How to...

Integrate BW to XI
(for message based inbound processing of data)

BUSINESS INFORMATION WAREHOUSE

Applicable Releases:  BW 3.5
                     XI 3.0
                     (NetWeaver 04)

January 2004

SAP (SAP America, Inc. and SAP AG) 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 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 does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP 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.

mySAP BI “How-To” papers are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using mySAP BI. Should you wish to receive additional information, clarification or support, please refer to SAP Professional Services (Consulting/Remote Consulting).
Business Scenario

This document describes how to send data from XI to BW with full Quality of Service (Exactly Once In Order). The solution is based on BW 3.5 and XI 3.0 which are both part of NetWeaver 04.

Introduction

Since BW 3.0B data can be transferred to the BW via “Push” into the BW Delta Queue. From there it is requested with the same mechanisms that are used for the extraction from R/3 source systems.

However, the Web Service and the generic SOAP/RFC Service do not support “Exactly Once” transfer of data as no transaction ID is handled in the framework. Therefore data might be transferred several times which is not a valid for all delta options that can be used in BW.

The following picture shows different options of the data flow.

XI 3.0 provides a variety of ways to access data from different sources. There are adapters to connect files, databases, messaging systems, Web Services. With R/3 systems (3.1h and higher), IDoc and RFC can be used.
The standard communication channel for SAP systems is the ABAP Proxy which is available for Web AS 6.20 and higher.

The Proxy communication supports the full Quality of Service (“Exactly Once In Order”) between XI and BW which other adapters do not:

- The RFC-Adapter does not support “In Order” but only “Exactly Once” (so-called tRFC)
- The SOAP-Adapter supports “In Order”, i.e. the order of the sender is maintained. However “Exactly Once” cannot be implemented in this scenario in conjunction with the receiver (see above).

SAP recommends to use the ABAP Proxy communication. Only the ABAP Proxy implementation is described in this paper.

The Result

The XML data is stored in the delta queue of the BW myself system (transaction RSA7 in the BW) and, consequently, integrated into the BW staging process.
The Step-By-Step Solution

Remarks and Prerequisites in the BW system

1. Remarks
   The proposed solution is based on the SOAP DataSource which is a feature since BW 30B. The usage is described in a different How-To Paper “Send XML Data to BW” which you find under <https://service.sap.com/~sapidb/011000358700011142902001E/HOWTOSENDXMLDATATOBW.DOC>.

2. Prerequisites
   The BW is integrated to the XI landscape.
   If this is not yet the case, follow the configuration guide of XI which you find here: SAP Service Marketplace (http://service.sap.com/instguides)
   Hint: As only the proxy communication is used, there is no configuration of IDOC or RFC communication necessary.

3. Perform the steps 1 to 20 ("…create a new InfoPackage …") from the How-To Paper “Send XML Data to BW"
   Results:
   1: A DataSource in the myself system has been generated
   Name: 6ASENDXMLDATATOBW (in general: 6A<File-DataSource>)
   2: You find an RFC-enabled function module which will be used to perform the inbound processing of data in the following steps.
   Name: /BI0/QI6ASENDXMLDATATOBW_RFC (in general: /BI0/QI<datasource>_RFC)
Tasks in the Integration Builder (XI) and BW Proxy Framework

Repository and Proxy Framework

4. Open the Repository (Design) of the XI Integration Builder. You may start the application by executing transaction SXMB_IFR in the BW system. (Further information can be found in the XI documentation - > Design and Configuration Time - > Design.)

Choose your Software Component in the tree and open it with double-click: here we use: XITestComponent

Enter the system data of the BW system that you want to connect.

Note: If any problems occur when executing the Integration Builder check the client software installation (follow note 580351)

5. Choose “Import RFC/IDOC” in the context menu of “Imported Objects” and enter your user data to logon to the BW system. Select the function module /BI0/Q16ASENDXMLDATATOBW_RFC

Press “Continue”

Press “Start”

Choose “Interface Objects” and then “Message Interface”
Create a Message Interface there by choosing “New” from the context menu

Press “Create”

7. Choose the settings like this:
Direction: Inbound
Mode: Asynchronous
Input Message Type: DataToBW

Select the message type by executing the value help
8. In the window that is opened you select the function module
   `/BI0/QI6ASENDXMLDATATOBW_RFC`
   from "Imported Objects"
   Save the message interface

9. Activate the Change List which you find behind the Tab "Changelist"
10. Create the proxy in the BW system:
Start transaction SPROXY and Refresh the tree

Choose a package (e.g. $tmp)
Choose a prefix (we use “Z” here)

Activate the Proxy

11. Navigate to the implementing class ZCL_DATA_TO_BW by double clicking

Navigate to the method “EXECUTEASYNCHRONOUS” by double-clicking.

Choose the change mode
12. Insert the implementation of the method based on this sample coding and change the highlighted elements.
   l_t_data: tables parameter for data transfer to the function module (look up the type in the tables parameter of the function module)
   l_s_data: structure with the same structure as l_t_data.

   Enter the name of the function module to the one you use (here: '/BI0/QI6ASENDXMLDATATOBW_RFC')

   Enter the literal for the parameter "datasource" which is here: '6ASENDXMLDATATOBW' – Doing so the parameter "datasource" does not have to be entered in the payload of the XML data later.

   METHOD zii_data_to_bw~execute_asynchronous.
   DATA: l_s_data TYPE zoxab50111,
      l_t_data TYPE TABLE OF zoxab50111.
   FIELD-SYMBOLS <l_line> LIKE LINE OF input-data-item.
   LOOP AT input-data-item ASSIGNING <l_line>.
      MOVE-CORRESPONDING <l_line> TO l_s_data.
      APPEND l_s_data TO l_t_data.
   ENDLOOP.
   CALL FUNCTION '/BI0/QI6ASENDXMLDATATOBW_RFC'
      EXPORTING
         datasource     = '6ASENDXMLDATATOBW'
      TABLES
         data           = l_t_data
      EXCEPTIONS
         internal_error = 1
         OTHERS         = 2.
      IF sy-subrc NE 0.
         ROLLBACK WORK.
      ENDIF.
   ENDMETHOD.

13. Save and activate the method and all class/interface components

14. Result

   The BW is now ready to receive data from the Integration Server via the Proxy Framework and the implemented Proxy. Data which is running through XI may now be transferred to the BW system according to routings and mappings that still have to be created.

Verification Szenario

XI Configuration

15. Open the Directory (Configuration):
   Execute transaction SXMB_IFR (in the BW or XI system) and select “Directory” under SAP Integration Builder
16. Select the tab “Scenarios” and choose “New” from the context menu

Enter “DataToBW” and a description

Press Create

Save the Scenario

17. Press “Create New Object” under Collaboration Profile Objects (tab Scenario Objects)

Choose Service and enter the name of your BW system (Business System which is maintained in the System Landscape Directory)

Here the BW system is: AB5_003

Press “Create”

18. Add the inbound interface “DataToBW” in the namespace you use (here: “http://customer.com/xi/BW”) on the tab Receiver

Press Save
19. Create a communication channel for the BW system
Press “Create New Object” which you find under “Communication Channels” of the Service
Enter the name “DataToBW” and a description
Press “Create”

20. Maintain the communication data:
Enter (or choose respectively)
Adapter type: XI
Message Protokoll: XI 3.0
Adressing Mode: URL address
Host, Port and Path of the XI engine
Authentication data
Save the Communication Channel
Further information can be found in the XI documentation.

21. Create a “virtual” sender for which is used later in the verification
Press “Create New Object” on the Scenario object under
Collaboration Profile Objects (tab Scenario Objects)
Select: “Service”
22. Specify the Service name and add a description

Here: “XXX_000”
Press “Create”
Save the service


There is no “Communication Channel” required

24. Create the “Receiver Determination” from the Scenario Tab (under “Logical Routing and Collaboration Agreement Objects”)

Enter the Sender, Interface and Namespace and add a description
Press “Create”. 
25. Enter the BW system under "Configured Receivers", here AB5_003
Press “Save”

26. Create an "Interface Determination" from the Scenario Tab
Enter the sender service, the interface the namespace and the receiver system
Press “Create” and save

27. Enter the Interface name under “Configured Inbound Interfaces”
Press Save
28. Create a “Receiver Agreement” from the Scenario Tab

Enter the Sender Service, the Receiver Service, the interface, the namespace and add a description

29. Choose the Receiver Communication Channel, here: “DataToBW”

Save the Receiver Agreement

30. Activate your Change list

Hint: You may check the Scenario element on the Scenario Tab by executing “Check Scenario” in the menu “Scenario”.

Result: Messages can now been sent from the system XXX_000 via the XI system to the BW system

Runtime: Send Data and transfer it to the BW

31. In order to verify the processing of the messages we will use the native http adapter of the integration server and send an XML file to the integration server. (It is the fastest and easiest way to verify the load with the http adapter.)

32. Activate the plain http adapter (/sap/xi/adapter_plain) on your XI Integration Server in transaction SICF if not yet active
33. Save the following source code of a verification program to your PC (e.g. to the file name DataToBW.htm).

Remark:
You may also use a different client written in Java or VBA if available.

Note: This program is for demonstration purposes only. It is not meant to be a delivered SAP product. Therefore SAP cannot give any CSN support concerning problems that might occur when using it.

```html
<html>
<head>
<title>Send XML Data to BW System</title>
<script language="javascript">
<!--
function SendData() {
    var mypath = document.myform.filename.value;
    var myescns = escape(document.myform.mynamespace.value);
    var mycall = 'http://'
        + document.myform.myhost.value + ':'
        + document.myform.myport.value + '/sap/xi/adapter_plain?bs='
        + document.myform.mysystem.value + '&namespace='
        + myescns + '&interface='
        + document.myform.myinterface.value + '&qos='
        + document.myform.myqos.value;
    var xmlstream = new ActiveXObject("ADODB.Stream");
    xmlstream.Mode = 3;
    // 1=read
    xmlstream.Open();
    xmlstream.Type = 1;
    //
    3=read/write
    xmlstream.Open();
    xmlstream.Type = 1;
    //
    1=adTypeBinary 2=adTypeText
    xmlstream.LoadFromFile(mypath);
    var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    xmlhttp.Open("POST",mycall,false);
    xmlhttp.setRequestHeader("Content-Length",xmlstream.Size);
    xmlhttp.setRequestHeader("Content-Type","text/xml");
    xmlhttp.send(xmlstream.Read(xmlstream.Size));
    XICall.innerText = mycall;
    XIAnswer.innerHTML = xmlhttp.responseText;
}
//-->
</script>
</head>
<body>
<form name="myform">
<p>Path to file: </p>
<input type="file" name="filename" size=50 maxlength=80
value="c:\temp\demo.xml">
<p>XI adapter parameters: </p>
<table border=0>
<tr><td>hostname:</td><td>
<input type="text" name="myhost" size=50 maxlength=100
value="PWDF0321.wdf.sap.corp"></td></tr>
<tr><td>port:</td><td>
<input type="text" name="myport" size=50 maxlength=100
value=""/>
</form>
</body>
</html>
```
34. Save the following XML data to a file on your PC (e.g. to the filename data.xml)

Change the highlighted elements to the actual names:
The interface name of the Sender Interface has to be edited in the start and end tag, here "DataToBW" is used.
The namespace (after "ns1:" has to be edited to the one you use.
(Here "http://customer.com/xi/BW" ist used)

The elements within the "item" node must correspond to the field structure of your DataSource.

```xml
<?xml version="1.0" encoding="utf-8"?>
<ns1:DataToBW xmlns:ns1="http://customer.com/xi/BW" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DATA>
    <item>
      <CALDAY>20030905</CALDAY>
      <MATERIAL>4711</MATERIAL>
      <AMOUNT>100</AMOUNT>
      <CURRENCY>EUR</CURRENCY>
    </item>
    <item>
      <CALDAY>20030917</CALDAY>
      <MATERIAL>4712</MATERIAL>
      <AMOUNT>200</AMOUNT>
      <CURRENCY>EUR</CURRENCY>
    </item>
  </DATA>
</ns1:DataToBW>
```
35. Open the file DataToBW.htm in an Internet Explorer 5.1 or higher (The used ActiveX Objects will probably not work in different or older web browsers).

Enter the server name of your Integration Server, the port of the http services, the name of the business system maintained above, the namespace you are working in, the interface name “DataToBW” the quality of service “EO” (for asynchronous processing)

The XI documentation (Runtime -> Adapters -> Plain http adapter) provides more detailed information about the adapter.

36. Browse to your XML file (e.g. “data.xml”) containing the demo records Send the records by pushing the “Send Records” button.

You find the call to the XI plain http adapter below the button.

Check result (error search)

37. Open the transaction RSA7 in the BW system. Use “Display Data Entries” to check the data has been transferred correctly.
38. Error Search:
If the data does not occur in the
delta queue you may examine the
message flow in the transaction
SXMB_MONI on the XI system
and on the BW system.
Further details in the XI
documentation under “Runtime”

Comments

- In general it can be assumed that data is sent from a different SAP or JAVA system in a
  productive scenario. In this case the mapping (design) and the configuration part in XI has to
  be implemented in a different way: messages that are sent from a sender to a receiver would
  be duplicated in a “message split”, converted to a flat structure and sent to the BW system.

- The proposed method of loading data from XML files could be enhanced for scenarios where
  files of different kinds (based on different XML Schema Definitions) are transferred to a BW
  system using an interface mapping. However, it is in any case mandatory to transform it to a
  format which corresponds to the interface of the generated RFC function module. This
  always contains a “tables parameter” with flat structure (no deep parameters possible).

Summary

This instruction is based on the How-to “…Send XML Data to BW” which enables a BW system to
receive data records via the generic SOAP/RFC Service. There the XML DataSource is used to read
data from an XML file and send it to a BW.

After performing the first 20 steps of the How To “…Send XML Data to BW” there is a generated
DataSource within the BW source system (“MySelf” system) which is intended for uploading delta
records, and has an RFC function module for supplying the delta queue.

In this paper it is shown how to use the function module to enable a communication via a Proxy from
XI to BW for inbound messaging. Proxy communication is used as this supports the full Quality of
Service (Exactly Once in Order) between XI and BW for the processed data messages.

In the Design Area of the Integration Builder a message interface is defined which is then used to
create the Proxy in the BW system. The a template for the implementation has to be copied and
changed according to the actual Function Module which is used.

After that in the Configuration Area of the Integration Builder the system specific settings are done.
As this paper should be used separated from other scenarios a Virtual System is defined in the
Configuration.

For verification purpose an XML file is used to create a message which is sent to the Integration
Service through the HTTP-Adapter by means of the Internet Explorer. The data occurs in the Delta
Queue of the BW.