

SAP NetWeaver Process Integration 7.1



Demo Examples Exercise Cancel Flight Booking Using Integration Process (ccBPM)

Document Version 2.0 – July 2008



SAP AG

Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

© Copyright 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Documentation in SAP Community Network (SDN)

You can find this documentation in SAP Community Network (SDN) at

<https://www.sdn.sap.com/irj/sdn/soa-servicebus>
→ *Getting Started* → *Getting Started Documents* → *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking (With ccBPM)*.

1	Introduction	4
1.1	Prerequisites	4
1.2	Purpose	4
1.3	Business Description of the Process Integration Scenario	5
1.4	Overview of the Integration Process	5
1.5	Systems and Development Process	6
1.6	Users and Authorizations	6
2	Design	6
2.1	Calling the Enterprise Services Builder (Enterprise Services Repository)	6
2.2	Defining Namespaces	6
2.3	Enhancing the Process Integration Scenario	6
2.4	Creating an Abstract Service Interface	6
2.5	Creating the Operation mapping	7
2.6	Creating the Integration Process	7
2.6.1	Overview of the Integration Process	7
2.6.2	Developing the Integration Process in Detail	8
2.7	Defining Another Component View	9
3	Configuration	11
3.1	Creating the Integration Process Component	11
3.2	Configuring the Process Integration Scenario	11
3.2.1	Manual Editing After Generation	12
3.3	Configuration Settings in the SAP System	12
4	Execution	12
4.1	Checking the Result	13

1 Introduction

This document explains how to develop and test a simple cross-component process integration scenario using SAP NetWeaver Process Integration (SAP NetWeaver PI). The development guide takes you through the most important development steps when designing cross-component processes.

You will need approximately 1-2 hours to complete the example.

1.1 Prerequisites

This exercise is based on the first exercise *Canceling a Flight Booking*. You will enhance the proxy-to-proxy scenario described there. The proxy-to-IDoc scenario is not required.



You can find the description of the exercise *Canceling a Flight Booking* in SAP Community Network (SDN) at <https://www.sdn.sap.com/irj/sdn/soa-servicebus> → *Getting Started* → *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking*.

For more information regarding the definition of an integration process in the Enterprise Services Repository, see the chapter [Integration Processes](#) in SAP Help Portal.

For more information regarding SAP NetWeaver Process Integration (including the user interfaces of the Enterprise Services Repository and the Integration Directory), see the [SAP NetWeaver Process Integration Library](#) in SAP Help Portal.

The individual steps are not described in as much detail in this document as in the first exercise *Canceling a Flight Booking*. It is assumed that you are familiar with the exercise *Canceling a Flight Booking*.

However, for any topics that are not covered in the exercise *Canceling a Flight Booking* (for example, Creating an Integration Process), the steps are described in detail.

To be able to send alerting mails as a prerequisite SMTP must be configured in the PI system. For information concerning SMTP configuration check out the [SMTP Configuration Guide](#).

1.2 Purpose

Many scenarios require the sender (for example, an administrator) to be actively informed if message delivery fails during asynchronous communication.

This can be achieved by requesting an acknowledgment in the sender application. However, this can only be achieved by corresponding programming in the sender's application code.

Another way to implement the active notification of the sender is a technical acknowledgment, which adds an additional process on the Integration Server. This checks that the message has been delivered to the recipients. If there is no successful response, a mail is sent to a registered recipient within a predefined time by an exception on the Integration Server and the Alert Framework of the Application Server. It is not necessary to change the application code of the sender or receiver.

You must make the following additional enhancements in the Enterprise Services Repository:

- Create an abstract service interface whose message type references the existing message type.
- Create an operation mapping that references an existing message mapping.
- Create the integration process.
- Create another component view in the process integration scenario.

The changes in the Integration Directory are also relatively easy, since you simply use the new component view from the Enterprise Services Repository.

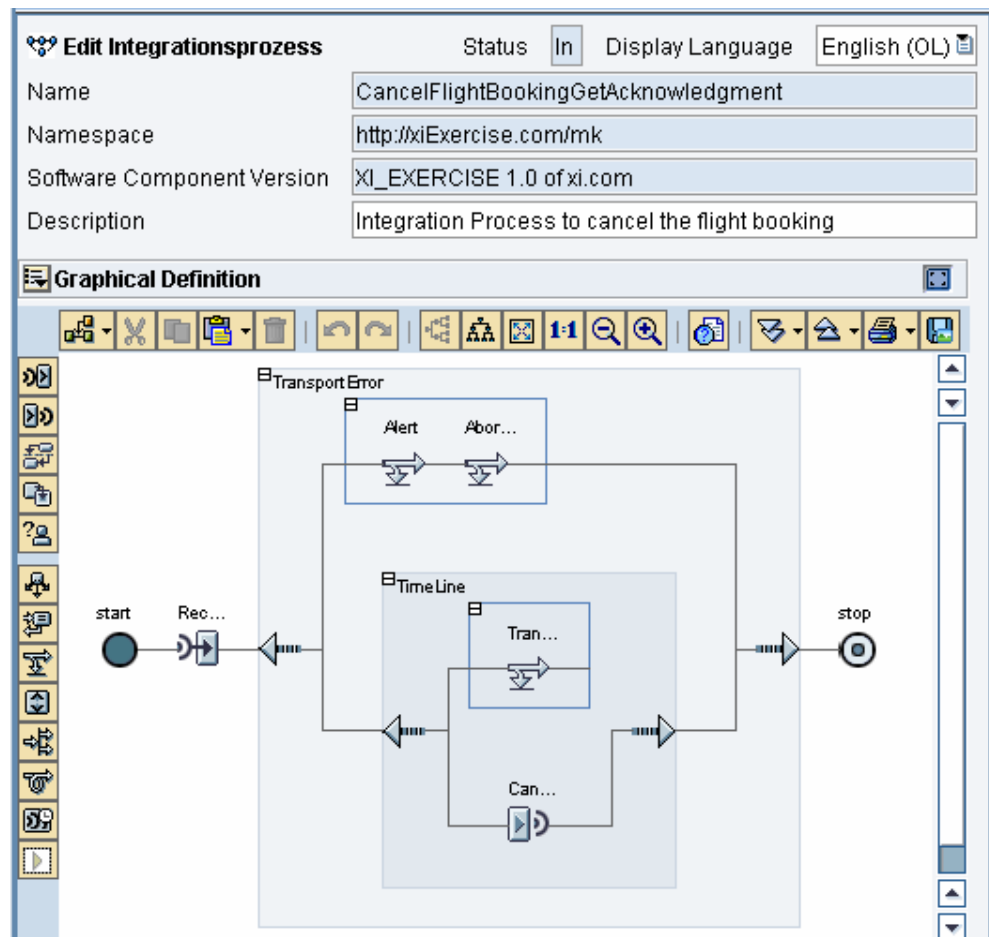
1.3 Business Description of the Process Integration Scenario

For a business description, see *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking*.

This process integration scenario is enhanced in such a way that an integration process is inserted in the communication between the travel agency and the airline *Lufthansa(LH)* (proxy-to-proxy communication). The integration process requests a technical (=transport) acknowledgment as proof of successful processing on the Integration Server. If the acknowledgment does not arrive within a minute, an exception is thrown, which in turn causes a mail to be sent to the sender.

1.4 Overview of the Integration Process

The following screenshot gives an overview of the integration process to be developed.



Legend:

- *Rec...* = *ReceiveCancelFlightBooking* (receive step)
- *TransportError* = *TransportError* (block)
- *Alert* = *Alert* (control step)
- *Abor...* = *AbortProcess* (control step)
- *TimeLine* = *TimeLine* (block)

- *Tran...* = *TransportErrorIE* (control step)
- *Can...* = *CancelFlightBooking* (send step)
- *Alert and Abor...* are in the exception branch of *TransportErrorIE*
- *Tran...* is in the deadline branch of *TimeLine*

1.5 Systems and Development Process

See *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking*.

1.6 Users and Authorizations

See *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking*.

2 Design

This section describes all steps that you must perform to enhance the process integration scenario in the Enterprise Services Repository.

2.1 Calling the Enterprise Services Builder (Enterprise Services Repository)

To call the Enterprise Services Builder (the user interface of the Enterprise Services Repository), perform the following steps:

1. In the user menu of the SAP system of the Integration Server choose *Process Integration* → *Start Integration Builder*.
2. On the SAP NetWeaver PI start page, choose the *Enterprise Services Repository* hyperlink.
3. Log on using the XIDEMO user. The system calls the Enterprise Services Builder (Enterprise Services Repository).

2.2 Defining Namespaces

You use the namespace from the exercise *Canceling a Flight Booking*.

For more information, see the document *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking*.

2.3 Enhancing the Process Integration Scenario

To enhance the existing process integration scenario *CancelFlightBooking*, perform the following steps:

- Create an abstract service interface.
- Create an operation mapping.
- Create an integration process.
- Create another component view for the process integration scenario *CancelFlightBooking*.

2.4 Creating an Abstract Service Interface

Integration processes exchange messages with other application components by using abstract service interfaces. The abstract service interfaces of an integration process are collectively known as the *process signature*.

Create the abstract service interface *BookingOrderCancelationRequest_Abstract* and specify the attributes of the service interface as follows:

1. Use the context menu entry *New* on the service interfaces entry in the tree structure of your namespace to create a new service interface.
2. In the *Name* field, enter `BookingOrderCancellationRequest_Abstract`.
3. The software component version and namespace are already entered.
4. In the *Description* field, enter `Cancel flight booking order using Integration Process`.
5. Choose *Create*.
The system opens the *Edit Service-Interface* object editor.
6. On the *Definition* tab page, select category `Abstract, Interface Pattern Stateless (XI30-Compatible), Operation Pattern Normal Operation and Mode Asynchronous`.
7. Save the message interface and close the object editor.
The outbound interface will be defined later.
8. As *Request - Message Type*, select `BookingOrderCancellationRequest` and your namespace.
9. Save the service interface.

2.5 Creating the Operation mapping

Create the operation mapping `BookingOrderCancellation_BPM2Airline`.

1. Use the context menu entry *New* on the Operation Mapping entry in the tree structure of your namespace to create a new service interface.
2. In the *Name* field, enter `BookingOrderCancellation_BPM2Airline`.
3. The software component version and namespace are already entered.
4. In the *Description* field, enter `Cancel flight booking order using Integration Process`.
5. Choose *Create*.
The system opens the *Edit Operation Mapping* object editor.
6. As *Source Operation* select `BookingOrderCancellationRequest_Abstract` (software component version `XI_EXERCISE 1.0`, namespace `http://xiExercise.com/<p>`) and as *Target Operation* select `FlightBookingOrderCancellationRequest_In` (software component version `SAP BASIS 7.10`, namespace `http://sap.com/xi/XI/Demo/Airline`).
7. Select *Read Operations*.
8. As Message mapping select `BookingOrderCancellation_Agency2Airline` using the Input Help.
Since the abstract interface of the signature matches `BookingOrderCancellationRequest`, you can reuse the existing message mapping in this operation mapping `BookingOrderCancellation_Agency2Airline`.
9. Save the operation mapping and close the object editor.

2.6 Creating the Integration Process

Create the integration process `CancelFlightBookingGetAcknowledgment` in your software component version in your namespace.

Note that any subelements that are shown with a symbol in this document appear around the edge of the *GraphicalDefinitions* screen area, and you can insert them into the integration process by using drag and drop.




2.6.1 Overview of the Integration Process

The integration process receives the cancellation message from the travel agency in a receive step and then forwards it on to the pipeline in a send step.



The send step expects a technical acknowledgment. To ensure that this happens within a defined time period, a deadline branch is inserted for monitoring purposes. Another send step in this deadline branch throws an exception if the deadline for the technical acknowledgment

is not kept. Finally, an exception branch determines how the system is to react to the missed deadline (by throwing an alert).

2.6.2 Developing the Integration Process in Detail

1. Use Object → New in the menu in Enterprise Services Builder. Under *Process Integration Scenario Objects* select *Integration Process*.
2. In the *Name* field, enter `CancelFlightBookingGetAcknowledgment`.
3. For software component version and namespace select `XI_EXERCISE 1.0, http://xiExercise.com/<p>`.
4. In the *Description* field, enter `Integration Process to cancel the flight booking`.
5. Choose *Create*.
The *Edit Integration Process* screen opens up.
6. Use drag and drop to insert a receive step () in the graphical process editor between *Start* and *Stop*.
7. Specify the properties of the receive step. To do this, create a container with the following properties in the bottom right screen area:
 - Name: `CancelFlightBooking`
 - Category: `Abstract Interface`
 - Type: `BookingOrderCancellationRequest_Abstract` (use the input help to select from your software component version/namespace)
 - Multiline: deselected
 - Scoper: `Process`
8. Define the other properties of the step in the *Properties* screen area:
 - Step Name: `ReceiveCancelFlightBooking`
 - Message: `CancelFlightBooking`
 - Start Process: selected
 - Mode: `Asynchronous`
9. Insert a block () between the send step and the end.
Specify the properties of the step in the *Properties* screen area:
 - Step Name: `TransportError`
 - Mode: `Default`
 - Exceptions: `TransportErrorIE`
10. Check whether the container screen area contains the same information as the send step.
11. Insert another block with the name *Timeline* and mode *Default* in the *TransportError* block. Again, the container is the same as the surrounding block.
12. Define an exception branch for the outer block: To do this, first position the cursor on the outer block *TransportError* and choose *Insert → Exception Branch* in the context menu. Next, click the exception branch and use the input help to enter `TransportErrorIE` as the exception handler in the *Properties* screen area.
13. Insert a send step () in the inner block *Timeline* and specify the following properties:
 - Step Name: `CancelFlightBooking`
 - Mode: `Asynchronous`
 - Message: `CancelFlightBooking` (select the interface variable from the input help)
 - Acknowledgment: `Transport`
 - Receiver From: `Send Context`
 - System Error (under Exceptions): `TransportErrorIE`
14. Define a deadline branch for the inner block *Timeline*: To do this, first position the cursor on the block *Timeline* and choose *Insert → Deadline Branch* in the context menu. Next, click the deadline branch and, in the *Properties* screen area, enter

Creating the Step as the reference date/time of the step and 1 Minute as the duration.

15. Insert a control step () in this deadline branch. In the *Properties* screen area, enter the following (or select from the input help):
 - Step Name: TransportErrorIE
 - Action: Throw Exception
 - Exception: TransportErrorIE
16. Define two control steps () one after the other for the exception branch of the outer block.

Specify the first control step as follows:

 - Step Name: Alert
 - Action: Throw Alert
 - Alert Category: <p>_ALERT
 - Alert Message: Transport Error in IS

The alert category is used to establish the connection to the ABAP Alert Service. The corresponding settings in the Web AS are described in section 3.3.
17. Specify the other control step as follows:
 - Step Name: AbortProcess
 - Action: Cancel Process
18. Activate the integration process on the *Change Lists* tab page.

2.7 Defining Another Component View

Enhance the process integration scenario *CancelFlightBooking* by adding a component view with the name *ABAP_Proxy_2_ABAP_Proxy_With_BPM_Monitor*.

Copy the component view *ABAP_Proxy_2_ABAP_Proxy* to reuse as much information as possible.

Perform the following steps:

1. Open the process integration scenario *CancelFlightBooking*.
2. Switch to change mode.
3. Select the component view *ABAP_Proxy_2_ABAP_Proxy*.
4. In the object menu, choose *Component View* → *Copy*.
5. Change the description of the duplicate component view to *ABAP_Proxy_2_ABAP_Proxy_With_BPM_Monitor*.

Specify the new component view as follows.

First insert the BPMMonitor as application component:

1. In the graphical editor, position the cursor in the component view panel. In the context menu, choose *Insert Application Component*.
2. On the *Insert Application Component* screen, select the *Product Version* radio button.
3. Call the input help for the product version.
4. In the input help, select the product version XI EXERCISE, 1.0 of xi.com and confirm the selection by choosing *Apply*.
5. On the *Insert Application Component* screen, select the *Role* tab page. In the *Name* and *Description* fields, enter *BPMMonitor*
6. Select the *Integration Process* tab page. Select the process *CancelFlightBookingGetAcknowledgment* from your namespace using input help.
7. Select the *Further Attributes* tab page. As WSCI Name set *BPMMonitor*.
8. Choose *Apply*.
9. Delete the existing connections between the two old application components and move the new application component between the other two simply by using Drag and Drop.

Insert the actions:

Insert a new action *MonitorCancelFlightBooking* in the application component *BPMMonitor*. To do this do the following:

1. Position the cursor on the application component *BPMMonitor*.
2. In the context menu, choose *Create Action*.
3. On the *Create a New Action* screen, under *Type of Usage*, select the *Internal* radio button.
4. Choose *Continue*.
5. In the *Name* field, enter *MonitorCancelFlightBooking*.
The software component version and namespace are already entered.
6. Choose *Finish*.
The system opens the *Display Action* object editor.
7. Switch to change mode. To do this, choose *Switch Between Display and Edit Modes* in the object editor toolbar.
8. In the *Outbound and Inbound Interfaces* frame select the *Operation BookingOrderCancelationRequest_Abstract* of your namespace *http://xiExercise.com/<p>* in the dropdown list box.
9. Save the action and close the object editor.

Add new connections:

1. Click the action *CancelFlightBookingOrder*. Hold down the *SHIFT* key and click the action *MonitorCancelFlightBooking*.
2. In the context menu, choose *Create Connection Between Selected Actions*.
3. On the *Change: Connection* screen, on the *Assign Interfaces* tab page, select the following:
 - In the *From: Asynchronous Outbound Interfaces of Action* frame, the interface *BookingOrderCancelationRequest_Out*
 - In the *To: Asynchronous Inbound Interfaces of Action* frame, the interface *BookingOrderCancelationRequest_Abstract*No mapping is required here.
4. Choose *Apply*,
5. Click the action *MonitorCancelFlightBooking*. Hold down the *SHIFT* key and click the action *CancelSingleFlightBooking*.
6. In the context menu, choose *Create Connection Between Selected Actions*.
7. On the *Change: Connection* screen, on the *Assign Interfaces* tab page, select the following:
 - In the *From: Asynchronous Outbound Interfaces of Action* frame, the interface *BookingOrderCancelationRequest_Abstract*
 - In the *To: Asynchronous Inbound Interfaces of Action* frame, the interface *FlightBookingOrderCancelationRequest_In*
8. Select the *Assign Mapping* tab page and select the mapping *BookingOrderCancelation_BPM2Airline* you just created.
9. Choose *Apply*,
10. Save the process integration scenario.

3 Configuration

You perform all activities in this section in the Integration Builder (Integration Directory). To start the Integration Directory, proceed as follows:

1. In the user menu choose *Process Integration* → *Start Integration Builder*.
2. On the SAP NetWeaver PI start page, choose the *Integration Directory* hyperlink.
3. Log on using the XIDEMO user. The system calls the Integration Builder (Integration Directory).

To group the relevant configuration objects, use the configuration scenario that you used to configure the process integration scenario *CancelFlightBooking*.

3.1 Creating the Integration Process Component

1. In the Integration Directory navigation area, select the *Objects* tab page.
2. Expand the *Communication Component* node.
3. Position the cursor on the *Integration Process* node and choose *New* in the context menu.

The system calls the wizard for creating an integration process communication component.

4. Choose *Continue*.
5. The next screen displays all integration processes from the Enterprise Services Repository in a list.
6. Select the integration process *CancelFlightBookingGetAcknowledgment* from your namespace.
7. On the next screen, give the process the following name: *BPMMonitor_<p>*.
<p> is the three-figure abbreviation for your namespace.
8. Choose *Finish*.
9. Activate the change list containing the integration process component.

3.2 Configuring the Process Integration Scenario

1. Open the configuration scenario.
2. Switch to change mode.
3. In the object editor, choose *Model Configurator*.
4. Select the push button *Apply New Version from ES Repository* directly next to the process integration scenario name to get the just created BPM View from ES Repository.
5. Select the push button *Select Component View*.
6. In the preview area, select the component view *ABAP_Proxy_2_ABAP_Proxy_With_BPM_Monitor*.
7. Confirm the warning message (about overwriting) by choosing *Continue*.
8. Perform the following configuration steps in the same way as in the exercise described in *Demo Example - Developing an Process integration scenario* (Configuring Proxy-to-Proxy Communication).

Make the following configuration settings for the process integration scenario enhancement:

- In the Assign Services configuration step use the input help to assign the Integration Process *BPMMonitor_<p>* to the *BPMMonitor..*
- As before, assign the partyless business systems <SID>_105 and <SID>_106 to the application components *Agency* and *Airline*.
- In the Configure Connections configuration step you do not need a mapping or a communication channel for the connection between the actions *CancelFlightBookingOrder* and *MonitorCancelFlightBooking*.
- In the connection between the actions *MonitorCancelFlightBooking* and *CancelSingleFlightBooking*, select the communication channel *GeneratedReceiverChannel_XI*. On the *Mapping* tab page the operation

mapping *BookingOrderCancelation_BPM2Airline* from your namespace should be displayed.

9. Generate the configuration objects as described in the *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking* (Configuring Proxy-to-Proxy Communication).

3.2.1 Manual Editing After Generation

- Create the routing condition (*Airline = LH*) for the receiver service *BPMMonitor* in the receiver determination *<SID>_105/BookingOrderCancelation_Request_Out* of the scenario. Delete the line with the previous receiver *<SID>_106* and the routing condition (*Airline = LH*) to make sure the cancellation for LH is really sent to the process.
- Optional: Delete the interface determination *<SID>_105/BookingOrderCancelationRequest_Out||<SID>_106*.

Save the receiver determination and activate the change list

3.3 Configuration Settings in the SAP System

Make the following settings in the SAP system of the Integration Server client:

- Establish the connection to the central alert service that sends the mail. First, you call transaction **ALRTCATDEF** to create the alert category *<p>_ALERT* used in the integration process.
You have to create a customizing request to do this.
- Set the *Message Title, Long and Short Text* for this Alert: *Timeout in Cancel Flight Booking occurred*.
- Next, assign XIDEMO User to this alert category by choosing the *Fixed Recipients* menu function.
- Save the Alert Category.
- Use transaction **SU01** to make sure that a mail address is maintained for the user.

4 Execution

First check that the cancellation of the booking works. Execute a flight booking for the airline and then cancel it with your report from the first exercise (see *SAP NetWeaver Process Integration 7.1 - Demo Examples Exercise Cancel Flight Booking*). Check that the booking was canceled correctly as described in the first exercise.

To throw the required exception, it is sufficient to enter a non-functional communication channel for the receiver agreement of the receiver *<SID>_106* in your scenario.

Create a new channel *Road2Nowhere* in business system *<SID>_106* for this purpose.

1. In the Integration Directory in the object tree select *Communication Channel*. Use the context menu entry *New* to create a new channel. Select the Communication Component *<SID>_106* and set *Road2Nowhere* as communication channel name. Select your cancellation scenario.
2. Press *Create*.
3. As adapter type Use *XI* and choose *Receiver*. As target host set *Road2Nowhere*, as port set 50000, as Path set *Road2Nowhere*. As Authentication Type select *Anonymous Logon*.
4. Save the channel.
5. Open the receiver agreement *BPMMonitor_<p>|<SID>_106*
|FlightBookingOrderCancelationRequest_In and select the change mode.
6. Select the channel *Road2Nowhere* and save the receiver agreement.
7. Activate the change list.

Now, execute a flight booking for the airline and then cancel it again with your report from the first exercise. After a short time (less than a minute), a mail should arrive at the specified mail

address because the process will send the alert after the timeout specified in the process definition.

4.1 Checking the Result

In addition to the options described in the first exercise, you can use BPM monitoring and workflow monitoring for this task.

- In the Enterprise Services Repository, you can check your integration process for errors. To do this, choose *Integration Process* → *Check* in the object menu of the integration process. An integration process without errors (warnings and information messages are permitted) should result in a functioning workflow in the ABAP cache (see below).
- To track the steps of the work item, use transaction **SXMB_MONI** in the client of the Integration Server (refresh possible) to select the message from the agency to the process. In the column *Outbound* you can jump to the process execution clicking the PE link. From here you can navigate to a graphical workflow representation. To do this, choose *Goto* → *Grafical Workflow Log in the menu*. In the event of an error, for example, you can find out how far the process has got and, if it was canceled, at which substep this occurred.
- When you activate your integration process component in the Integration Directory, it is replicated to the ABAP cache of SAP Web AS and a corresponding workflow is generated. You can check this by calling transaction **SXI_CACHE** and looking for your process under *Integration Processes* in the list in the window on the right (to make this easier, first sort the list by process names). A return code <> 0 means that the workflow generation was not successful. You can position the cursor on your process and initiate it again if necessary. If the workflow is active, you can display it graphically by choosing *Display Active Version*. You can execute tests from the graphical display.