SAP system sends data using SOAP protocol to PI, which in turns send it to a non SAP system in form of IDocs.
A non sap system sends IDocs to PI system over TCP/IP protocol. PI in turn sends it to another non SAP system over SOAP protocol.

There are several non SAP systems which are capable of sending/Receiving IDocs over TCP/IP protocol.
Here we had to exchange IDocs with non-SAP system SQL*LIMS which is capable of exchanging IDocs over TCP/IP. LIMS uses IDoc Interface to connect to any external system.

Here we had created Z-Idocs as per the requirements. I will use SOAP to IDoc & IDoc to SOAP scenarios. I will explains only the necessary steps for exchanging Idocs with non SAP system over TCP/IP and will skip steps like data types, mappings etc.

This is how IDOCs look like:

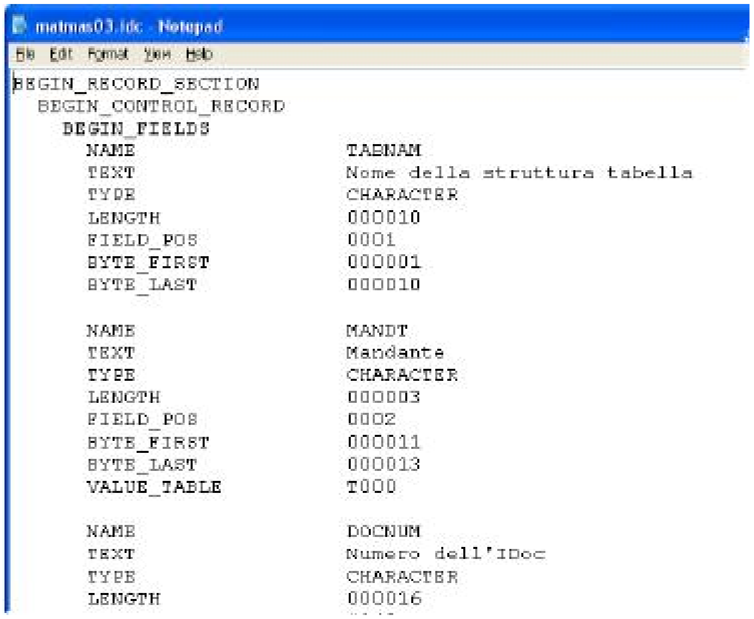

matmas03.dat - Notepad
File Edit Format View Help
Data record section begins here
Number of records in this IDOC
Segment-type names
Actual data

```
Table Name=IDOC_CONTROL_REC_40
Table Line=524
Table Fill=1
EDI_DC40  1000000000000019900446B 3012  MATMAS03
Table Name=IDOC_DATA_REC_40
Table Line=1063
Table Fill=6
E2MARAM003                      000000000000019902400000010

E2MAKTM001                      000000000000019902400000020

E2MARCM003                      000000000000019902400000030

E2MARMM001                    100000000000001990240000040

E2MVKEM001                    100000000000001990240000050

E2MLANM                       100000000000001990240000060
```

The control record section of an IDOC contains routing information. There might be multiple SAP systems using the same SAP gateway. The IDOC control record can be used to address a specific SAP system. Furthermore, it identifies the sender system, allowing the target SAP instance to activate the correct logic subsystem to process the inbound message.

The data record defines the structure to hold the actual information to exchange. Data records are organized on a per-line basis: each line contains a short header plus *segments* and *fields*. The short header identifies the record-type for the current line. Then there is the segment, which contains fields. The same IDOC can contain multiple data records, each one containing its own specific segment with a specific segment type.

SAP is capable of producing a text file with the description of a specific IDOC type. This file is called “**Parser File**”. External applications can process parser files in order adapt to the structure of the relating IDOC types and this is usually done during the initial configuration process of the interfaces.
This is how a parser file looks like:



```
BEGIN_RECORD_SECTION
  BEGIN_CONTROL_RECORD
    BEGIN_FIELDS
      NAME                    TABNAM
      TEXT                    Nome della struttura tabella
      TYPE                    CHARACTER
      LENGTH                  000010
      FIELD_POS               0001
      BYTE_FIRST              000001
      BYTE_LAST               000010

      NAME                    MANDT
      TEXT                    Mandante
      TYPE                    CHARACTER
      LENGTH                  000003
      FIELD_POS               0002
      BYTE_FIRST              000011
      BYTE_LAST               000013
      VALUE_TABLE             T000

      NAME                    DOCNUM
      TEXT                    Numero dell'IDoc
      TYPE                    CHARACTER
      LENGTH                  000016
```

For a standard IDOC, we can go into WE60, give the basic type and press F9 to get the parser file.
For a custom IDOC we should release all the segments of the IDOC before we can create a parser file.

LIMS IDoc interface includes a module that can parse the SAP Parser Files. Using the SAP Parser Files, it will be possible to describe the content of a specific IDOC type (be it standard or custom, basic or extended) to the SQL*LIMS. Specific tables will be created in order to store that specific IDOC type, enabling the developer to fetch the content of specific fields in the IDOC by means of classical “select” SQL statements.

A – Sending the IDOC to the LIMS:
Here a third party non SAP system will send data via SOAP to SAP PI. PI will convert it into IDoc xml and send the IDoc to LIMS via TCP/IP protocol.
LIMS inbound IDOCs will be received by means of a background Windows 32bit service, the **IdocReceiver** service.

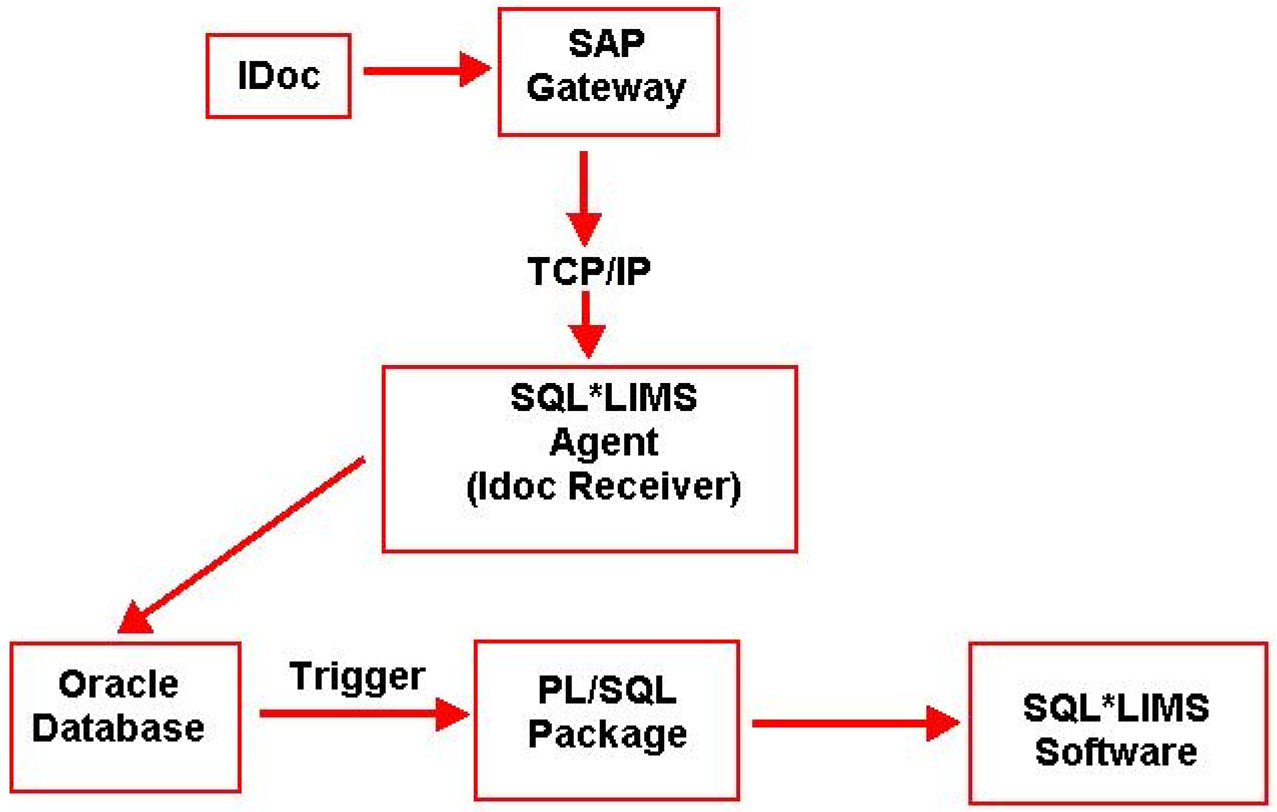
The **IdocReceiver** service, which is part of the Transport Layer, performs the following actions:

Listens for LIMS inbound IDOCs
Creates new inbound requests in the transactions table with status “DOWNLOADING”
Saves the downloaded IDOC as a plain-text clob attachment to the transaction
Changes the status of the request to “RECEIVED”, in order to pass it to the IdocAgent service for processing.

Error messages generated by the IdocReceiver service will be saved in the service’s log file.

IDoc received by LIMS are stored into the SQL*LIMS Oracle database in a way that its content can be easily extracted by means of SQL “select” statements

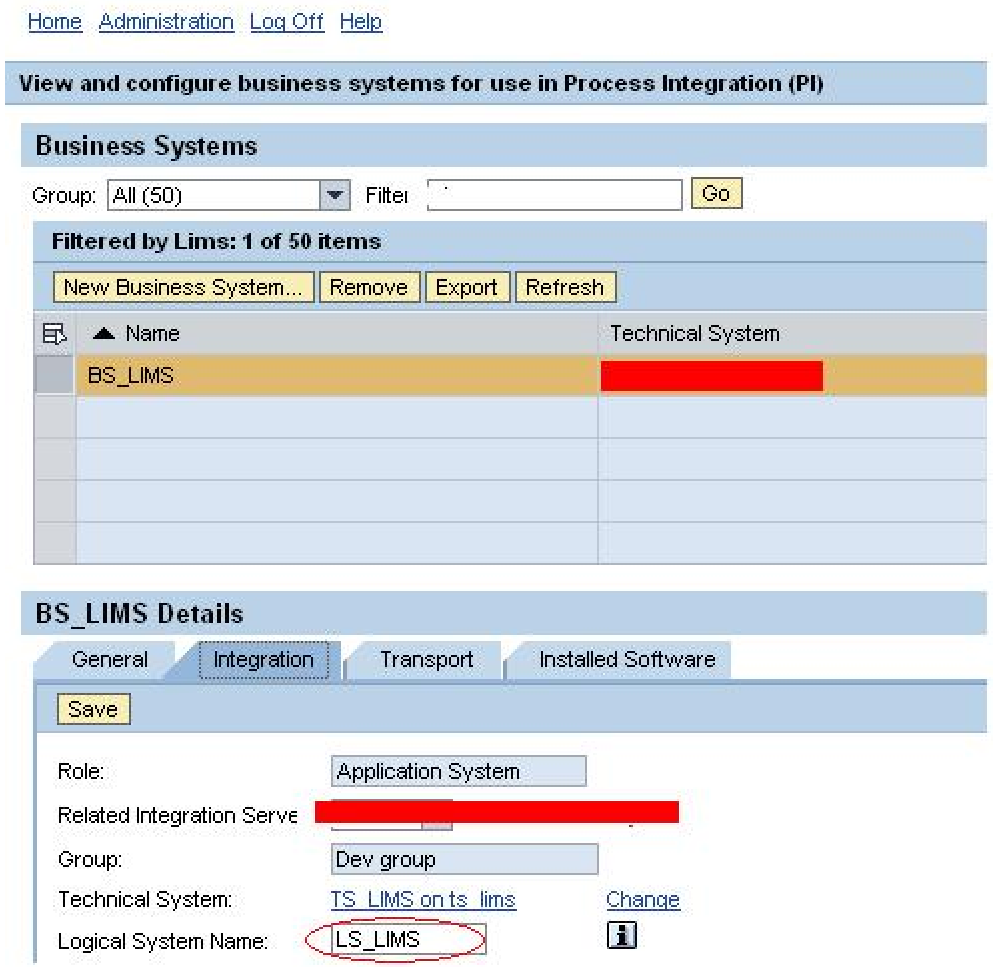
The wrapper package “**ABI_IDOC_SL_WRAPPER**” is used to perform the requested actions into the SQL*LIMS application.



Following are the important configurations steps that are necessary for sending IDocs to LIMS system.

1. Create the Business System for the Non SAP system in SLD (System Landscape Directory) with the logical system name mentioned in the Business System.

Note – The logical system name should be other than SAPXXX. Here in our example the logical system name of the Business system is **LS_LIMS** as shown below.



Home Administration Log Off Help

View and configure business systems for use in Process Integration (PI)

Business Systems

Group: All (50) Filter Go

Filtered by Lims: 1 of 50 items

New Business System... Remove Export Refresh

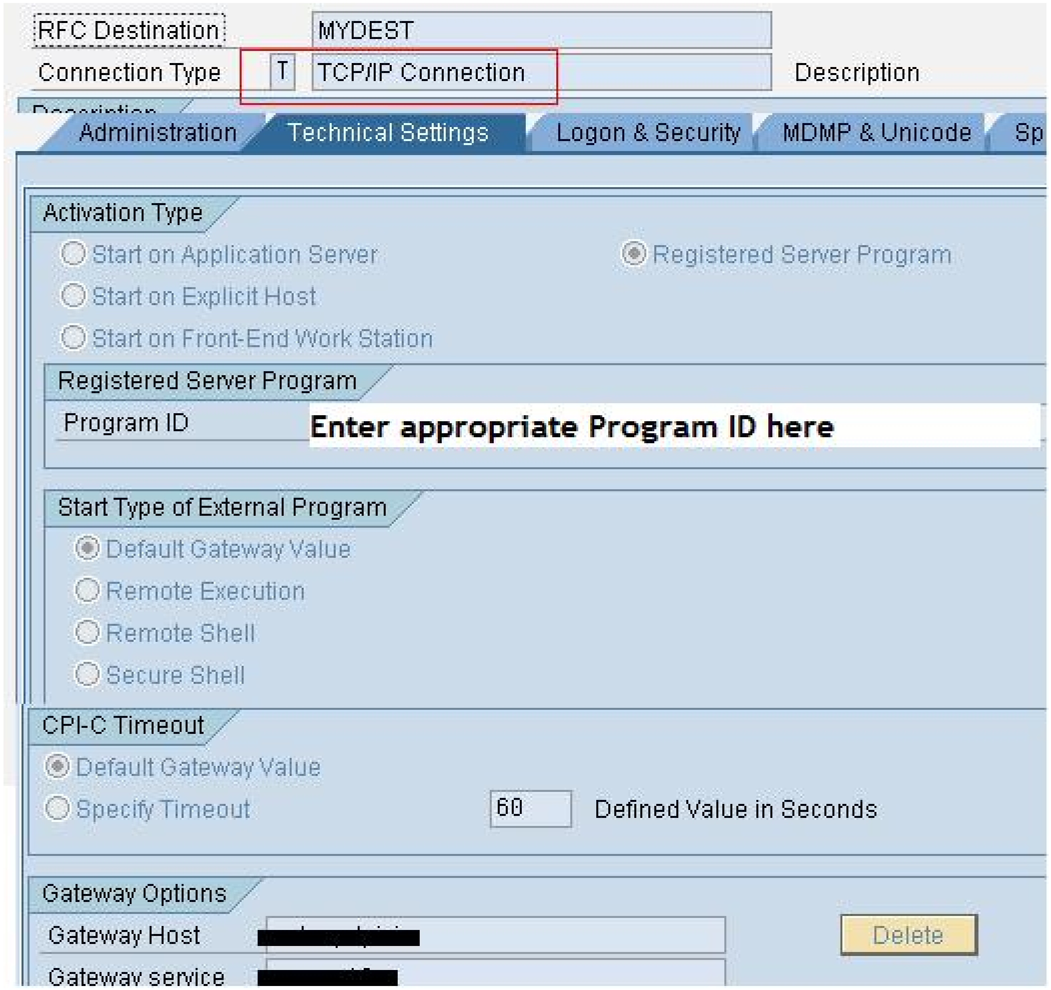
Name	Technical System
BS_LIMS	[redacted]

BS_LIMS Details

General Integration Transport Installed Software
Save

Role:	Application System
Related Integration Serve	[redacted]
Group:	Dev group
Technical System:	TS_LIMS on ts_lims Change
Logical System Name:	LS_LIMS

2. Create a TCP/IP connection (MYDEST) of type T in transaction SM59 in the PI system. The program ID should be register on SAP PI system. Take help of your basis team to register the program ID on PI server



RFC Destination: MYDEST
Connection Type: T TCP/IP Connection
Description

Activation Type
- Start on Application Server
- Start on Explicit Host
- Start on Front-End Work Station
- [✓ Registered Server Program]

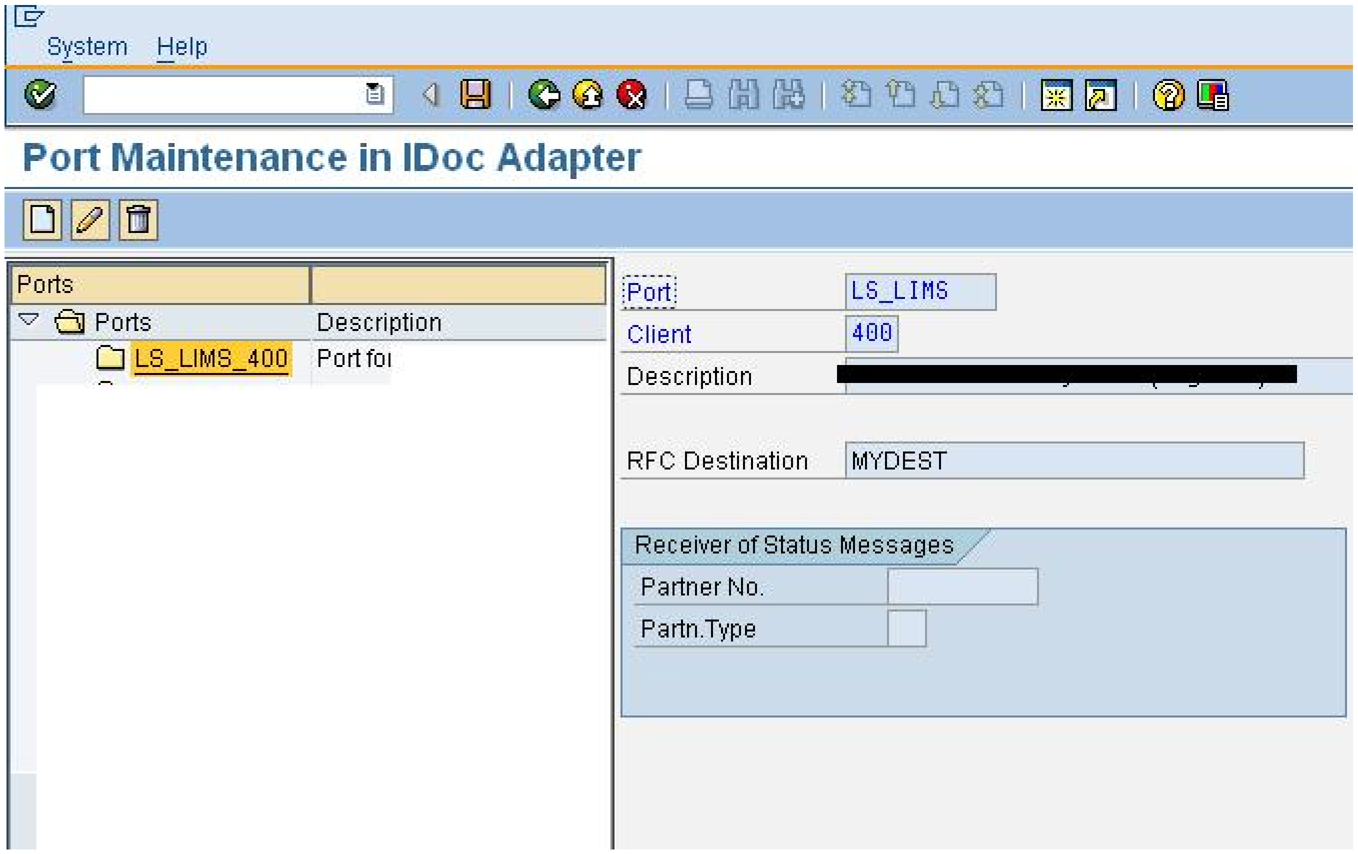
Registered Server Program
Program ID: **Enter appropriate Program ID here**

Start Type of External Program
- [✓ Default Gateway Value]
- Remote Execution
- Remote Shell
- Secure Shell

CPI-C Timeout
- [✓ Default Gateway Value]
- Specify Timeout 60 Defined Value in Seconds

Gateway Options
Gateway Host: [redacted]
Gateway service: [redacted]

3. Create the Port in PI in the transaction IDX1. The port name should be same as Logical system name of the business system defined in SLD (i.e. LS_LIMS). See troubleshooting section for the reason for having. We will use this Port name in the **SNDPOR** control record field of IDOC.



Port Maintenance in IDoc Adapter

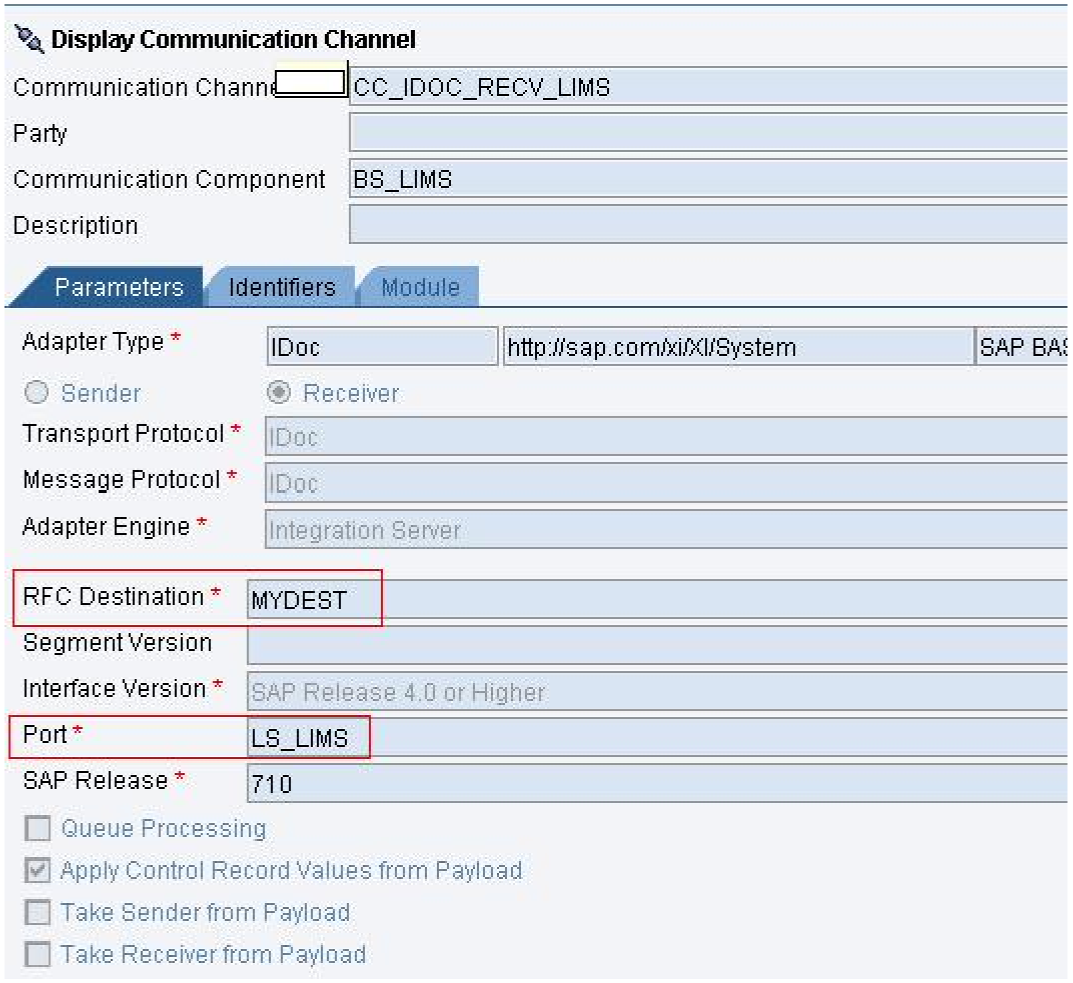
Ports	
Ports	Description
LS_LIMS_400	Port fo

Port	LS_LIMS
Client	400
Description	[redacted]
RFC Destination	MYDEST

Receiver of Status Messages

Partner No.	
Partn.Type	

4. Configure the IDOC Receiver communication channel in PI for sending the IDOC to the LIMS system.



Display Communication Channel

Communication Channel	CC_IDOC_RECV_LIMS
Party	
Communication Component	BS_LIMS
Description	

Parameters | Identifiers | Module

Adapter Type *	IDoc	http://sap.com/xi/XI/System	SAP BAS

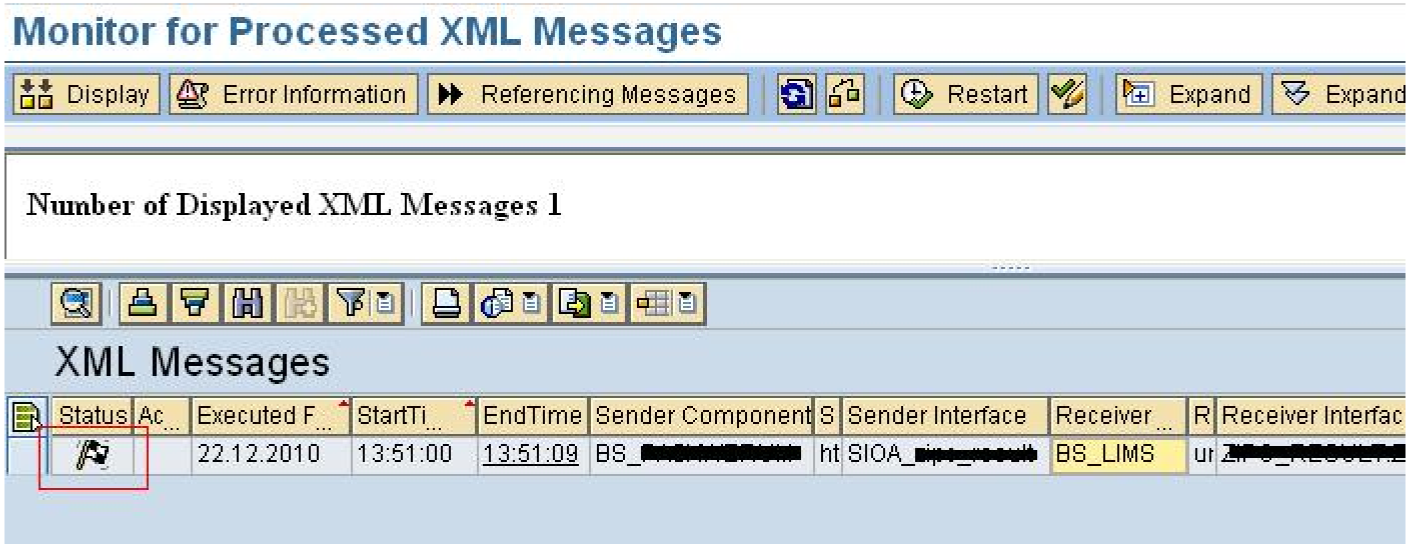
Sender [✓ Receiver]

Transport Protocol *	IDoc
Message Protocol *	IDoc
Adapter Engine *	Integration Server
RFC Destination *	MYDEST
Segment Version	
Interface Version *	SAP Release 4.0 or Higher
Port *	LS_LIMS
SAP Release *	710

- ☐ Queue Processing
- ☑ Apply Control Record Values from Payload
- ☐ Take Sender from Payload
- ☐ Take Receiver from Payload

Rest of the objects like Mappings, Receiver determination etc. are straight forward hence I will skip them.

Testing: A third party non- SAP system sends data to PI via SOAP. PI converts it into IDoc and sends it to LIMS. Check in Transaction SXMB_MONI to verify the status of the message as shown below.

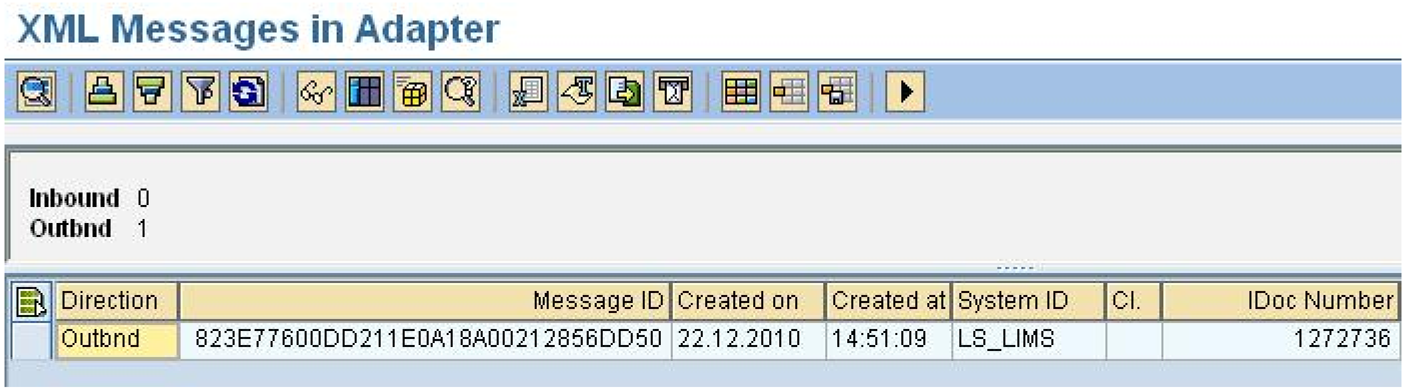


Number of Displayed XML Messages 1

XML Messages

Status	Ac...	Executed F...	StartTi...	EndTime	Sender Component	S	Sender Interface	Receiver ...	R	Receiver Interfac
🏁		22.12.2010	13:51:00	13:51:09	BS_[redacted]	ht	SIOA_[redacted]	BS_LIMS	ur	[redacted]

Also check in transaction IDX5 to monitor IDoc message as shown below.



Inbound 0
Outbnd 1

Direction	Message ID	Created on	Created at	System ID	Cl.	IDoc Number
Outbnd	823E77600DD211E0A18A00212856DD50	22.12.2010	14:51:09	LS_LIMS		1272736

B. Receiving the IDOC from the LIMS system:

Here a LIMS system will send IDocs to SAP PI via TCP/IP protocol. PI will convert IDoc xml into SOAP message and send it to another third party non-SAP system using SOAP.
The SQL*LIMS will create the IDOC into the SQL*LIMS Oracle database with direction “Outbound” from SQL*LIMS, and status as “REQUESTED”.

The LIMS SAP IDOC Interface will then:

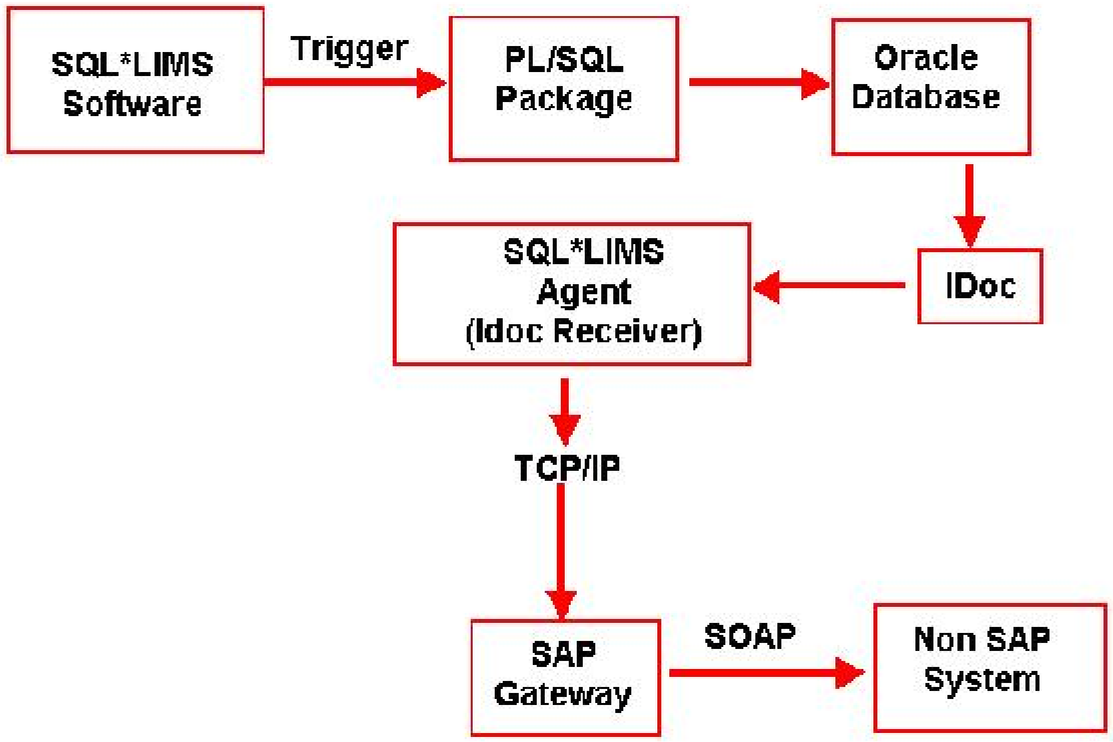
1. Build the IDOC from the Oracle tables;
2. Send the IDOC to the target SAP system.

LIMS Outbound IDOCs will be created by means of PL/SQL procedures/functions/packages called as status actions or events.

As soon as a new outbound transaction is created, its status is set to “CREATING”. During this stage of the process, segments and fields will be added to the IDOC into its internal database tables.
When all the segments and fields have been entered for the current IDOC, the status of the current transaction is set to “REQUESTED”.

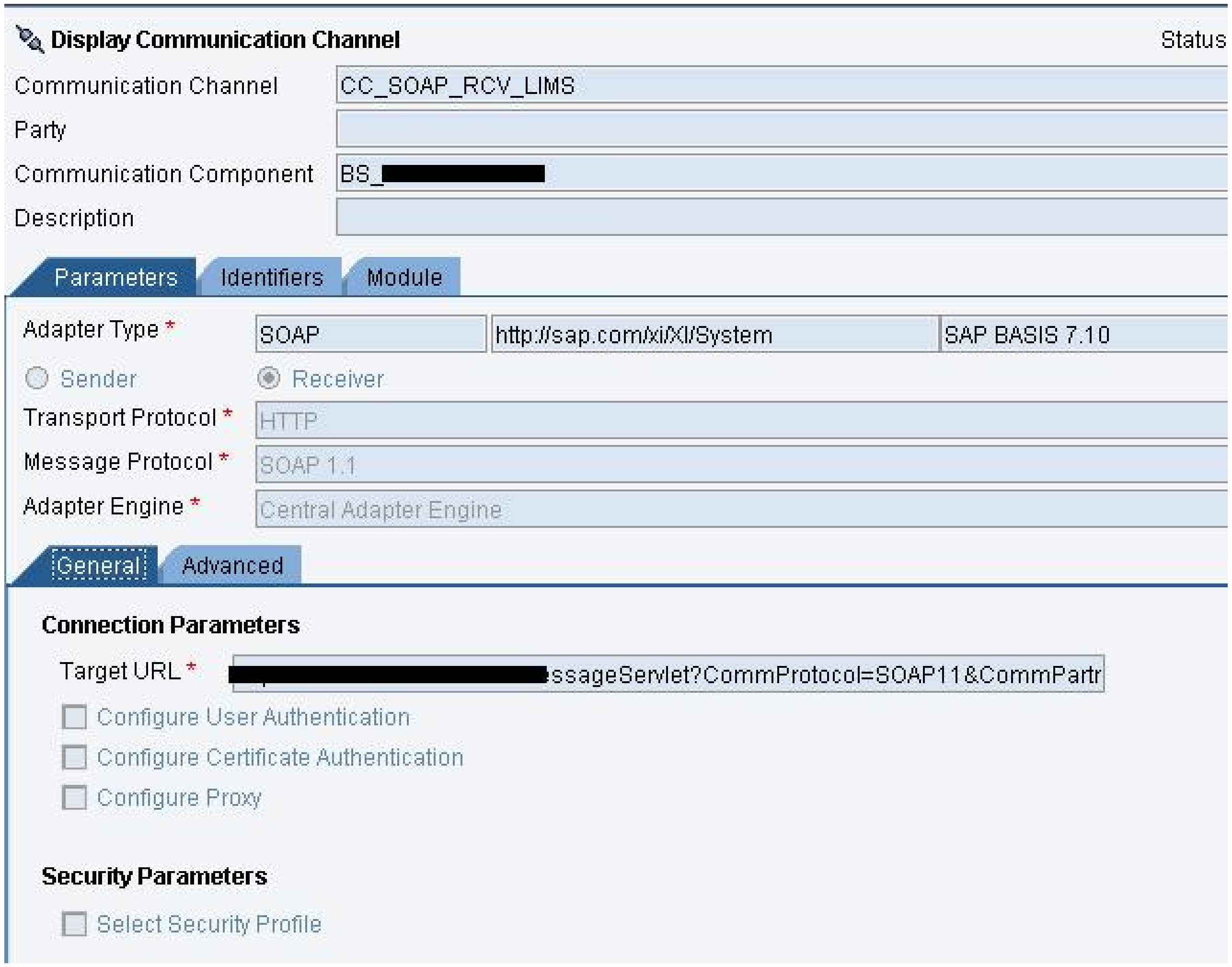
A specific API (PL/SQL package “ABI_IDOC_BUILDER”) allows building the IDOC from its internal tables. At this stage, the status of the transaction is set to “BUILDING”. At the end of this process, the IDOC will be saved as a plain-text clob attachment to the transaction, whose status will be set to “READY”.

When the new transaction is submitted, its status changes to “READY”.
The IdocSender Windows 32bit service will load the IDOC from the relating clob attachment and send it to the target SAP system.
Upon successful completion of sending the IDOC to the SAP, the status of the transaction is set to “COMPLETE”.



The Configuration steps are similar to those mentioned above and are highlighted as follows:

1. Create the Business System for the LIMS system in SLD (System Landscape Directory) with the logical system name (Other Than SAPXXX) mentioned in the Business System. Refer Step 1 above.
2. Create a TCP/IP connection (MYDEST) of type T in transaction SM59 in the PI system. The program ID should be register on SAP PI system. Refer Step 2 above.
3. Create the Port in PI in the transaction IDX1. The port name should be same as Logical system name of the business system defined in SLD (i.e. LS_LIMS). See troubleshooting section for the reason for having. We will use this Port name in the **SNDPOR** control record field of IDOC.
4. The IDoc received from Non sap system will be mapped and converted into SOAP xml and will be sent to another third party system using SOAP protocol. The receiver SOAP channel is as shown below.



Communication Channel	CC_SOAP_RCV_LIMS
Party	
Communication Component	BS_[redacted]
Description	

Adapter Type * SOAP | http://sap.com/xi/XI/System | SAP BASIS 7.10

Sender [✓ Receiver]

Transport Protocol * HTTP

Message Protocol * SOAP 1.1

Adapter Engine * Central Adapter Engine

Connection Parameters

Target URL * [redacted]ssageServlet?CommProtocol=SOAP11&CommPartr

- [ ] Configure User Authentication
- [ ] Configure Certificate Authentication
- [ ] Configure Proxy

Security Parameters

- [ ] Select Security Profile

Give following system details to the LIMS system for connection to PI. Details include:

— Server full name

— Sap System Number

— Gateway Service

— Client

— Program ID

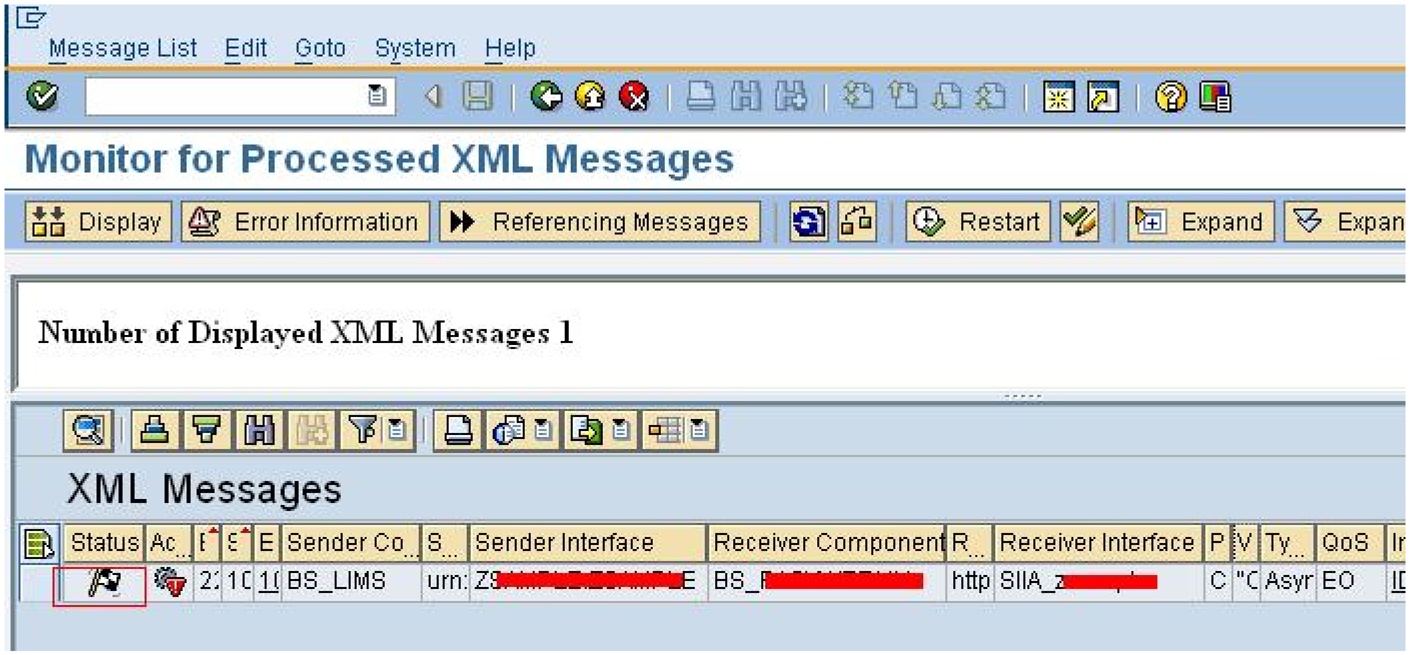
While sending IDoc, the non SAP system will send following Values in the Control record of IDOC.

RCVPOR = SAPXXX

RCVPRT	= LS
SNDPOR	= LS_LIMS
SNDPRT	= LS
SNDPRN	= LS_LIMS
MESTYP	= message type
IDOCTYP	= IDoc type.

Rest of the objects like Mappings, Receiver determination etc. are straight forward hence I will skip them.

Testing: Here LIMS will send IDocs to SAP PI using TCP/IP. Verify the status of the message in Transaction SXMB_MONI



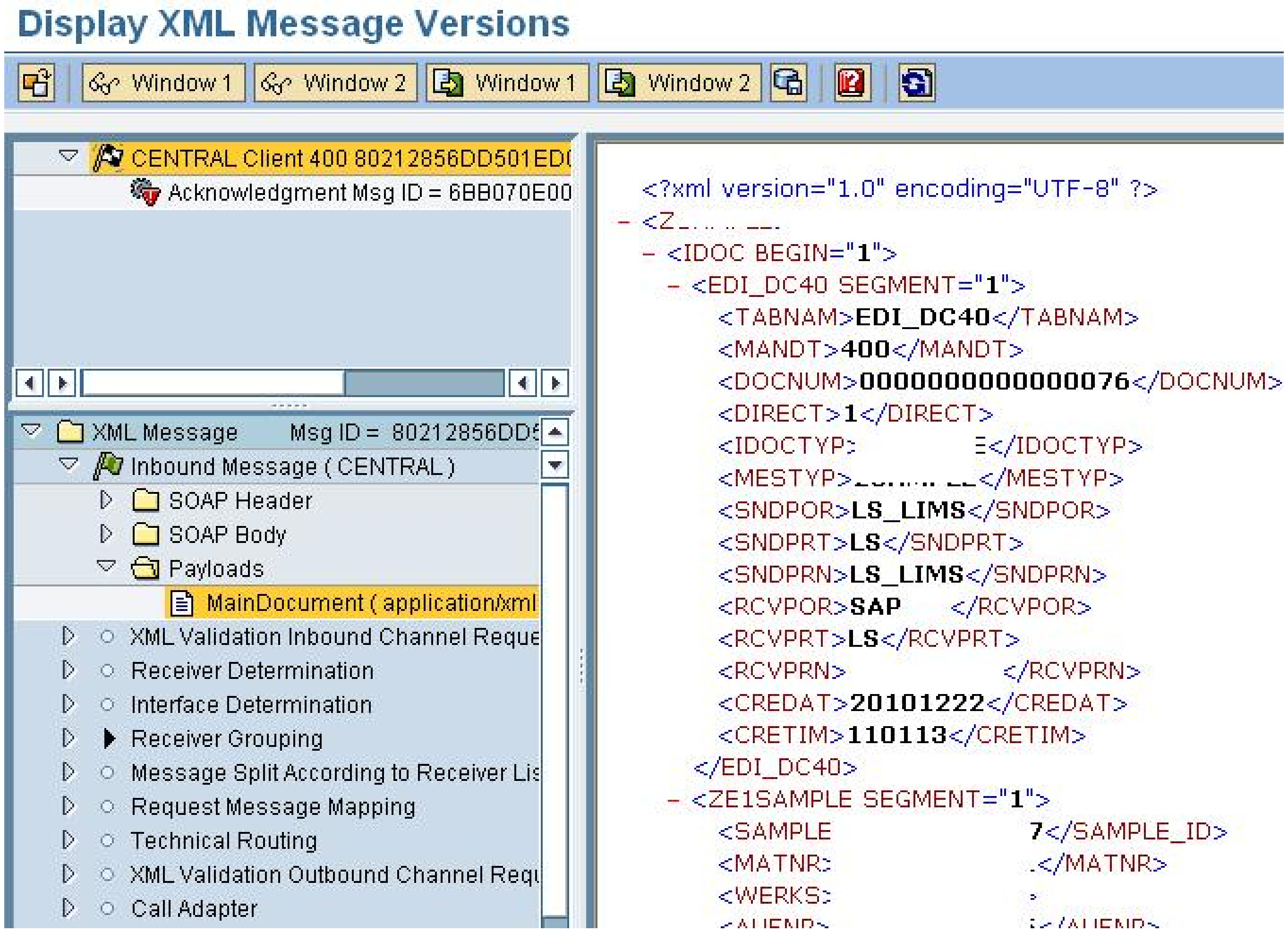
Monitor for Processed XML Messages

Number of Displayed XML Messages 1

XML Messages

Status	Ac..	E	S	E	Sender Co..	S...	Sender Interface	Receiver Component	R...	Receiver Interface	P	V	Ty..	QoS	Ir
		2:	1(	1(	BS_LIMS	urn:	Z[redacted]E	BS_[redacted]	http	SIIA_z[redacted]	C	"C	Asyr	EO	ID

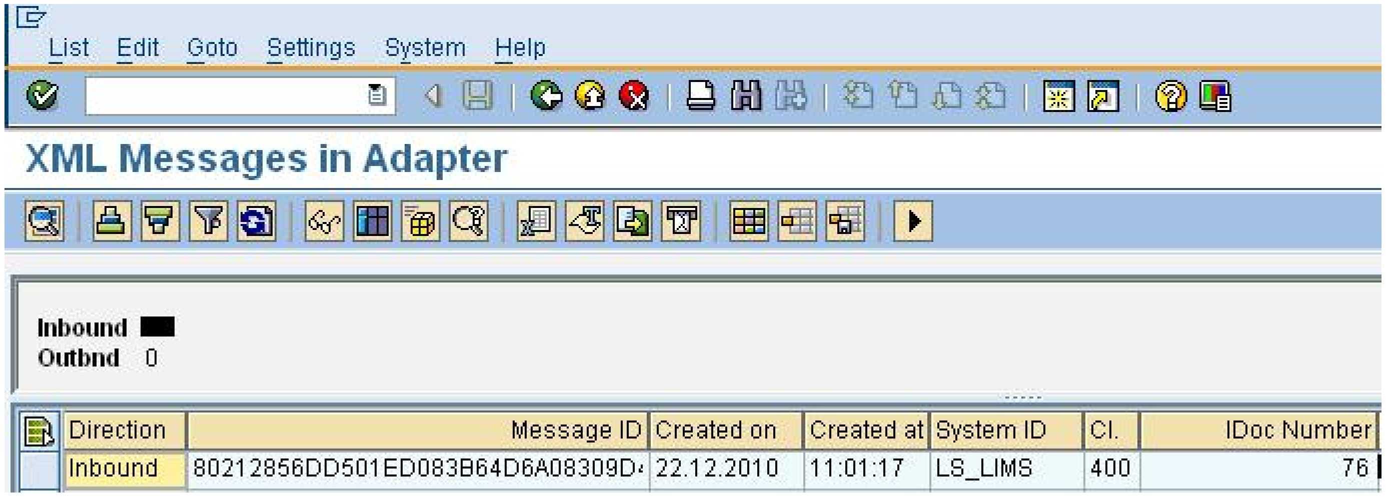
Check the Payload of the incoming message for the control record information as shown below.



```
CENTRAL Client 400 80212856DD501ED
    Acknowledgment Msg ID = 6BB070E00

XML Message      Msg ID = 80212856DD5
  Inbound Message ( CENTRAL )
    SOAP Header
    SOAP Body
    Payloads
      MainDocument ( application/xml
  XML Validation Inbound Channel Reque
  Receiver Determination
  Interface Determination
  Receiver Grouping
  Message Split According to Receiver Lis
  Request Message Mapping
  Technical Routing
  XML Validation Outbound Channel Requ
  Call Adapter
```

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Z[redacted]
 - <IDOC BEGIN="1">
  - <EDI_DC40 SEGMENT="1">
      <TABNAM>EDI_DC40</TABNAM>
      <MANDT>400</MANDT>
      <DOCNUM>0000000000000076</DOCNUM>
      <DIRECT>1</DIRECT>
      <IDOCTYP>[redacted]E</IDOCTYP>
      <MESTYP>[redacted]</MESTYP>
      <SNDPOR>LS_LIMS</SNDPOR>
      <SNDPRT>LS</SNDPRT>
      <SNDPRN>LS_LIMS</SNDPRN>
      <RCVPOR>SAP[redacted]</RCVPOR>
      <RCVPRT>LS</RCVPRT>
      <RCVPRN>[redacted]</RCVPRN>
      <CREDAT>20101222</CREDAT>
      <CRETIM>110113</CRETIM>
    </EDI_DC40>
  - <ZE1SAMPLE SEGMENT="1">
      <SAMPLE[redacted]7</SAMPLE_ID>
      <MATNR>[redacted].</MATNR>
      <WERKS>[redacted]
```

Check the IDoc Message in transaction IDX5 in the SAP PI system.



XML Messages in Adapter

Inbound [redacted]
Outbnd 0

Direction	Message ID	Created on	Created at	System ID	Cl.	IDoc Number
Inbound	80212856DD501ED083B64D6A08309D	22.12.2010	11:01:17	LS_LIMS	400	76

Troubleshooting: While triggering data to SAP PI, sender LIMS application was getting following error message **“No service for system SAPXXX client in Integration Directory”**

Solution: This is traced from the Transaction SM21 where the error is logged with the following solution

No service for system &1, client &2 in Integration Directory
&CAUSE&
Unable to determine the service for system &1, client &2.
&SYSTEM_RESPONSE&
You tried to determine a service for system &1, client &2.
The system ID is taken from the sender port of the IDoc control record,
which must have the form 'SAP' + system ID. In the case of external
systems, in the sender port field you must enter the logical system
that is assigned to the service in the Integration Directory.
&WHAT_TO_DO&
Maintain a service in the Integration Directory for system &1,
client &2.
&SYS_ADMIN&

Follow Like RSS Feed

Alert Moderator

Alerting is not available for unauthorized users

Assigned Tags

Cloud Integration

Similar Blog Posts

SAP PI 7.1 Tips and Tricks

By Former MemberJul 28, 2013

SAP NetWeaver PI 7.1(EHP1) and PO 7.3(EHP1) in Retrospect

By Former MemberMay 21, 2012

Track - SAP CPI generated Idocs in SAP ERP

By Former MemberApr 10, 2019

Related Questions

Is SAP PI required for HCM and SF Integration

By Sudesh MGApr 24, 2018

What is CPI-C in "Ports in Idoc processing (we21)?

By Former MemberApr 08, 2007

program ______.MDS not registered..PI 7.1 EHP 1-MDM 7.1 Integration-ABAP API

By Former MemberMar 12, 2011

/

3 Comments

You must be Logged on to comment or reply to a post.

Ravi Kanth Talagana

August 22, 2013 at 8:37 am

Hi Deepak Shah,

Very nicely written blog,

I am trying to do a similar scenario (X->PI->non SAP using IDOCs.

But im not able to get the solution working:

http://scn.sap.com/thread/3409634

Can you please help?

Best Regards,

Ravi

Like 0

Share

Right click and copy the link to share this comment

Former Member

September 27, 2016 at 7:55 am

Nice work!

Like 0

Share

Right click and copy the link to share this comment

Former Member

September 28, 2016 at 9:05 am

Hi Deepak,

We were facing issues in a similar scenario. With the help of the screenshots and info provided were able to successfully develop the scenario.

Good Work !!!

Like 0

Share

Right click and copy the link to share this comment

Find us on

Privacy

Terms of Use

Legal Disclosure

Copyright

Trademark

Newsletter

Support