

How to...

Create the StarterPack Scenarios

EXCHANGE INFRASTRUCTURE



A SAP “How to...” Paper



**Applicable Releases: XI 2.0, Support Package 2
August, 2003**

SAP Labs (SAP Labs, LLC) assumes no responsibility for errors or omissions in these materials.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

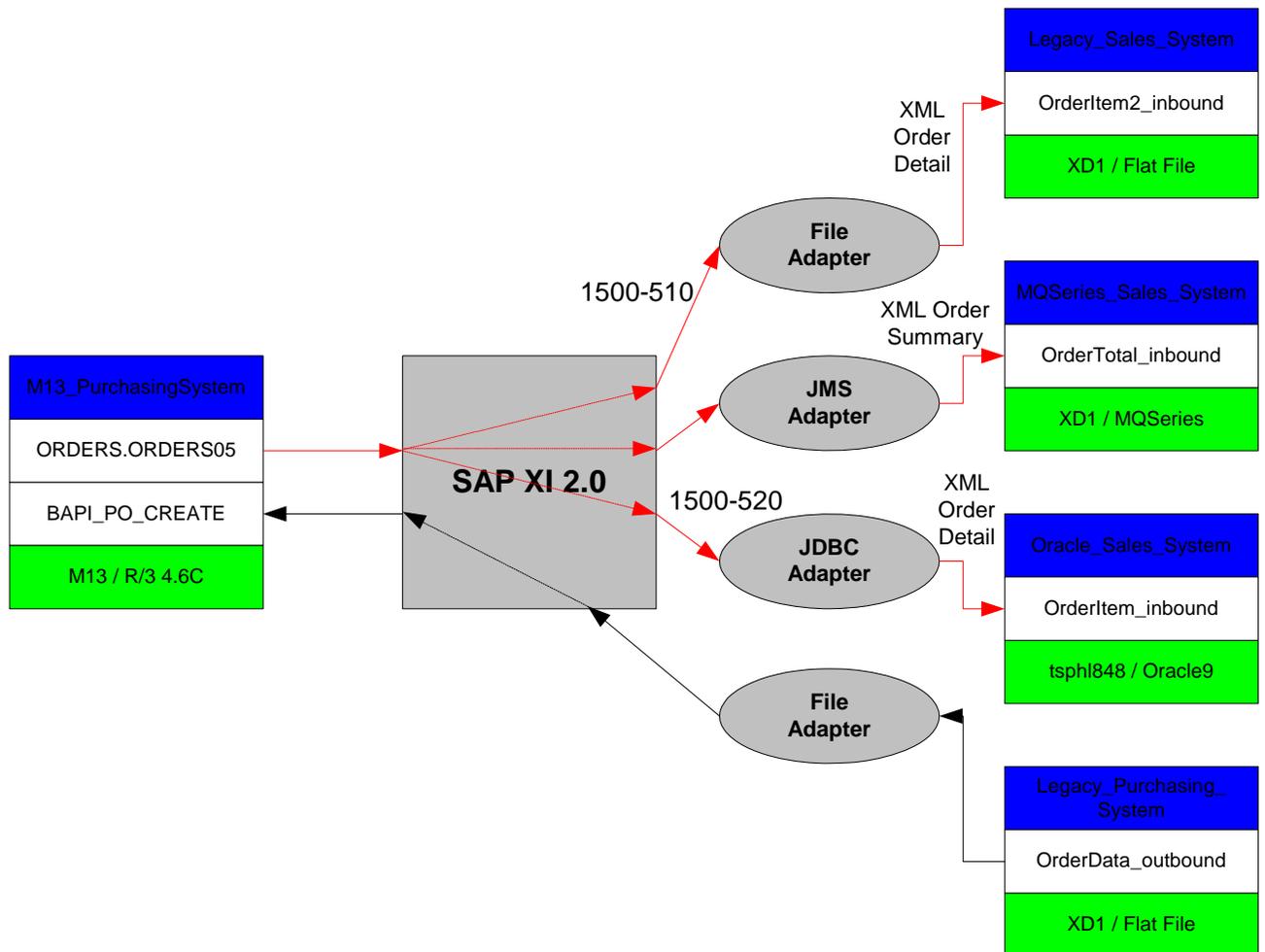
SAP Labs shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP Labs does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP Labs has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

1 Objective

This document describes the steps involved in creating the XI StarterPack scenarios. These scenarios can be used as templates or references during customer implementations. The scenarios include the use of 5 adapters (IDoc, RFC, File, JDBC and JMS), exchanging messages between 5 business systems.

The IDoc data sent from the R/3 system will be sent to 3 different systems. 2 of the systems will contain detail information; the 3rd will contain summarized information. When sending the detail information, the material number in the IDoc will be examined. Depending on whether the material number is '1500-510' or '1500-520', different receiving systems will be used.



2 The Process

In developing an XI application, the following steps are normally followed, as required:

System Landscape Directory:

1. Define software component and version
2. Define technical system
3. Define business system

Integration Repository: (Design)

1. Import software component version
2. Define namespace for the scenario
3. Define data types
4. Define message types
5. Define message interfaces
6. import IDOCs or RFCs
7. import archives containing XSLT or java mapping objects
8. Define message mappings
9. Define interface mappings

Integration Directory: (Configuration)

1. Define value mapping as needed
2. Define receiver system end points and logon data (as required)
3. Define interface determination for outbound interfaces
4. Define business scenario (as needed) and receiver determination

Adapter Configurations: (to be done as appropriate, depending on the scenario)

1. RFC
2. File
3. JDBC
4. JMS
5. IDoc

3 The Step-By-Step Solution

1. Preliminary remarks

The ALE configuration on R/3 to send an IDoc to XI will need to be completed before the scenario can begin. It is even advisable to send an IDoc before any configuration is done. Even though this will cause an error on XI, but it will be a good test to insure all the R/3 configurations and network connections are correct.

Since for our scenarios, a flat file from the Legacy_Purchasing_System is to be sent to XI, it is also advisable for us to configure the file adapter and send a document to XI for examination. This will also cause an error on XI (since we don't have XI configuration available, such as namespace, interface name), but the payload of the error'd message, which can be displayed in the XI monitor, can be examined and used during our data type creation process.

To purposely cause errors before any design and configuration have several advantages. It is good way to determine connections are done correctly, and it allows us to look at the payload of the messages as a reference for our subsequent configurations.

The System Landscape Directory configurations are very straight forward. Since configuration wizards are used during the configurations, we will skip its configurations in our how-to instructions.

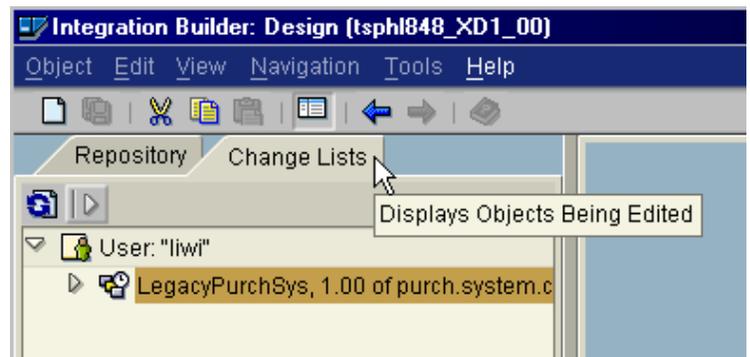
2. Please remember that during all the subsequent steps, all the changes must be saved and activated.

Save

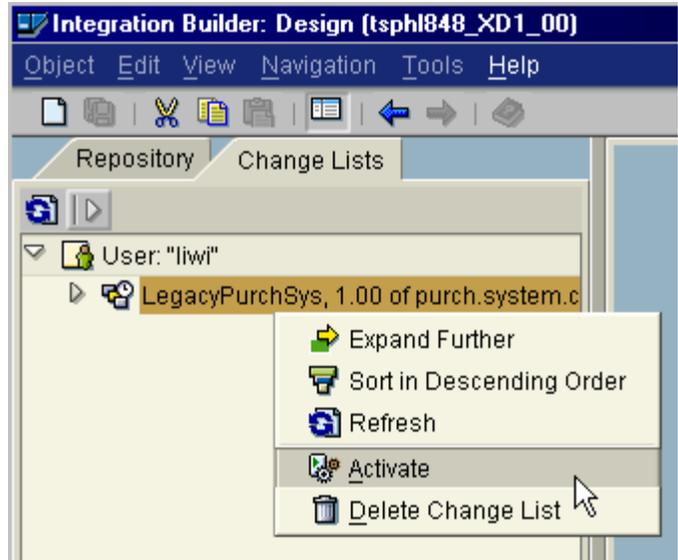


and Activate (see instructions on the right).

To activate, click on the "Change Lists" tab:



Right click on the software component and select "Activate".



3. Integration Repository (IR) configurations.

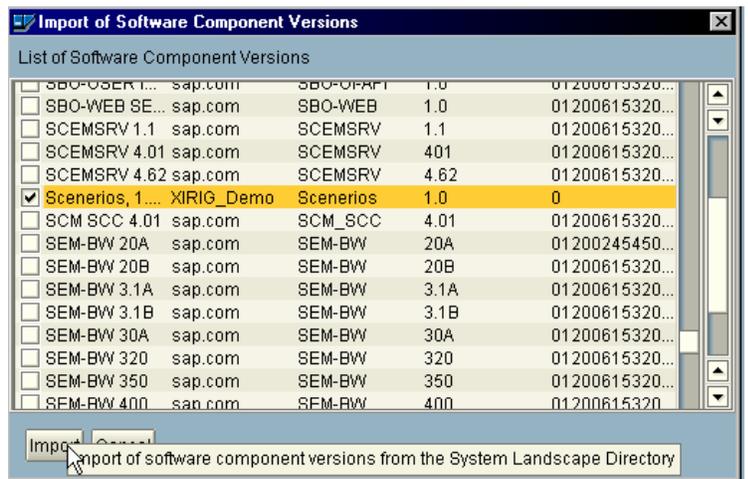
The IR is used to define all the data structures and mappings for the scenarios. All of the objects and components created are independent of any systems. The objects can be reused for more than one scenario.

All of the objects must be define within a software component. If one does not exist, then it will need to be imported.

To import a software component, click Object → Import Software Component Versions



The software component list will be loaded from the SLD. Select the component and click "Import".



4. Create namespace for the scenarios

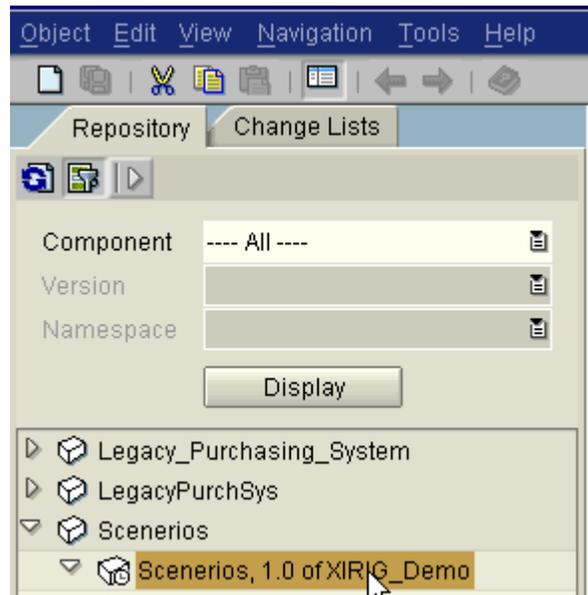
The namespace is used to distinguish objects that may have similar names. It is like having your own “sandbox” to develop the scenarios.

It is advisable that the software component used to develop the scenarios not be allowed to import IDocs and RFCs. It is better to designate a special software component, such as a Basis component, to host the IDocs and RFCs. This can eliminate duplication of IDocs and RFCs across more than 1 software components. Duplications can cause maintenance problems when changes occur.

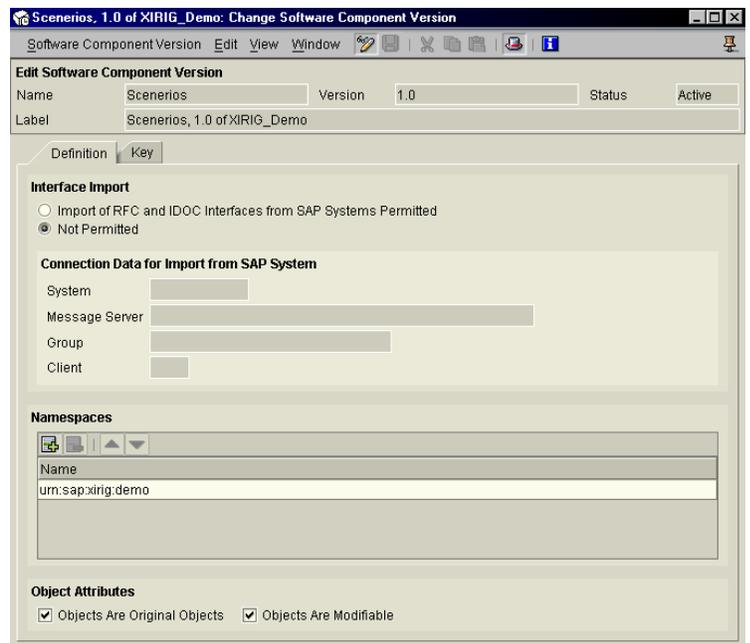
The namespace name should follow a naming convention, usually of an URL or URN.

In situations when import of IDocs and RFCs is selected, the system connection information must be provided.

To create a namespace, double-click on the software component.



Enter a namespace name in the following screen:



5. Define data types.

The formats of the data structures for inbound and outbound interfaces are defined with data types.

For our scenarios, we have the

To create a new data type, right-click on the “Data Types” and select “New”.

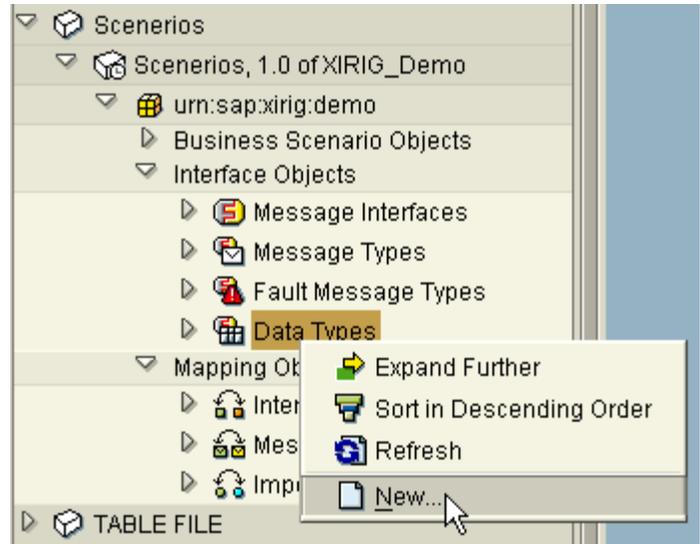
HOW TO... CREATE THE STARTERPACK SCENARIOS

following data types:

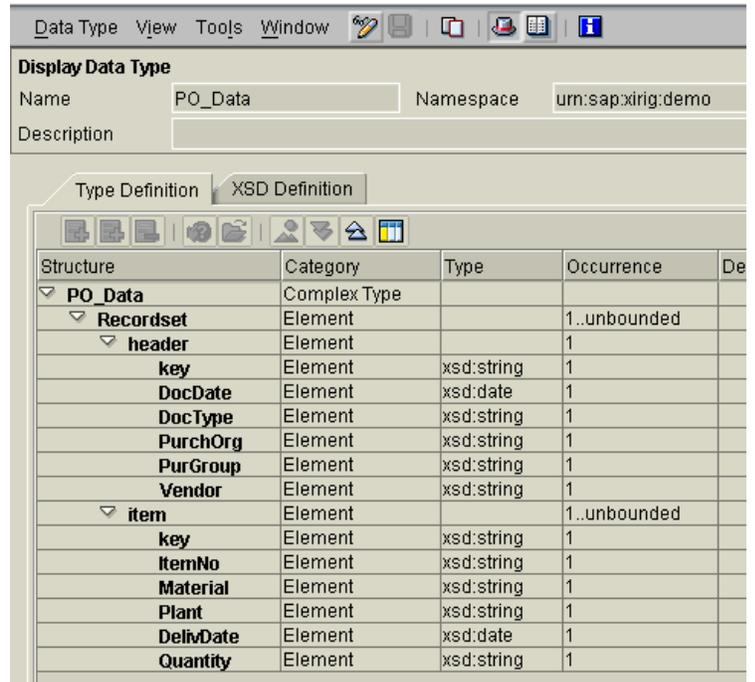
1. Outbound PO info sent from the Legacy_Purchasing_System.
2. Inbound PO detail sent to the Legacy_Sales_System.
3. Inbound PO detail sent to the Oracle_Sales_System.
4. Inbound PO summary sent to the MQSeries_Sales_System.

The names can be arbitrary.

For RFCs and IDocs, the meta data can be imported from the R/3 system, therefore, they need not be defined here.



Create data type for outbound PO info from the Legacy_Purchasing_System:



Create data type for inbound PO info to the Legacy_Sales_System and Oracle_Sales_System:

HOW TO... CREATE THE STARTERPACK SCENARIOS

Data Type View Tools Window

Display Data Type

Name: resultset Namespace: urn:sap.xirig:demo Status:

Description: JDBC Adapter based Interface

Type Definition XSD Definition

Structure	Category	Type	Occurrence	Details
resultset	Complex Type		1..unbounded	
row	Element		1	
CUSTOMER_ID	Element	xsd:string	1	length="10"
ORDER_ID	Element	xsd:string	1	length="20"
ITEM_ID	Element	xsd:string	1	length="3"
PRODUCT_ID	Element	xsd:string	1	length="15"
PRODUCT_DESC	Element	xsd:string	1	length="60"
PRICE	Element	xsd:string	1	length="20"
ORDERDATE	Element	xsd:string	1	length="16"

Create data type for inbound PO summary info to the MQSeries_Sales_System:

Data Type View Tools Window

Display Data Type

Name: OrderTotal Namespace: urn:sap.xirig:demo Status:

Description: OrderTotal

Type Definition XSD Definition

Structure	Category	Type	Occurrence	Details
OrderTotal	Complex Type			
customer_id	Element	xsd:string	1	length="10"
order_id	Element	xsd:string	1	length="20"
total	Element	xsd:string	1	length="20"
orderdate	Element	xsd:string	1	length="18"

6. Create message types.

Message types reference the data types directly. They have one-to-one relationships.

The message type names can also be identical (but, not necessary) to the data types.

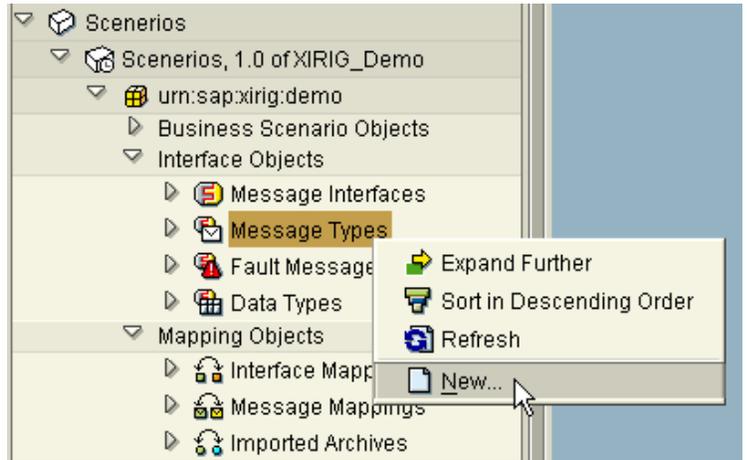
In our case, we will use the following names from data types to message types:

OrderTotal → OrderTotal

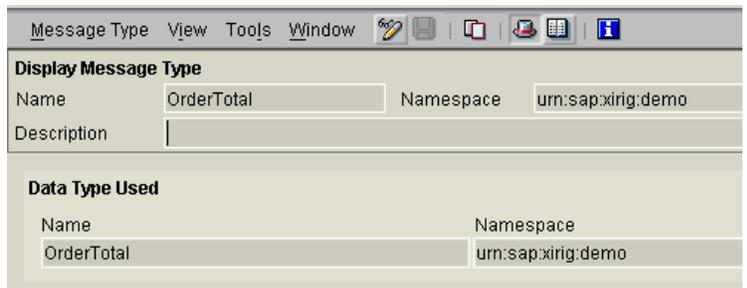
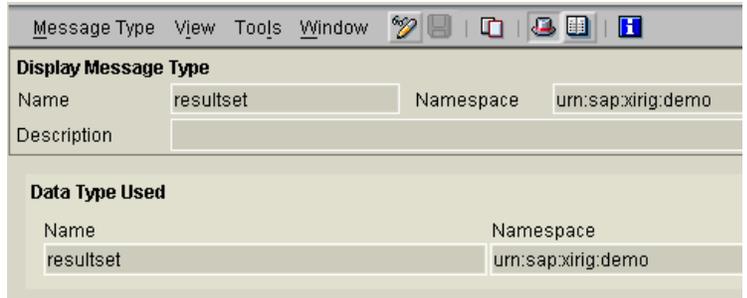
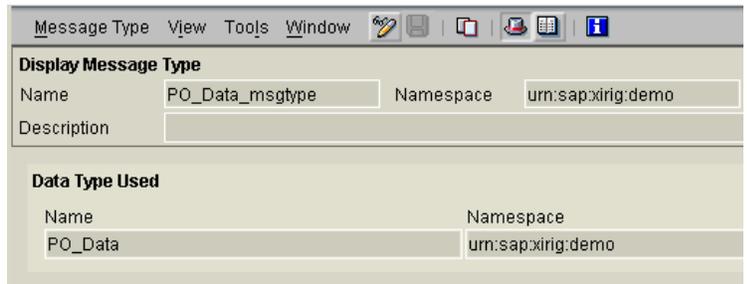
resultset → resultset

PO_Data → PO_Data_msgtype

To create message type, right-click on “Message Types” and select “New”:



Create the following message types:



7. Create message interfaces.

Message interfaces define the mode (synch or asynch) and direction (inbound or outbound) of the interfaces.

The message interfaces are defined using message types, which are defined using data types. Since IDocs and RFCs are not defined using data types, therefore they are also not to be defined here.

The direction of an interface is always from the perspective of the business system. In our scenarios, for the PO_Data_msgtype, the direction is outbound. All of the others are inbound.

Also, for all of our scenarios, the mode is asynchronous.

In order to ease configuration and clarify the interface type, it is recommended to include in the interface name the direction (e.g. inbound or outbound).

In our scenarios, we will create the following interfaces:

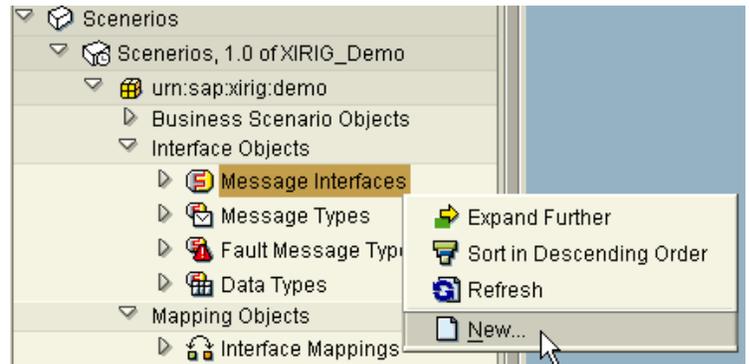
OrderData_outbound: for the Legacy_Purchasing_System using the message type, PO_Data_msgtype.

OrderItem_inbound: for the Oracle_Sales_System using the message type, resultset.

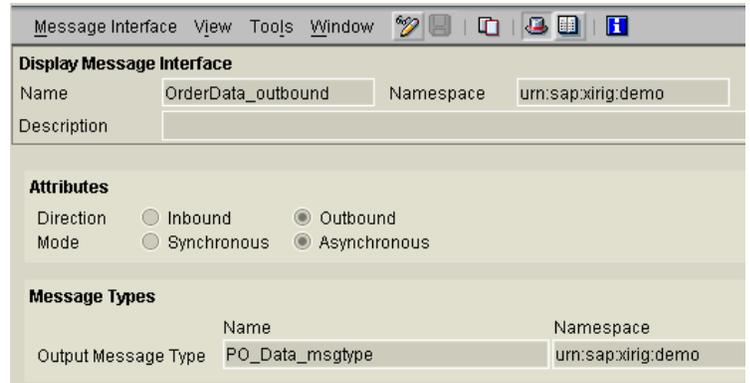
OrderItem2_inbound: for the Legacy_Sales_System using the message type, resultset. (Here we are re-using a msgtype.)

OrderTotal: for the MQSeries_Sales_System using the message type, OrderTotal.

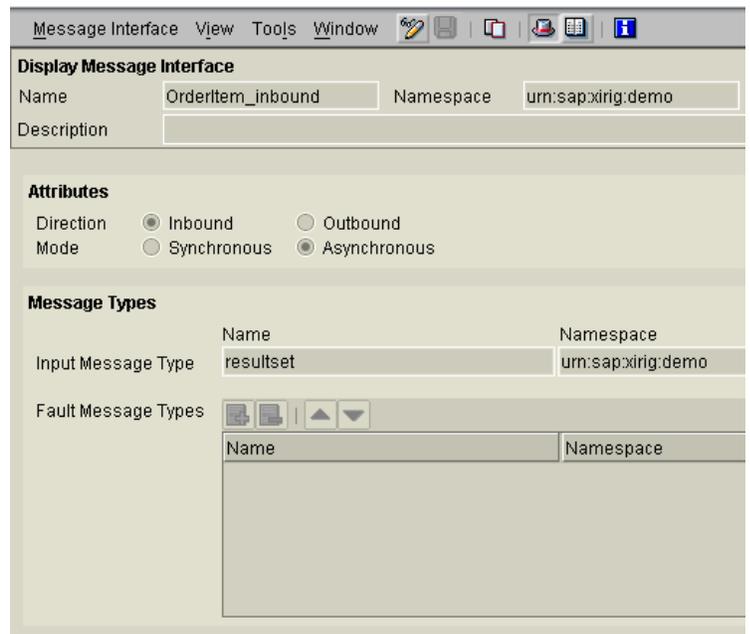
To create message interface, right-click on “Message Interface” and select “New”:



Create the interface for PO_Data_msgtype to be used with the Legacy_Purchasing_System: (direction is outbound, mode is asynch)



Create the interface for resultset to be used with the Oracle_Sales_System: (direction is inbound, mode is asynch)



Create the interface for resultset to be used with the

Legacy_Sales_System: (direction is inbound, mode is asynch)

The screenshot shows the 'Message Interface' configuration window. The title bar includes 'Message Interface', 'View', 'Tools', and 'Window'. The main area is titled 'Display Message Interface'. It contains the following fields and sections:

- Name:** OrderItem2_inbound
- Namespace:** urn:sap:xirig:demo
- Description:** (empty text box)
- Attributes:**
 - Direction:** Inbound, Outbound
 - Mode:** Synchronous, Asynchronous
- Message Types:**
 - Input Message Type:** resultset
 - Namespace:** urn:sap:xirig:demo
 - Fault Message Types:** (empty table with columns Name and Namespace)

Create the interface for OrderTotal to be used with the MQSeries_Sales_System: (direction is inbound, mode is asynch)

The screenshot shows the 'Message Interface' configuration window. The title bar includes 'Message Interface', 'View', 'Tools', and 'Window'. The main area is titled 'Display Message Interface'. It contains the following fields and sections:

- Name:** OrderTotal_inbound
- Namespace:** urn:sap:xirig:demo
- Description:** (empty text box)
- Attributes:**
 - Direction:** Inbound, Outbound
 - Mode:** Synchronous, Asynchronous
- Message Types:**
 - Input Message Type:** OrderTotal
 - Namespace:** urn:sap:xirig:demo
 - Fault Message Types:** (empty table with columns Name and Namespace)

HOW TO... CREATE THE STARTERPACK SCENARIOS

8. Import IDoc and RFC.

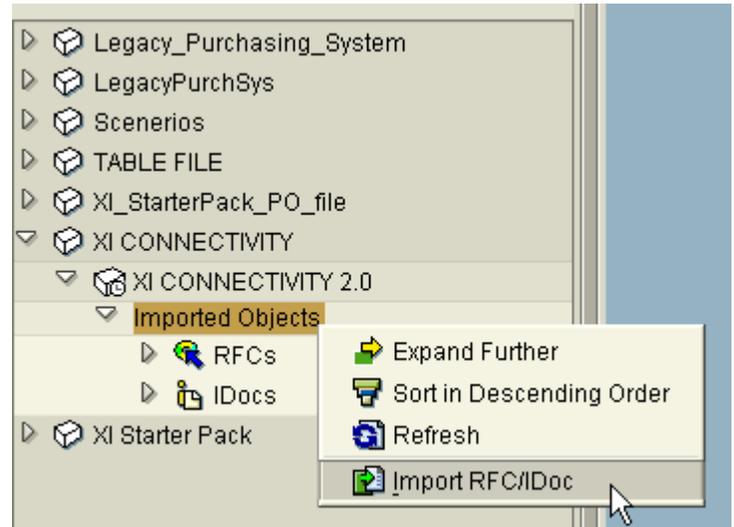
The StarterPack requires the use of an IDoc and an RFC. The meta data must be imported from the R/3 systems.

A separate software component has been imported to contain these objects.

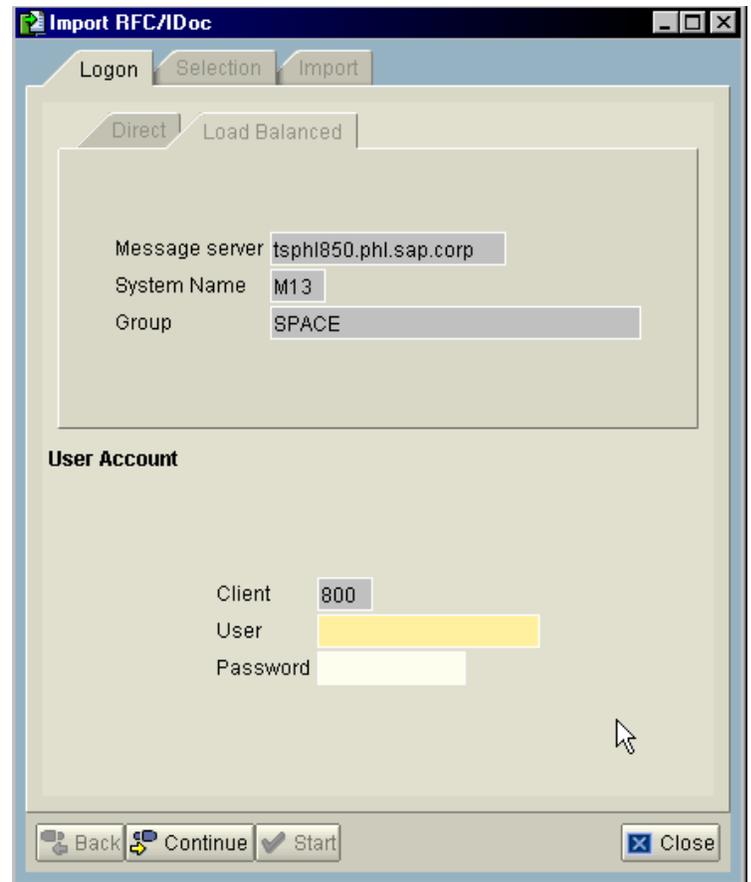
The IDoc to be import is:
ORDERS.ORDERS05

The RFC to be import is:
BAPI_PO_CREATE

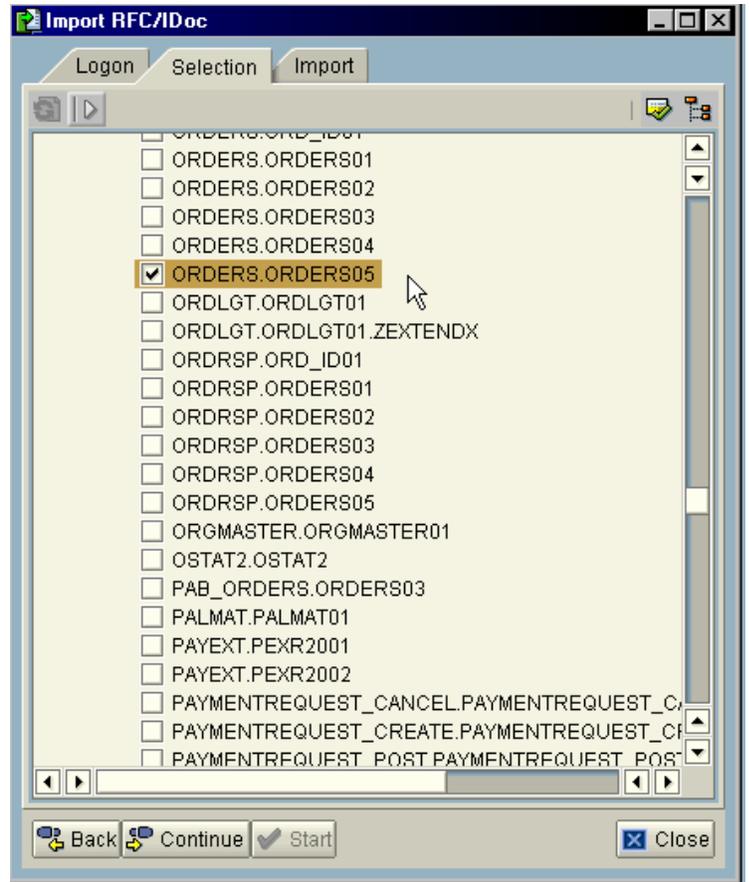
To import IDoc/RFC, right-click on the “Imported Objects” of the Software Component, and select “Import RFC/IDoc”:



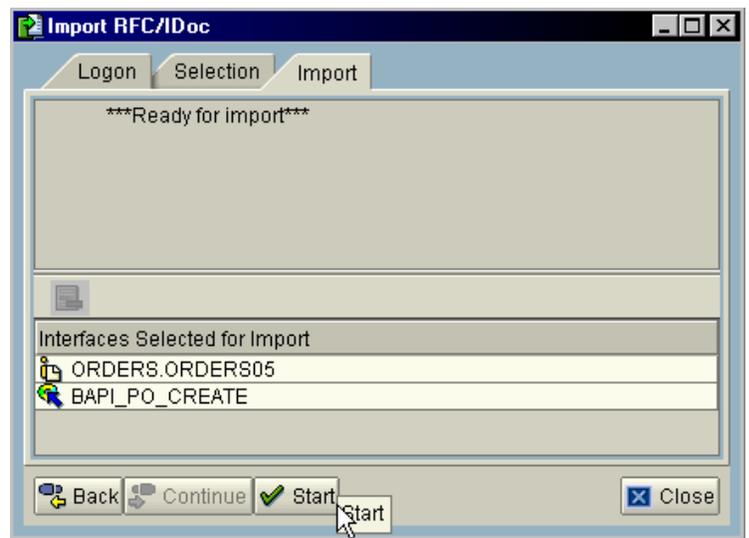
When a logon screen pops up, enter valid R/3 logon credentials:



A list of valid RFCs and IDocs can be selected:



Click on "Continue" to start the import:



After import, the following objects should be visible:



9. Now that we have the data structure and message types defined, we can proceed to create the mappings.

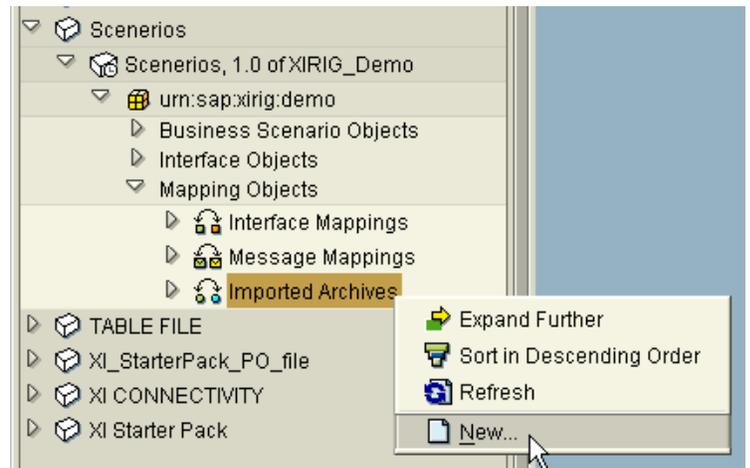
10. For the StarterPack scenarios, XSLT is used in the mapping to summarize the IDoc data for inbound to the MQSeries_Sales_System, and also to extract the detailed IDoc data for inbound to Oracle_Sales_System and Legacy_Sales_System.

The XSLT must be created externally to XI and be imported into the archive of the namespace.

For this part, it is assumed that an archive file containing the XSLT is available on the local system.

It is not the intent for this How-To guide to teach XSLT mapping.

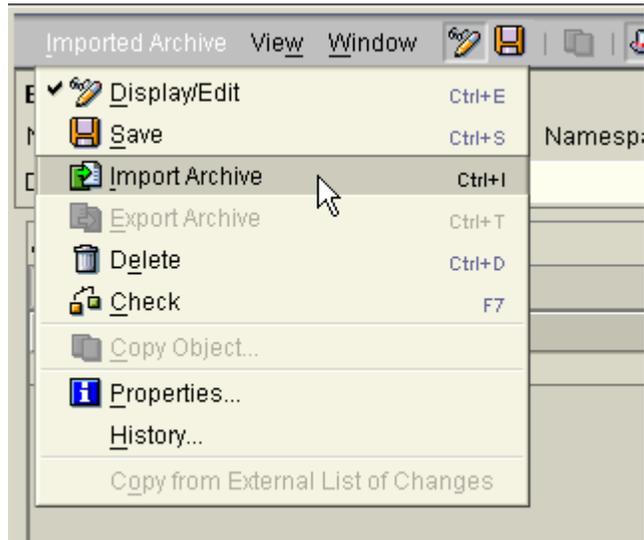
To import an archive, right-click on "Imported Archives" and select "New":



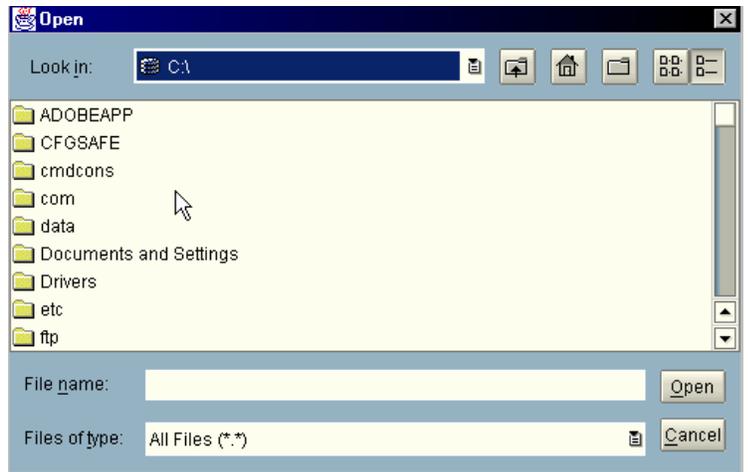
Enter a name for the archive:



Then, select menu: Imported Archive → Import Archive



Then, select the archive file from the Open file dialog box:



11. Create message mapping from the PO data file to BAPI_PO_CREATE.

The mapping name should be representative of the message type involved. In this case, the message types are PO_Data_msgtype and BAPI_PO_CREATE, so we can give it a name of PO_to_BAPI.

During the creation process, drag the PO_Data_msgtype to the left hand of the mapping screen over

 **Choose a Source Message**. Drag

BAPI_PO_CREATE to the right side of the mapping screen over

 **Choose a Target Message**. The meta data for the message types will appear from which graphical mapping can be performed.

Due to the number of fields involved, it is not feasible to show every screen of the graphical mapping.

It is assumed that the reader possesses basic graphical mapping knowledge.

The following mapping spec are used:

```
/BAPI_PO_CREATE/PO_ITEMS/item=/PO_Data_msgtype/Recordset/item
```

```
/BAPI_PO_CREATE/PO_ITEMS/item/PO_ITEM=/PO_Data_msgtype/Recordset/item/ItemNo
```

```
/BAPI_PO_CREATE/PO_ITEMS/item/PUR_MATERIAL=/PO_Data_msgtype/Recordset/item/Material
```

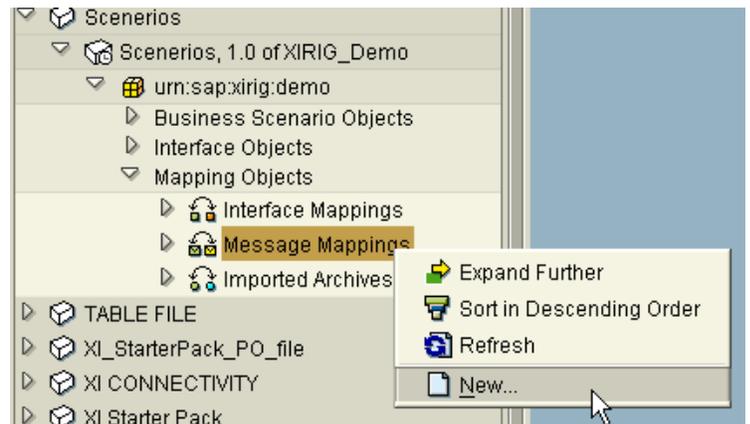
```
/BAPI_PO_CREATE/PO_ITEMS/item/PLANT=/PO_Data_msgtype/Recordset/item/Plant
```

```
/BAPI_PO_CREATE/PO_ITEM_SCHEDULES/item=/PO_Data_msgtype/Recordset/item
```

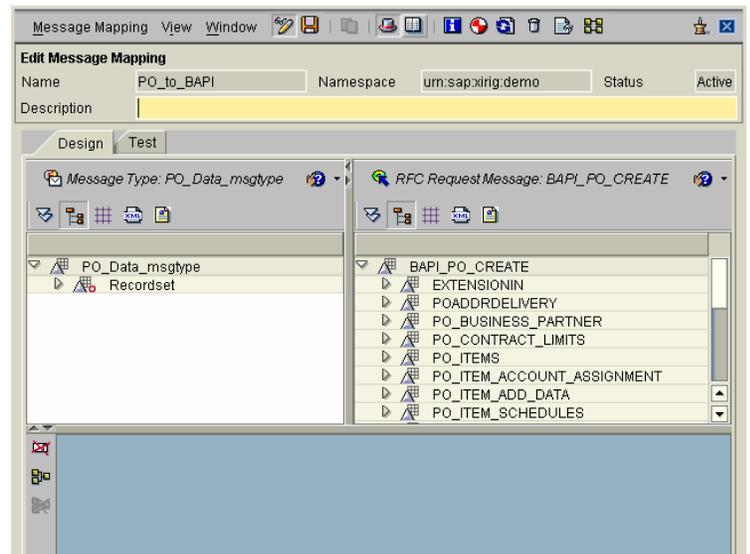
```
/BAPI_PO_CREATE/PO_ITEM_SCHEDULES/item/PO_ITEM=/PO_Data_msgtype/Recordset/item/ItemNo
```

```
/BAPI_PO_CREATE/PO_ITEM_SCHEDULES/item/DELIV_DATE=/PO_Data_msgtype/Recordset/item/DelivDate
```

To create a mapping, right-click on “Message Mapping” and select “New”:



Resulting view after dragging the message types to the mapping screen:



HOW TO... CREATE THE STARTERPACK SCENARIOS

/BAPI_PO_CREATE/PO_ITEM_SCHEDULE
S/item/QUANTITY=/PO_Data_msgtype/Recordset/item/Quantity

/BAPI_PO_CREATE/PO_HEADER=/PO_Data_msgtype/Recordset/header

/BAPI_PO_CREATE/PO_HEADER/DOC_DATE=/PO_Data_msgtype/Recordset/header/DocDate

/BAPI_PO_CREATE/PO_HEADER/DOC_TYPE=/PO_Data_msgtype/Recordset/header/DocType

/BAPI_PO_CREATE/PO_HEADER/PURCH_ORG=/PO_Data_msgtype/Recordset/header/PurchOrg

/BAPI_PO_CREATE/PO_HEADER/PUR_GROUP=/PO_Data_msgtype/Recordset/header/PurGroup

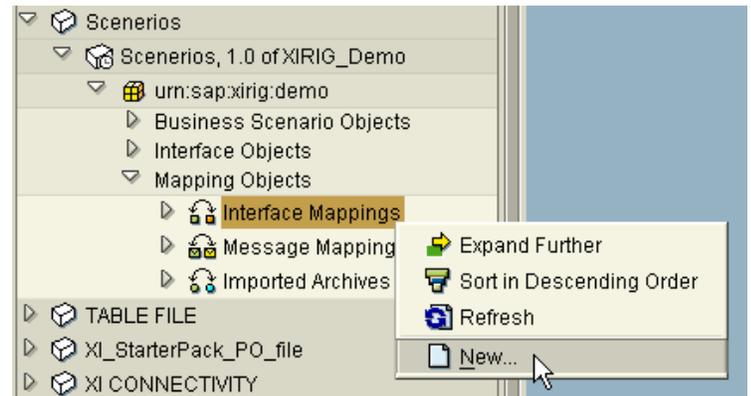
/BAPI_PO_CREATE/PO_HEADER/VENDOR=/PO_Data_msgtype/Recordset/header/Vendor

12. Create Interface Mappings.

There are 3 mappings involved in the StarterPack scenarios for 4 interface mappings. One of which is used twice.

1. PO_Data_msgtype → BAPI_PO_Create
2. ORDERS05 → resultset (this is used twice)
3. ORDERS05 → OrderTotal

To create an Interface Mapping, right-click on "Interface Mappings" and select "New":



Create PO_Data_msgtype → BAPI_PO_Create

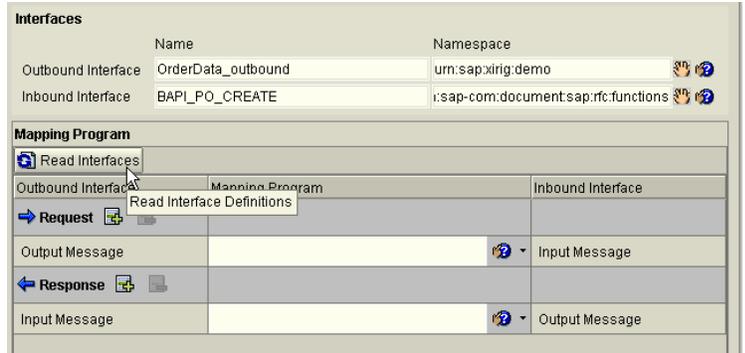
Note: Message Mapping is used

After dragging the interface to the , click on

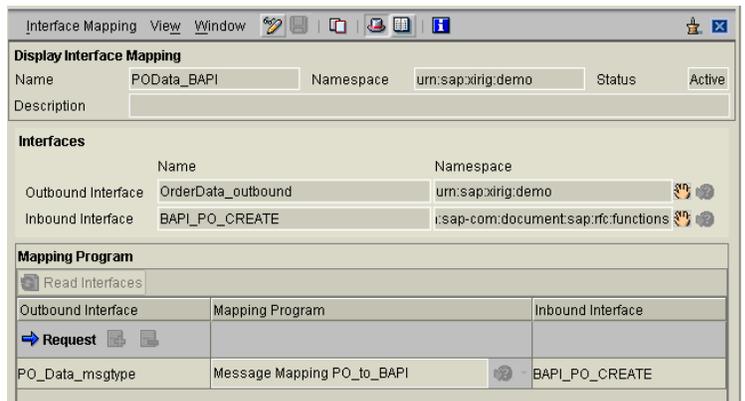
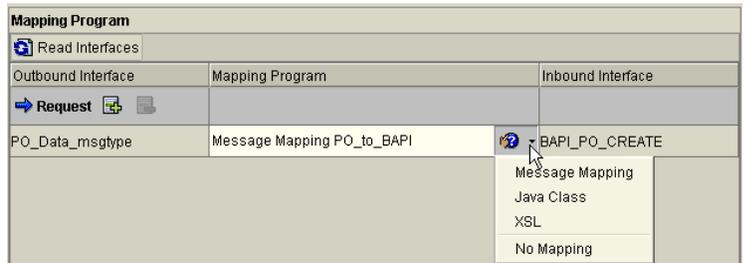
 Read Interfaces

. The message types will automatically be filled in on the lower half of the screen.

HOW TO... CREATE THE STARTERPACK SCENARIOS



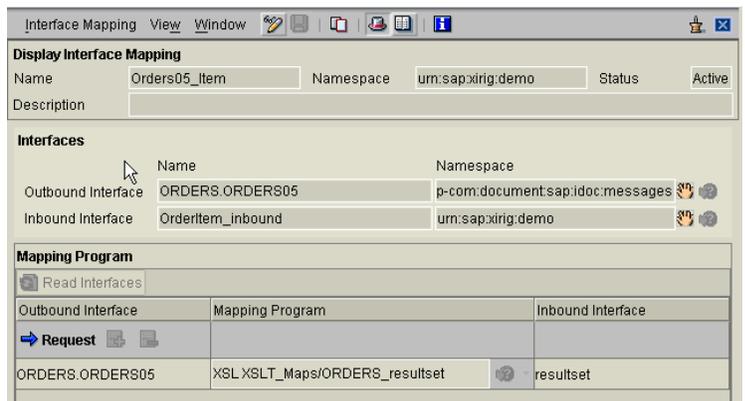
Select Message Mapping by left-click on the , and select the mapping method.



ORDERS05 → resultset (OrderItem_inbound for Oracle)

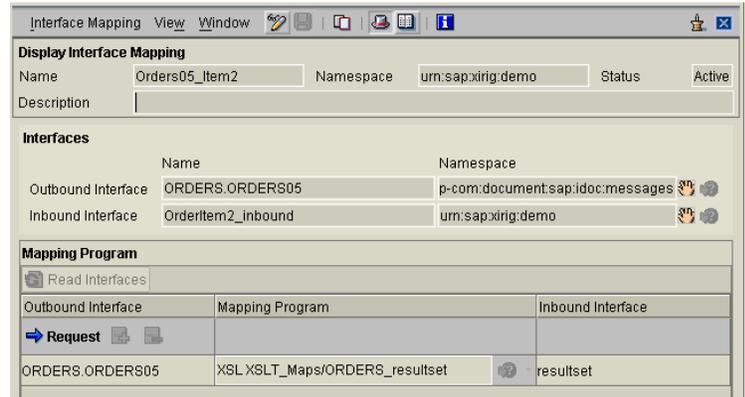
Note: XSLT Mapping is used

Follow the same process as above.



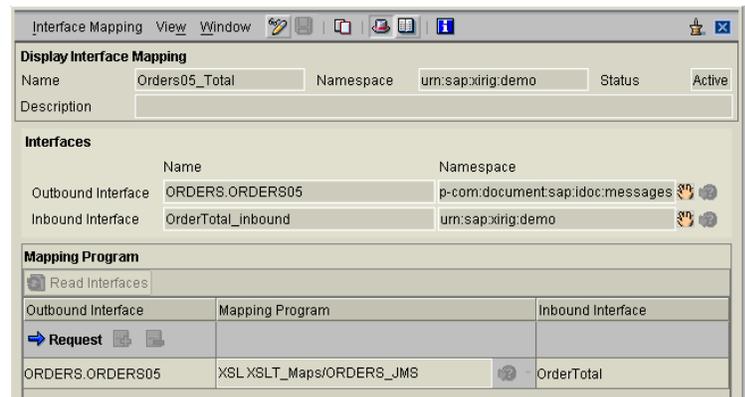
ORDERS05 → resultset (OrderItem2_inbound for file)

Note: XSLT Mapping is used



ORDERS05 → OrderTotal (for MQSeries)

Note: XSLT Mapping is used



13. The work involved with the IR is now complete. We will now proceed to the configuration with the Integration Directory (ID).

The StarterPack scenario involves 4 end-point systems:

1. R/3 system where BAPI_PO_CREATE is to be executed.
2. Legacy_Sales_System where a PO file is to be created.
3. Oracle_Sales_System where PO info is to be inserted into an Oracle table.
4. MQSeries_Sales_System where the PO total info is to be placed in a queue.

There are many ways to configure the Integration Directory. There is a top-down or the bottom-up approach.

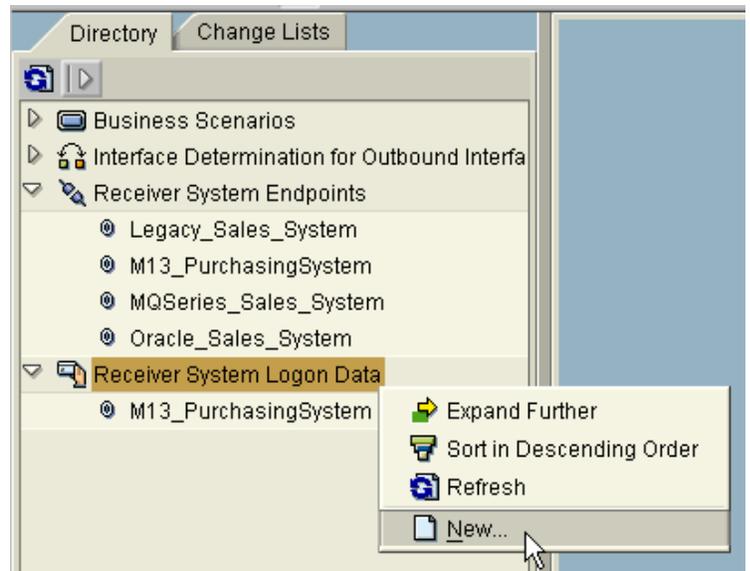
Here, we will use the bottom-up approach, which means we will configure all the required steps before the scenarion itself. So, during the final scenarion configuration, if there are any missing configurations, then we know we've made a mistake and can identify it quickly.

- 14. The BAPI_PO_CREATE will be executed on the M13 system. JCO is used to execute the BAPI.

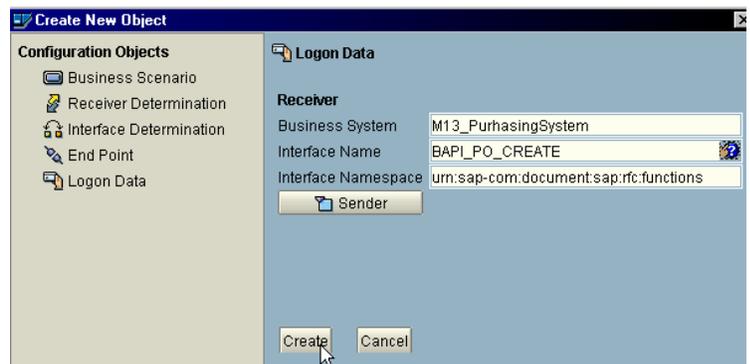
Therefore JCO will require logon information to connect to M13.

A "Receiver System Logon Data" will need to be created for M13.

To create logon information, right-click on "Receiver System Logon Data", and select "New":



Enter the following on the creation screen:



Enter logon info:

Edit Logon Data		Status
Sender	*	*
Receiver	M13_PurchasingSystem	BAPI_PO_CREATE
Description	Authorization for logon	
Authentication Procedure	Password-Based	
SAP Client	800	
User Name	i007286	
Password	*****	
Logon Language	English	

15. Create "Receiver System Endpoint".

Create the endpoint configurations based on the information on the right.

To create a receiver system, right-click on "Receiver System Endpoint", and select "New". The following screens will appear, enter the values as indicated.

Receiver: Legacy_Sales_System

Interface Name: OrderItem2_inbound

URL: HTTP://tsphl848.phl.sap.corp:8210/file/receiver

Edit End Point			Status
Sender	*	*	Active
Receiver	Legacy_Sales_System	OrderItem2_inbound	urn:sa
Description			
Endpoint Type	XI Connectivity		
Target			
Addressing Mode	URL Addressing		
URL	HTTP://tsphl848.phl.sap.corp:8210/file/receiver		
Logon			
<input checked="" type="radio"/> Anonymous Logon <input type="radio"/> Apply Logon Data			

Receiver: Oracle_Sales_System

Interface Name: OrderItem_inbound

URL: HTTP://tsphl848.phl.sap.corp:8242/db/receiver

Edit End Point		Status
Business System	Interface Name	
Sender	*	*
Receiver	Oracle_Sales_System	Orderitem_inbound
Description		Oracle database
Endpoint Type: XI Connectivity		
Target		
Addressing Mode	URL Addressing	
URL	HTTP://tsphl848.phl.sap.corp:8242/db/receiver	
Logon		
<input checked="" type="radio"/> Anonymous Logon <input type="radio"/> Apply Logon Data		

Receiver: MQSeries_Sales_System

Interface Name: OrderTotal_inbound

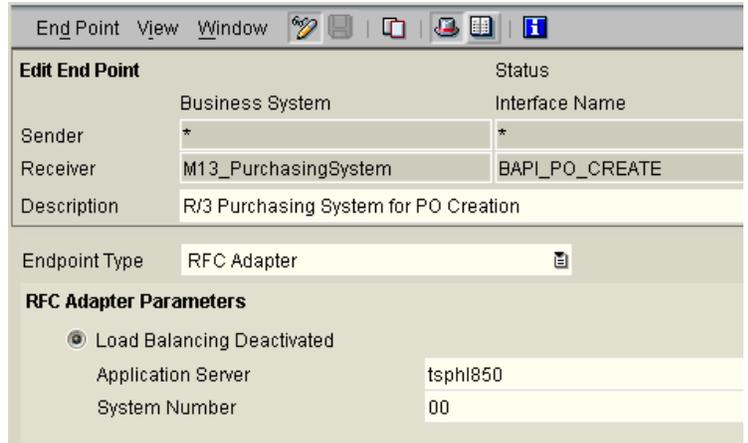
URL: HTTP://tsphl848.phl.sap.corp:8261/JMS

Edit End Point		Status
Business System	Interface Name	
Sender	*	*
Receiver	MQSeries_Sales_System	OrderTotal_inbound
Description		MQSeries queue
Endpoint Type: XI Connectivity		
Target		
Addressing Mode	URL Addressing	
URL	HTTP://tsphl848.phl.sap.corp:8261/JMS	
Logon		
<input checked="" type="radio"/> Anonymous Logon <input type="radio"/> Apply Logon Data		

Receiver: M13_PurchasingSystem

Interface Name: BAPI_PO_CREATE

URL: HTTP://tsphl848.phl.sap.corp:8261/JMS



16. Create Interface Determination for Outbound Interface.

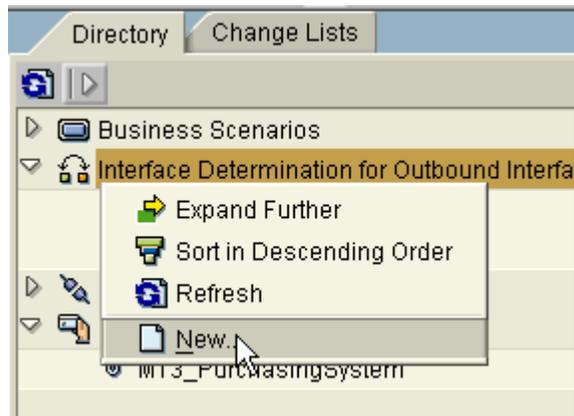
For the StarterPack scenarios, we have the following outbound interface determinations:

1. Legacy_Purchasing_System / OrderData_outbound → M13_PurchasingSystem / BAPI_PO_CREATE
(mapping: POData_BAPI)
2. M13_PurchasingSystem / ORDERS.ORDER505 → Legacy_Sales_System System / OrderItem2_inbound
(mapping: Orders05_Item2)
3. M13_PurchasingSystem / ORDERS.ORDER505 → Oracle_Sales_System / OrderItem_inbound
(mapping: Orders05_Item)
4. M13_PurchasingSystem / ORDERS.ORDER505 → MQSeries_Sales_System System / OrderTotal_inbound
(mapping: Orders05_Total)

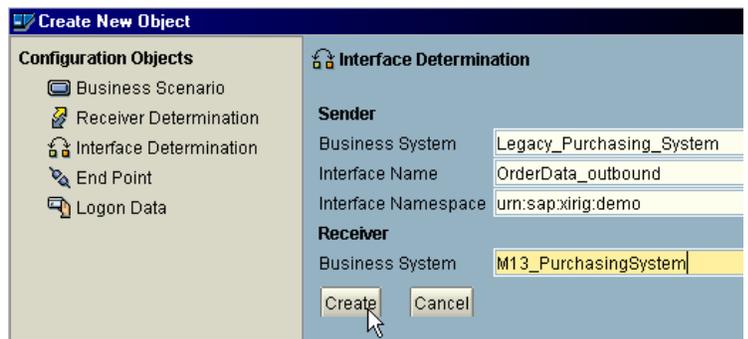
Each one of them has a different mapping requirement.

Use the following sample to create each of the Interface Determinations.

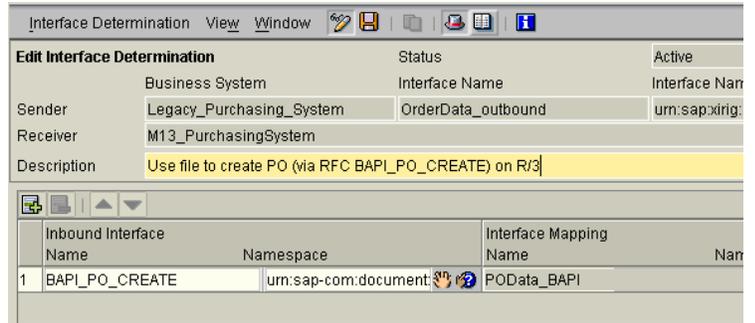
To create, right-click on the “Interface Determination for Outbound Interface”, and select “New”:



Enter the system and interface name, per the information on the left side:



Enter the Inbound Interface name and select the mapping to be used:



Repeat this process for the remaining 3 determinations.

17. Create the Business Scenarios.

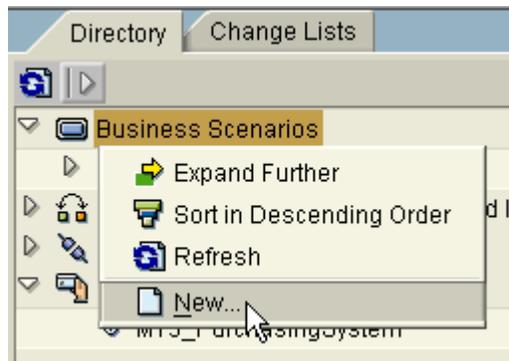
If a business scenario name does not exist, then one is to be created. This is an arbitrary name, which should describe the scenarios.

The Business Scenario puts everything together. All of the following information are considered:

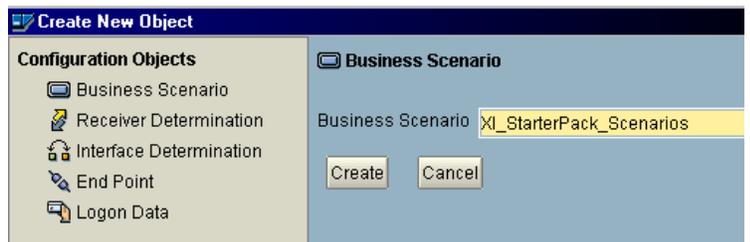
1. The sender and outbound interface.
2. The receiver and inbound interface.
3. The end point type to be used, e.g. RFC or URL. And, whether logon info is required.
4. The interface mapping.
5. Whether more than 1 receiver is to be used, and under what conditions (by using XPath statements).

Since we have entered all of the required information previously, the configurations should be filled in automatically when “refresh” is used. If there is any information not filled in or in error, then we know we have not configured one or more of the previous steps correctly. In this case, we should go back and trace the problem.

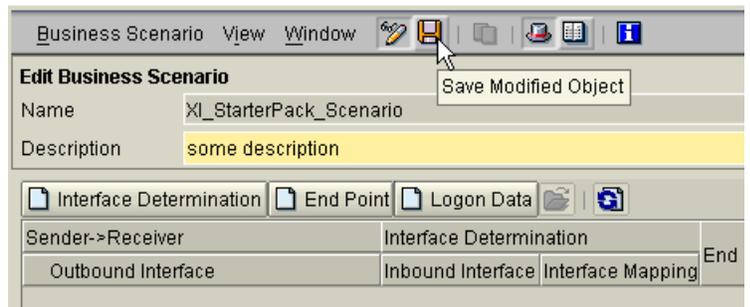
To create a scenario name, if one does not already exist, right-click on “Business Scenarios”, and select “New”:



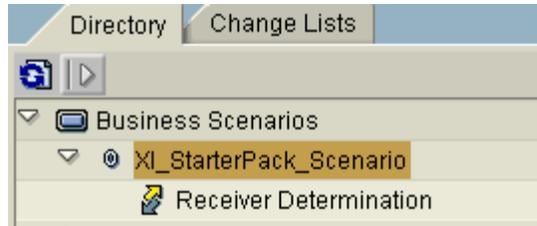
Enter a name, and select “Create”:



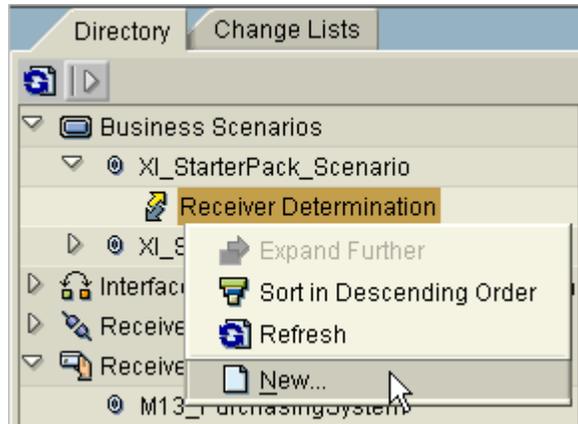
Click “Save” on the next screen:



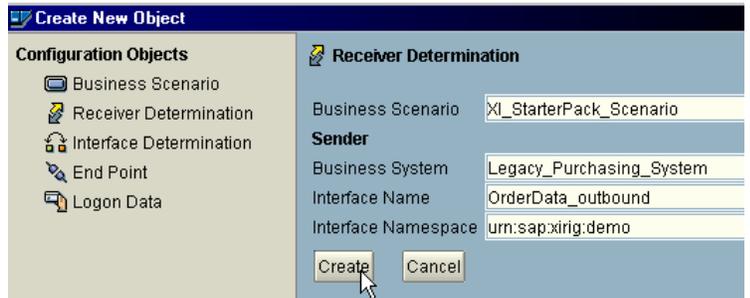
After saving, the following view should appear:



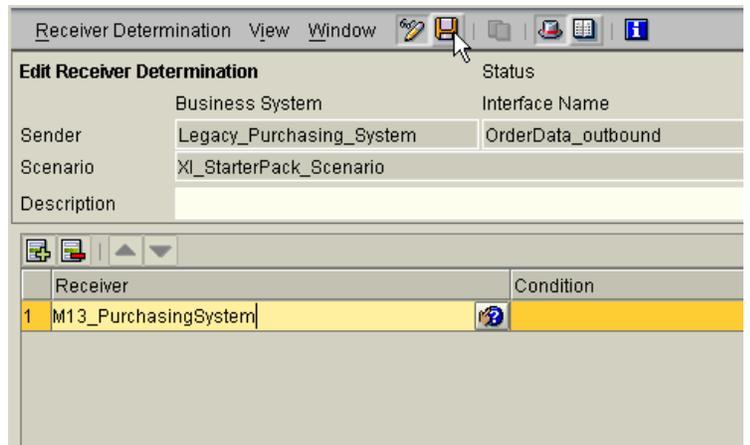
Right-click on “Receiver Determination” and select “New”:



Enter the sender and interface name:

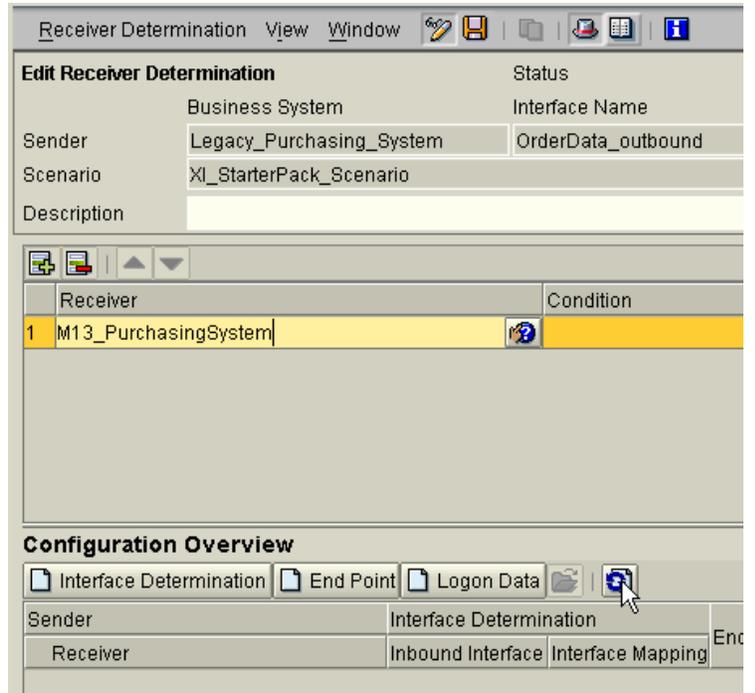


On the next screen, enter the receiver system name and save:

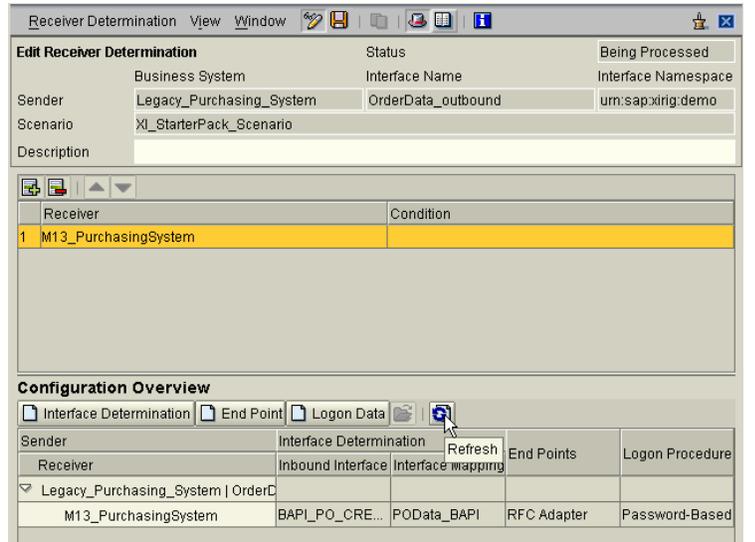


After saving, highlight (by clicking on the receiver

name), and click “Refresh”:



At this point, all the information in the “Configuration Overview” should be populated. If not, then there is an error from one of the previous steps.

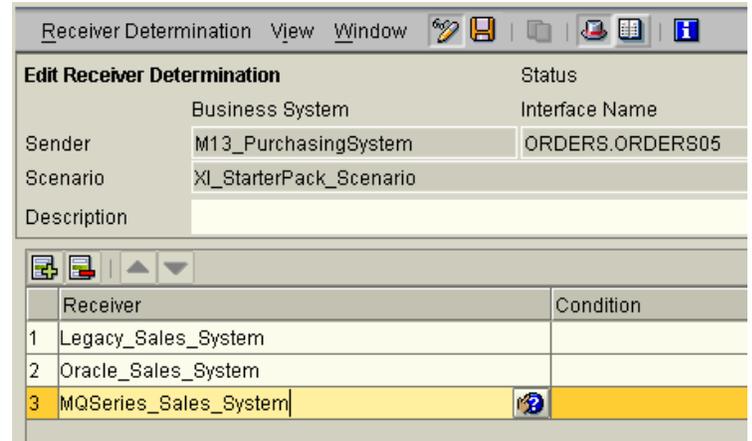


18. Create Receiver Determination for the IDoc.

For the receiver determination involving the IDoc, please pay special attention it. The IDoc will be sent to 3 different business systems, and 2 of those systems have conditions attached to them. The condition examines the material number of the IDoc. In addition, since they involve the same sending system and outbound interface, we can enter them on the same screen at the same time.

Create Receiver Determinations as the previous step until the following screen appears (during configuration, use sender = M13_PurchasingSystem, interface = ORDERS.ORDERS05):

Enter all the receiver names as shown.

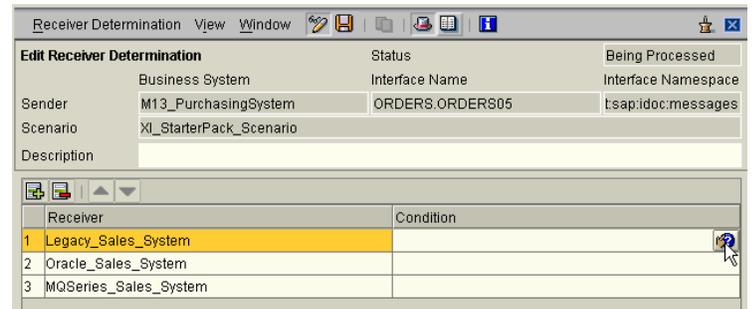


19. Create conditions for the receivers.

XPath conditions must be enter to dictate if a message is to be passed to the specific receiver.

Click on the "Condition" entry box of the Legacy_Sales_System "Receiver",.

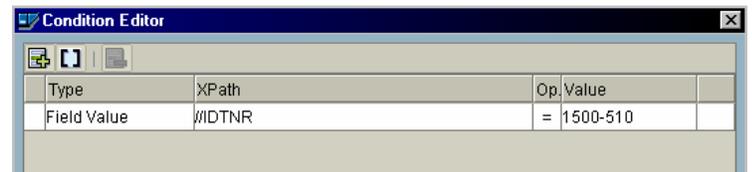
Then, click  on the same row.



The following screen will appear to provide XPath entries: (click on  to add an entry)

The field value = "//IDTNR" - material number

The value = "1500-510"



Do the same for the Oracle_Sales_System.

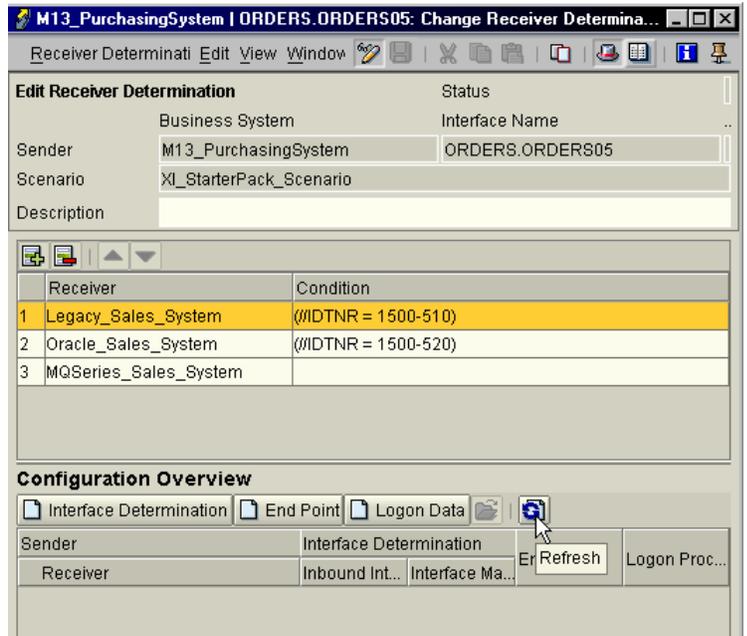
After the entries, the screen should look like the following:



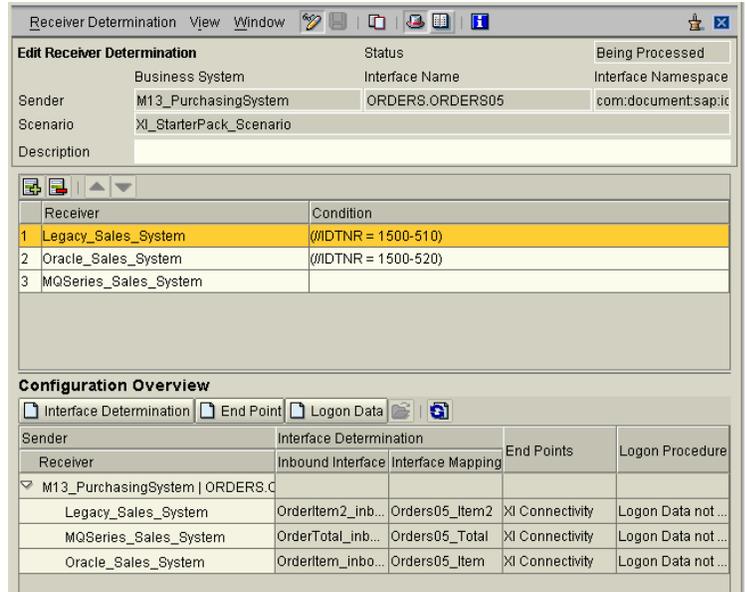
Click "Save", .

Click and highlight a receiver line, e.g. Legacy_Sales_System.

Click "Refresh", .



All the values in the "Configuration Overview" should be filled automatically. If not, then there is an error due to one of the previous steps.



- 20. The XI Integration Builder’s configurations are complete.

The Technical Adapter configurations will need to be done to complete our StarterPack configurations.

It is possible to have performed some of the adapter configurations already. As mentioned before, we may have used it to send XML message to XI. The purpose was to look at the payload to help with our data type creation process.

The direction of the interface for the adapters must be from the XI’s perspective. For example, sending a file from the Legacy_Purchasing_System to XI, we must configure an inbound file adapter.

After making the configuration entries, the adapter can be started by clicking on the “Restart” button.

The configurations outlined below are entered on the Technical Adapter on the XI server. All of

the adapters, except the RFC adapter, can also be installed elsewhere, i.e. your laptop. Keep in mind, if using a local adapter (i.e. on your laptop), the endpoint configuration in the Integration Directory must use the IP address of your PC, or else an error will most likely occur.

21. Inbound file adapter for the Legacy_Purchasing_System.

These file adapter configurations is to retrieve file(s) from a file directory and reformats the data into XML and send it to XI for processing.

Please reference the Adapter Engine documentation for details and meanings of the property keys.

The data file contains PO data in a complex structure, with header and item information. The structure contains 1 header record and undetermined number of item records. A "key" column is used to determine the type of record being referenced.

If the key = header, then it is header record. If the key = item, then it is an item record.

The header record contains the following information, delimited by a ";" separator:

1. key (always header)
2. DocDate
3. DocType
4. PurchOrg
5. PurGroup
6. VendorId

```
## file adapter java class
classname=com.sap.aii.messaging.adapter.ModuleFile2
XMB

mode=FILE2XMBWITHSTRUCTURECONVERSION

## Integration Engine address and document settings
(example, see docu)
XMB.TargetURL=http://tsphl848.phl.sap.corp:8000/sap
/xi/engine?type=entry

XMB.SenderBusinessSystem=Legacy_Purchasing_System
XMB.SenderInterfaceName=OrderData_outbound
XMB.SenderInterfaceNamespace=urn:sap:xirig:demo

XMB.ContentKind=T
XMB.ContentType=text/plain
XMB.QualityOfService=EO

XMB.Client=800
XMB.Language=en
XMB.User=xiappluser
XMB.Password=xipass

##File Adapter specific parameters (example, see
docu)
file.sourceDir=F:/data/File_send
file.sourceFilename=POTest2.txt
file.processingMode=test
file.archiveDir=F:/data/File_archive
file.pollInterval=600000000

##xml conversion properties if mode
FILE2XMBWITHROWCONVERSION specified (example, see
docu)
xml.fieldSeparator=;

xml.recordsetStructure=header,1,item,*
xml.keyFieldName=key
xml.keyFieldType=CaseInsensitiveString
xml.documentName=PO_Data_msgtype

xml.header.fieldNames=key,DocDate,DocType,PurchOrg,
PurGroup, Vendor
xml.header.keyFieldValue=header
xml.header.keyFieldInStructure=ignore
xml.header.fieldSeparator=;

xml.item.fieldNames=key,ItemNo,Material,Plant,Deliv
Date,Quantity
xml.item.keyFieldValue=item
```

The item record contains the following information, delimited by a “;” separator:

1. key (always item)
2. ItemNo
3. Material number
4. Plant
5. Delivery date
6. Quantity

```
xml.item.keyFieldInStructure=ignore
xml.item.fieldSeparator=;
```

22. Outbound file adapter for the Legacy_Sales_System.

These file adapter configurations is to send data to a file directory from XI and reformats the XML data into a file format.

Please reference the Adapter Engine documentation for details and meanings of the property keys.

The [httpPort](#) and [httpService](#) values must match that in **step 15's** URL information for the Legacy_Sales_System.

```
## File adapter java class
classname=com.sap.aii.messaging.adapter.ModuleXMB2F
ile

#mode=XMB2FILE
mode=XMB2FILEWITHCONVERSION

##Address for XMB endpoint
XMB.httpPort=8210
XMB.httpService=/file/receiver

##File Adapter specific parameters
file.createDir=1
file.targetDir=F:/data/File_receive
file.targetFilename=POdata.txt
file.writeMode=addCounter
file.counterMode=immediately
file.counterSeparator=_
file.counterFormat=00000
file.counterStep=1

##xml conversion properties if mode
XMB2FILEWITHCONVERSION is specified
xml.addHeaderLine=0
xml.headerLine=
xml.fieldSeparator=;
```

23. Outbound JDBC adapter for the Oracle_Sales_System.

These JDBC adapter configurations is to insert row(s) to a table in Oracle from XI.

Please reference the Adapter Engine documentation for details and meanings of the property keys.

The [httpPort](#) and [httpService](#) values must match that in **step 15's**

```
## jdbc adapter java class
classname=com.sap.aii.messaging.adapter.ModuleXMB2D
B
#transformClass=com.sap.aii.messaging.adapter.trans
.

mode=XMB2DB

##Address for XMB endpoint
XMB.httpPort=8242
XMB.httpService=/db/receiver
XMB.QualityOfService=EO

##DB Adapter specific parameters
db.jdbcDriver=oracle.jdbc.driver.OracleDriver
db.connectionURL=jdbc:oracle:thin:system/exchange@t
sph1848.phl.sap.corp:1521:XDB
db.table=orders
db.exactlyOnceErrorInPendingState=IGNORE
```

URL information for the Oracle_Sales_System.

The Oracle JDBC driver must be include in the classpath of the Technical Adapter.

From the configuration:

1. The Oracle database is on the server tsphl848.phl.sap.corp.
2. The Oracle instance is [XDB](#).
3. The userid / password is [system / exchange](#).
4. The Oracle table name is [orders](#).

24. Outbound JMS adapter for the MQSeries_Sales_System.

These JMS adapter configurations is to put data into the MQSeries queue from XI.

Please reference the Adapter Engine documentation for details and meanings of the property keys.

The [httpPort](#) and [httpService](#) values must match that in **step 15's** URL information for the Oracle_Sales_System.

The IBM MQSeries driver must be include in the classpath of the Technical Adapter.

From the configuration:

1. The MQSeries resides on the server tsphl848.phl.sap.corp.
2. The queue manager is [QM_tsphl848](#).
3. The queue name is [JMS_XI_StarterPack](#).

```
# JMS-Adapter Configurarion
classname=com.sap.aii.messaging.adapter.ModuleTrans
port2JMS
JMSMessage.type=TextMessage
#
# TransportMessage Type can be set to
TransportMessage of XMBMessage. It is used for
sending and receiving Transport and XMBMessages
#TransportMessage.type=TransportMessage
TransportMessage.type=XMBMessage
```

```
transformClass=com.sap.aii.messaging.adapter.trans.
JMSSinglePayloadBinding
```

```
XMB.httpPort=8261
XMB.httpService=/JMS
```

```
JMS.QueueConnectionFactoryImpl.classname=com.ibm.mq
.jms.MQQueueConnectionFactory
JMS.QueueConnectionFactoryImpl.method.setHostName=j
ava.lang.String tsphl848.phl.sap.corp
JMS.QueueConnectionFactoryImpl.method.setChannel=ja
va.lang.String JAVA.CHANNEL
JMS.QueueConnectionFactoryImpl.method.setTransportT
ype=java.lang.Integer
{com.ibm.mq.jms.JMSC.MQJMS_TP_CLIENT_MQ_TCPIP}
JMS.QueueConnectionFactoryImpl.method.setQueueManag
er=java.lang.String QM_tsphl848
JMS.QueueImpl.classname= com.ibm.mq.jms.MQQueue
JMS.QueueImpl.constructor=java.lang.String
JMS_XI_StarterPack
JMS.QueueImpl.method.setTargetClient=java.lang.Inte
ger {com.ibm.mq.jms.JMSC.MQJMS_CLIENT_NONJMS_MQ}
```

25. RFC adapter configurations include both inbound and outbound interfaces.

The RFC adapter can only be configured on the Adapter Engine on the XI server. Local Adapter Engine (i.e. the one running on your laptop) will not have the RFC adapter.

In this example, please keep in mind, M13 is the R/3 application server, and XD1 is the XI server.

The values marked in **blue** are for the M13 system, where the BAPI is to be executed.

The values marked in **dark red** are for the XI system.

The progid, `Rfc2XmbServiceM13`, must be define in M13 using SM59. This name must match that in the RFC adapter configuration. The name can be anything, but recommended to be `Rfc2CmbService`. The reason M13 was added to the suffix is because multiple XI server are using the M13 system as the application server, and each must have a unique progid. The SM59 name is recommended to be something like `XI_RFC`. This is the “destination” name to use during for an RFC call sent to XI.

The progid, `Xmb2RfcService`, must be define in XD1 using SM59. The SM59 name must be `AI_RFCADAPTER_JCOSERVER`. This should have already been configured during the XI installation process. If not, proceed to create one. . Since XI is a Unicode system, the Unicode option must be selected in the SM59 configuration.

```
#
# RFC ADAPTER CLASSNAME FOR ADAPTER ENGINE
#
classname=com.sap.aii.rfcadapter.core.RfcAdapterModule
#
# RFC ADAPTER CONFIGURATION PROPERTIES
#
# Wed Feb 22 14:46:00 CET 2002
#
# 1. GENERAL SECTION
#
# 1.1 trace settings
#
RfcAdapter.trace=5
RfcAdapter.traceFile=./LogFiles/Adapter_rfc.log
#
# 1.2 connectivity settings
#
RfcAdapter.registerInbound=sourceBackend
RfcAdapter.accessInbound=toXmb
RfcAdapter.registerOutbound=fromXmb
RfcAdapter.accessOutbound=targetBackend
#
# 2. SECTION FOR REGISTRATION TO CLIENT APPLICATION SYSTEMS
# (requires callback service user for repository access)
#
RfcAdapter.sourceBackend.host=tsphl850
RfcAdapter.sourceBackend.sysnr=00
RfcAdapter.sourceBackend.progid=Rfc2XmbServiceM13
RfcAdapter.sourceBackend.poolsize=2
RfcAdapter.sourceBackend.unicode=0
RfcAdapter.sourceBackend.client=800
RfcAdapter.sourceBackend.user=krajan
RfcAdapter.sourceBackend.passwd=nachu01
RfcAdapter.sourceBackend.lang=EN
#RfcAdapter.sourceBackend.trace=0
#
# 3. SECTION FOR ACCESS TO XMB INBOUND RFC SERVICE
# (only one entry possible)
#
RfcAdapter.toXmb.host=tsphl848.phl.sap.corp
RfcAdapter.toXmb.sysnr=00
RfcAdapter.toXmb.poolsize=3
RfcAdapter.toXmb.client=800
RfcAdapter.toXmb.user=XIAPPLUSER
RfcAdapter.toXmb.passwd=XIPASS
RfcAdapter.toXmb.lang=EN
#RfcAdapter.toXmb.trace=1
#
# 4. SECTION FOR REGISTRATION OF XMB OUTBOUND RFC SERVICE
# (only one entry possible)
#
RfcAdapter.fromXmb.host=tsphl848.phl.sap.corp
RfcAdapter.fromXmb.sysnr=00
RfcAdapter.fromXmb.progid=Xmb2RfcService
RfcAdapter.fromXmb.poolsize=2
RfcAdapter.fromXmb.unicode=1
#RfcAdapter.fromXmb.trace=1
#
# 5. SECTION FOR ACCESS TO SERVER APPLICATION SYSTEMS
# (pre-allocated connection pools via service user)
#
RfcAdapter.targetBackend.host=tsphl850
```

```
RfcAdapter.targetBackend.sysnr=00
RfcAdapter.targetBackend.poolsize=3
RfcAdapter.targetBackend.client=800
RfcAdapter.targetBackend.user=krajan
RfcAdapter.targetBackend.passwd=nachu01
RfcAdapter.targetBackend.lang=EN
RfcAdapter.targetBackend.trace=0
#
# END OF RFC ADAPTER CONFIGURATION
#
```

26. The StarterPack scenario configuration is now complete.

Testing can be initiated using the file adapter, Legacy_Purchasing_System.

4 Additional Information

CONNECTION TO/FROM OTHER R/3 SYSTEM

The RFC adapter as shown in #25 above is only one of the connection configuration. The other R/3 connections are:

1. Connection from the Integration Repository to R/3 to import meta data for RFC and IDoc's. The connection configuration requirements are as follow:
 - a. An entry must be present in the services file of the operating system where XI server is installed. This entry must be in the following format:

```
sapmsXXX    36nn/tcp
```

where XXX is the system id and nn is the system number.
Using M13, which has a sysnr of '00', as an example:

```
sapmsM13    3600/tcp
```
 - b. The group name must be valid. The group name can be checked using the transaction SMLG on the R/3 system where the RFC/IDoc resides. If the group name does not exist, then create one.
 - c. The group name must be in UPPERCASE, or else XI will not be able to find it.
 - d. If this entry is the last line in the file, then another blank line must be added after it.
2. Connection configuration to send IDoc. The ALE name must be properly entered in the SLD during the XI Business system configuration.
3. Connection configuration to receive IDoc. Configuration must be done on the XI system using the transaction IDX1.