SAP NetWeaver 7.31

SAP NetWeaver Process Integration

Simple Use Cases for De-central Adapter Engine

Document Version 1.0 – March 2012

THE BEST-RUN E-BUSINESSES RUN SAP

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

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 expressively 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 Developer Network (SDN)

You can find this documentation in SAP Developer Network (SDN) titled
Simple Use Cases for Decentral Adapter Engine.

1 Preface

Constraints

The texts, references, and graphics contained in this manual have been compiled with utmost care; nevertheless, it is impossible to guarantee that they are fully without error. SAP cannot assume any responsibility for the correctness or completeness of the following documentation; the user alone is responsible for verifying the information contained therein.

SAP will only assume liability for damage arising from the use of this documentation – irrespective of the pertinent legal basis – in the case of intentional or active negligence, under no other circumstances will a warranty be made.

Definition

This manual describes simple application cases for the de-central adapter engine for process integration and all the configuration steps that are necessary to execute the application cases on the basis of SAP NetWeaver 7.31.

Intended Audience

This manual is intended to be used by both technology and application consultants.

Structure

The structure of this document follows the sequence of steps required to configure and run the use cases.

Additional Documentation

List of related documentation

Documentation	Purpose
SAP NetWeaver Library documentation: Demo Examples	Introduction to demo examples and description of the process integration scenarios
SAP NetWeaver Documentation Simple Use Cases	Introduction to Simple Use Cases and description of the simple use case scenarios for Process Integration
Configuration of SAP NetWeaver Process Integration	Configuring usage type Process Integration

This document is updated on a regular basis in SAP Developer Network (SDN) and is available as *Simple Use Cases for Decentral Adapter Engine*.

2 Introduction and Prerequisites

2.1 Introduction

Use cases are based on simple examples that enable you to learn the main features of SAP NetWeaver Process Integration. These instructions will enable you to understand, configure, and execute simple scenarios (that are based on fundamental communication patterns), as well as check that the scenarios are executed correctly. The prerequisite for this document is that you have a general understanding of SAP NetWeaver Process Integration.

The following use case variants for the de-central adapter engine are available at present:

Variant	Description

Variants 1 and 6: One sender and one receiver (see **Error! Reference source not found.** and 8)	Represents message exchange between one sender and one receiver on the de-central adapter engine. The message processing is done locally on the de-central adapter engine. Variant 6 is executed using EOIO protocol. The message exchange is done between Adapter Engine and Integration Server.
Variant 2: One sender and two receivers (see **Error! Reference source not found.**),	Represents message exchange between one sender and two receivers. The message processing is done locally on the de-central adapter engine.
Variant 3: One sender and two receivers – with routing condition (see **Error! Reference source not found.**)	Represents message exchange between one sender and two receivers in which the way the message is forwarded to the receivers is determined by routing conditions The message processing is done locally on the de-central adapter engine.
Variant 4: One sender and two receivers – with mapping (see **Error! Reference source not found.**)	Represents message exchange between one sender and two receivers. A mapping is executed when the message is forwarded to one of the two receivers. The message processing and mapping are done locally on the de-central adapter engine.
Variant 5: One sender and two receivers – with XML-Validation (see **Error! Reference source not found.**)	Represents message exchange between one sender and two receivers. A mapping is executed when the message is forwarded to one of the two receivers. The payload is validated using XML-Validation in inbound processing of the integration engine. The message processing, mapping and XML validation are done locally on the de-central adapter engine.

You can aim to learn about the following with the use cases:

- Working with the Enterprise Services Builder – Enterprise Services Repository: Understanding existing design objects
- Working with the Integration Builder – Integration Directory: Configuring simple scenarios and understanding created configuration objects
- Working with the runtime: Executing the use cases and checking the results by using monitoring

SAP provides the required design objects and files for these learning targets. This document contains a complete set of instructions that enable you to configure and execute the use cases without any additional special prerequisites.

SAP NetWeaver Process Integration must be installed and configured, and be error free. In this respect, note the information under 2.2.1.

Within the scenarios file directories on the host of the SAP system on which the de-central Adapter Engine is installed serve as the senders and receivers of messages. The use cases rely on file directories (and therefore the file/FTP adapter) to keep the technical prerequisites as simple as possible.

In variants 1 to 5 the messages are sent directly from the sender adapter to the receiver adapter without using the ABAP IS runtime to show the possibility of using the Advanced Adapter Engine.

In variant 6 the messages are processed in sequence, the message exchange is done between adapter engine and integration server.

This document assumes that you will configure the variants **in succession**, in the order specified. While it is technically possible to configure the variants in a different order, doing so would mean that the order of the step-by-step instructions in this document would no longer be correct.

> If you need to configure and execute the use cases as quickly as possible, skip the chapters concerning the design objects used (**Error! Reference source not found.**, **Error! Reference source not found.**…) and start instead with configuration. You can then return to the skipped chapters at a later stage for a better understanding of the various concepts.

2.2 Prerequisites

2.2.1 General Prerequisites

To configure and execute the use cases, SAP NetWeaver with usage type Process Integration must be correctly installed and configured. The following table lists the prerequisites and the relevant guides:

List of required steps

Step	Documentation
1. You have installed SAP NetWeaver 7.31.	*Master Guide – SAP NetWeaver* *Installation Guide – SAP Web AS ABAP* and *Java* You can find the corresponding guides on SAP Service Marketplace in the Implementation Documentation Center for SAP NetWeaver 7.3 and SAP NetWeaver 7.3 including Enhancement Package 1
2. You have imported the content for the Enterprise Services Repository corresponding to the latest support package/patch.	SAP Note 836200 The simple use cases are located in the Enterprise Services Repository in the software component *SAP BASIS*, software component version *SAP BASIS 7.31*, in the namespaces *http://sap.com/xi/XI/System/Patterns*.
3. You have configured SAP NetWeaver Process Integration.	Configuration of SAP NetWeaver Process Integration

4. You have configured the System Landscape Directory (SLD).	Configuring, Working with and Administering System Landscape Directory

Note that you set up the use cases in your development or test system, but **not** in your productive system.

2.2.2 Providing the necessary User and Authorizations

To log on to the SAP system to configure the simple use cases, you have to create a user *XIDEMO* with the following roles in the integration server client (001):

- `SAP_XI_ADMINISTRATOR`
- `SAP_XI_CONFIGURATOR`
- `SAP_XI_MONITOR`
- `SAP_XI_DEMOAPP`

2.2.3 Setting Up the File Directories

The variants are based on a very simple system landscape that comprises various different file systems (on the host of the SAP system on which the decentral Adapter Engine is installed). When the use cases are executed, files are moved between file directories. Therefore, to be able configure the use cases, the file directories must have been created and described in the System Landscape Directory. Only then can you access the directories during configuration in the Integration Directory.

Perform the steps below to create the required directories on the host of the SAP system on which the de-central Adapter Engine is installed.

1. Using Windows Explorer, open the file directory of the SAP system on which the de-central Adapter Engine is installed.
2. Create a directory in which you can save files temporarily and create sub directories XiPatterns and XiPatternsO in it.

   A work directory already exists on many computers that can be used for this purpose and under which you can create the required directories.
3. Create three further sub directories in the sub directory XiPatterns. Suggested names: `XiPattern1`, `XiPattern2`, `XiPattern3`.
4. Create two further sub directories in the sub directory XiPatternsO. Suggested names: `XiPattern4, XiPattern5.`

   You must have authorization to access this directory to be able to test the process integration scenario. Bear this in mind when you define the directory, and, if applicable, contact your system administrator.
5. Copy the following three files to directory *XiPattern1*
   - *ZXiPatternSenderFile1.xml*
   - *ZXiPatternSenderFile2.xml*
   - *ZXIPatternSenderFile3.xml*

6. Copy the following file to directory *XiPattern4*

   - XiPatternSenderFilesEOIO.zip (This file contains 1000 xml files for the EOIO scenario, they will be unpacked when executing the scenario.)

     These files are provided by SAP in a ZIP file. The ZIP file for 7.31 containing also the EOIO test files is available in SAP Developer Network (SDN) and can be found using direct link EHP 1 for SAP NetWeaver Process Integration 7.3 – Test Files for Simple Use Cases.

2.2.4 Creating the Technical and Business Systems in the System Landscape Directory

Perform the following steps to create the required technical and business systems in the System Landscape Directory (SLD):

1. To call the System Landscape Directory, choose *Process Integration* → □*Start Integration Builder* in the user menu.
2. Start the SLD. To do so, on the Process Integration start page, choose *System Landscape Directory*.
3. Log on using the XIDEMO user you created in 2.2.2.
4. On the initial screen of the SLD, choose *Technical Systems*.
5. Create a new technical system. To do so, choose *New Technical System.*
6. Select the type *Standalone* by choosing the appropriate radio button.
7. Choose *Next*.
8. On the *Technical System Wizard - System Details* screen, enter the following information about the technical system:
   - *Technical System Identification:* Enter a name according to the following naming convention: `<SID>_FileSystem`. In this case, `<SID>` is the **system ID of the de-central Adapter Engine**.
   - *System Host Name*: Enter the **host name of the de-central adapter engine**, for example `pwdf2638`.
9. Choose *Finish*.
10. Now create five new business systems. To create the first business system, on the SLD start page, choose *Business Systems*.
11. Choose *New Business System.*.
12. Select the type *Standalone* by selecting the appropriate radio button.
13. Choose *Next*.
14. Select the technical system that you defined previously (dropdown list box in field *System*).
15. Choose *Next*.
16. Enter the name `<SID>_dAAE_FileSystem_XiPattern1`. <SID> is the system ID of the de-central adapter engine.
17. Choose *Next*.
18. Specify the assigned Integration Server to be used for process integration (field *Related Integration Server*).

19. Choose *Finish*.
20. Following the same procedure, create business systems with the following names:
    - `<SID>_dAAE_FileSystem_XiPattern2`
    - `<SID>_dAAE_FileSystem_XiPattern3`
    - `<SID>_dAAE_FileSystem_XiPattern4`
    - `<SID>_dAAE_FileSystem_XiPattern5`

2.2.5 Defining Business System Components in the Integration Directory

In the following steps you define communication components for the business systems you just created. You need to create communication components to be able to address the business systems as senders/receivers of messages when you configure the process integration scenario in the Integration Directory.

To prepare the configuration of the process integration scenario, you must do the following:

- Define the involved business systems as communication components (business system components) in the Integration Directory

  Only then can you address the business systems as senders or receivers of messages.

- Define communication channels for outbound message processing (for all relevant receiver business systems)

You define communication components in the Integration Directory.

Therefore, to execute the activities described below, call the Integration Builder (user interface of Integration Directory).

1. To call the Integration Builder, choose *Process Integration* → *Start Integration Builder* in the user menu.
2. On the initial screen, choose the hyperlink *Integration Builder* (under *Integration Directory*).
3. Log on using the XIDEMO user you created in 2.2.2. The system opens the Integration Builder.

You use the steps below to define a communication component (business system component) in the Integration Directory for each business system.

1. Clear the SLD cache. To do this, select *Environment* → *Clear SLD Data Cache* in the menu of the Integration Builder.
2. In the Integration Builder navigation area, on the *Objects* tab page, position the cursor on the *Communication Components Without Party* node.
3. To call the wizard for assigning business systems, choose *Assign Business System* in the context menu.
4. Choose *Continue*.
5. On the next screen, choose *Continue*.

   The business system components are not assigned to a party.
6. On the next screen, select the following business systems by using the corresponding checkboxes:
   - `<SID>_dAAE_FileSystem_XiPattern1`
   - `<SID>_dAAE_FileSystem_XiPattern2`

- `<SID>_dAAE_FileSystem_XiPattern3`
- `<SID>_dAAE_FileSystem_XiPattern4`
- `<SID>_dAAE_FileSystem_XiPattern5`

7. Make sure that the *Create Communication Channels Automatically* checkbox is **deselected**.

   The relevant communication channels are created when the process integration scenario is configured.

8. Choose *Finish*.

   The system displays the communication components in the Integration Builder navigation area, on the *Objects* tab page, under *Communication Components Without Party → Business System*.

Perform the following steps to activate the communication components.

1. In the navigation area, choose the *Change Lists* tab page.
2. Choose *Open → Standard Change List*.
3. Check that the five communication components you just created are contained in the change list.
4. Position the cursor on the *Standard Change List* node and choose *Activate* () in the context menu.

2.3 Design Objects Used

This section explains all the design objects that are used for describing the variants.

The subsequent description of each variant will refer to the description in this section; the relevant design objects will only be listed briefly in each case.

Process Integration Scenario/Component View

A process integration scenario describes the process flow and represents the points in the process flow at which messages are exchanged between application components (in this case, the application components *Sender* and *Receiver*). Component View *File2File_NoMapping_InOrder* represents the variant with message exchange in EOIO. The other variants are not based on component views because they are using the integrated configuration.

Interface Objects

Interface objects describe the communication mode and the data structure for exchanging messages. Interface objects comprise service interfaces, message types, and data types, which you can open by means of forward navigation, starting with the Service Interfaces.

Mapping Objects

If the interfaces at the sender and receiver are different, a mapping is required.

Communication Channel Templates

You use communication channel templates to define details, which were defined when the process integration scenario was defined, for the inbound or outbound processing of the message. You can then use the communication channel template to create the corresponding communication channel when configuring in the Integration Directory. You then need to specify all the details, which are only known at configuration time, in the communication channel.

The specific design objects will be listed in the specific variants below.

3 Variant 1: One Sender and One Receiver using de-central Advanced Adapter Engine

This variant enables you to configure and execute the simplest kind of message exchange:

One sender (file directory *XiPattern1*) sends a message to one receiver (file directory *XiPattern2*). Hereby, the message is exchanged between sender and receiver adapter by using one single de-central Adapter Engine without using the integration server runtime. No mapping is executed between the outbound and inbound interface in this variant.

Summary of Design Objects Used

The design objects used are summarized in the following table.

For a general description of the various object types, see SAP Help Portal at: Enterprise Services Repository.

Design Objects Used

Object Type (Name)	**Description**
Service Interface (*XiPatternInterface1*)	Specifies the communication mode (asynchronous) and references the message type used.
Message Type (*XiPattermMessage1*)	Describes the message sent at runtime and references the data type used.
Data Type (*XiPatternDataType1*)	Describes the data structure of the message
Communication Channel Templates for Sender and Receiver (*XiPatternChannel_FileSender*, *XiPatternChannel_FileReceiver*)	Specifies the sender (and receiver) adapter configuration details, which are defined at design time. Communication channel templates enable a communication channel to be preconfigured at design time.

3.1 Configuring the Process Integration Scenario

You perform the following configuration steps in the Integration Directory.

For this scenario, the integrated configuration object is used for configuration.

For more information, see Defining the Integrated Configuration.

Calling the Integration Directory

First, call the Integration Directory. On the *Process Integration* initial screen, choose *Integration Directory.*

Creating a Configuration Scenario

Perform the following steps to create a configuration scenario to group the configuration objects.

1. In the Integration Directory main menu, choose *Object* → *New*.
2. Under *Administration* select *Configuration Scenario*.
3. Enter the name `XiPatternOneStepScenario_dAAE_AAE2AAE`.
4. Choose *Create*.
5. Save the configuration scenario.

Creating the Communication Channels for Sender and Receiver

Perform the following steps to create the communication channels.

1. In the Integration Directory main menu, choose *Object* → *New*.
2. Under *Collaboration Profile* select *Communication Channel*.
3. Call the input help (□▾) for the Communication Component.
4. Select the communication component *<SID>_dAAE_FileSystem_XiPattern1* and choose *Apply*.
5. In field *Communication Channel* enter `XiPatternChannel_FileSender`.
6. Under *Add to Scenario* select `XiPatternOneStepScenario_dAAE_AAE2AAE` (input help).
7. Choose *Create*.

   The newly created communication channel is opened.
8. In the menu of the communication channel object editor select *Communication Channel* → *Apply Template*.
9. Select the template `XiPatternChannel_FileSender`.
10. Choose *Apply*.
11. Under *File Access Parameter* in the *Source Directory* field, enter the path in the file directory that you defined under **Error! Reference source not found.**. In the case of the sender communication channel, enter the whole path for the file directory *XiPattern1*.

    Example: `C:/<selected path>/XiPatterns/XiPattern1`
12. In the field *Adapter Engine* select the de-central adapter engine af.<SID>.<hostname>

    Example: `af.v4d.hostname`
13. Save the communication channel.
14. Activate it by using the menu *Communication Channel* → *Activate*.

15. To check if the channel works you can directly jump to the channel monitoring using the menu path Communication Channel → Channel Monitoring
16. The PI Communication Channel Monitoring page will open up where you can do a self test for the channel. To do this click on the button *Ping Channel*.
17. The status page for the channel will show up with detailed information of the executed self test.
18. Following the same procedure, create the communication channel *XiPatternChannel_FileReceiver* (communication component `<SID>_dAAE_FileSystem_XiPattern2`) using the template *XiPatternChannel_FileReceiver*. Select the de-central adapter engine and enter the path for the file directory *XiPattern2*.

    If there already is an entry for the communication component in the *Create Communication Component* screen delete it before calling the input help. Otherwise the input help may not provide the `<SID>_dAAE_FileSystem_XiPattern2.`

19. Save the communication channel and activate it.
20. Open the Channel Monitoring and check the status of the channel. Save the scenario.
21. Activate the change list.

Creating the Integrated Configuration

1. In the Integration Directory main menu, choose *Object* → *New*.
2. Under *Collaboration Agreement* select *Integrated Configuration*.
3. Call the input help (❐) in the field *Communication Component*.
4. Select the communication component `<SID>_dAAE_FileSystem_XiPattern1` and choose *Apply*.
5. In the fields *Interface* and *Namespace* enter `XiPatternInterface1` and `http://sap.com/xi/XI/System/Patterns`, respectively.

    The input help may not show the interface and namespace. Create the entry manually or choose Category *Any* in the F4 Help.

6. Under *Add to Scenario* select *XiPatternOneStepScenario_dAAE_AAE2AAE* (input help).
7. Choose *Create*.
8. In tab *Inbound Processing* select the communication channel `XiPatternChannel_FileSender`. For the software component version of sender interface select `SAP BASIS 7.31`.
9. In the tab *Receiver* select the communication component `<SID>_dAAE_FileSystem_XiPattern2`.
10. In tab *Receiver Interfaces* for name and namespace enter `XiPatternInterface1` and `http://sap.com/xi/XI/System/Patterns`, respectively. In Software Component Version set SAP BASIS 7.31.

The input help may not show the interface and namespace. Create the entry manually.

11. In the tab *Outbound Processing* select the communication channel `XiPatternChannel_FileReceiver`.
12. Save the integrated configuration.
13. Save the Configuration Scenario.

Activating the Change List

1. To do this, select the *Change Lists* tab in the Integration Directory navigation area.
2. Choose *Open* → *Standard Change List*.
3. Position the cursor on the *Standard Change List* node and choose *Activate* ()) in the context menu.

3.2 Executing the Use Case

You perform the following steps to check that the use case is executed without errors.

1. On the host of the de-central Adapter Engine, open the file directory *XiPattern1*.
2. Rename the file `ZXiPatternSenderFile1.xml` as `XiPatternSenderFile1.xml`.

   The File/FTP adapter is designed in a way that a file with exactly this name is recognized and moved from the source directory *XiPattern1* to the target directory *XiPattern2* every 60 seconds. If the configuration was executed without errors, the file will be moved from the source directory within 60 seconds after it is renamed.
3. Open the directory *XiPattern2* and check that the file has arrived.
4. Open the file and check whether the content has changed (compare with the original file *ZXiPatternSenderFile1.xml*).

The file must not have been changed when this variant is executed.

Furthermore, you can check the processing of the message in monitoring. To do this, proceed as follows:

1. Start the Monitoring by calling the *Configuration and Monitoring Home* on the de-central AAE: http(s)://<AAE-host>:<AAE-port>/pimon.
2. Select *Monitoring* for the *Adapter Engine*.
3. Choose *Message Monitor*.
4. Select *Database*.
5. Use appropriate filter criteria to restrict the number of XML messages displayed. Together with an adapted processing period the following advanced selection criteria for message header fields should be specified using the input field value help:

Selection Criteria	Value
Sender Component	`<SID>_dAAE_FileSystem_XiPattern1`
Receiver Component	`<SID>_dAAE_FileSystem_XiPattern2`
Interface	`XiPatternInterface1`
Interface Namespace	`http://sap.com/xi/XI/System/Patterns`

6. Choose *Go*.

   The system displays the selected messages in a table. You want to locate the messages that were sent during execution.

7. Select the message to be displayed by selecting the appropriate radio button and then choose *Open Message*.

   The tab *Payload* displays the message contents for the various versions of the message during processing.

Instead of the monitoring on the de-central AAE you can also use the *Runtime Workbench* as described in the Appendix.

Messages can also be viewed in the monitoring available in the Solution Manager. To do this proceed as follows:

1. Log into the connected SAP Solution Manager System and call the Solution Manager Workcenters using transaction **SOLMAN_WORKCENTER**.
2. Change to Tab *Technical Monitors* and follow the link to *PI Monitoring*.
3. Select the PI Domain you want to monitor (for example domain.<SID>.<hostname>).
4. Launch the Message Monitor either embedded or in a new window using the button *Message Monitor*.

Note that the message can not be seen in the Integration Server as the message is sent directly from the sender adapter to the receiver adapter.

4 Variant 2: One Sender and Two Receivers using de-central Advanced Adapter Engine

This variant extends variant 1 by adding an additional receiver. The message is sent from the sender to two receivers using the de-central Advanced Adapter Engine, without a mapping being executed at runtime.

One sender (file directory *XiPattern1*) sends a message to two receivers (file directory *XiPattern2 and XiPattern3*). The message is exchanged between sender and receiver adapter using one single Adapter Engine but without the integration server runtime. No mapping is executed between the outbound and inbound interface in this variant.

4.1 Design Objects Used

In this variant the same interfaces are used as in variant 1. Note that no process integration scenario exists for this variant.

4.2 Configuring the Process Integration Scenario

When you configure this variant, you will extend the configuration settings from variant 1. You use the same configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE* as in variant 1.

Opening the Configuration Scenario

In this variant, the configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE* created in variant 1 is extended.

The integrated configuration object is used again for configuration for this scenario. The model configurator is not used in this case.

For more information, see Defining the Integrated Configuration.

Creating the Communication Channel for the Second Receiver

Perform the following steps to create the communication channels.

1. In the Integration Directory main menu, choose *Object* → *New*.
2. Under *Collaboration Profile*, select *Communication Channel*.
3. Call the input help ( ) for the communication component.
4. Select the communication component *<SID>_dAAE_FileSystem_XiPattern3* and choose *Apply*.
5. In field *Communication Channel* enter `XiPatternChannel_FileReceiver`.
6. Under *Add to Scenario* select `XiPatternOneStepScenario_dAAE_AAE2AAE` (input help).
7. Choose *Create*.

   The newly created communication channel is opened.
8. In the menu of the communication channel object editor, select *Communication Channel* → *Apply Template*.
9. Select the template `XiPatternChannel_FileReceiver`.
10. Choose *Apply*.
11. Under *File Access Parameters* in the *Target Directory* field, enter the path in the file directory that you defined under **Error! Reference source not found.**. In the case of the sender communication channel, enter the whole path for the file directory *XiPattern3*.

    Example: `C:/<selected path>/XiPatterns/XiPattern3`

22. In the field *Adapter Engine* select the de-central adapter engine af.<SID>.<hostname>

    Example: `af.v4d.hostname`

12. Save the communication channel.

13. Open the Channel Monitoring and check the status of the channel as described in variant 1.

14. Save the scenario.

15. Activate the change list.

Changing the Integrated Configuration

1. In the Integration Directory main menu, open the Integrated Configuration Object <SID>_dAAE_FileSystem_XiPattern1 / XiPatternInterface1 / http://sap.com/xi/XI/System/Patterns.
2. Switch to change mode ( ).
3. In the tab *Receiver* add a new line and additionally select the communication component `<SID>_dAAE_FileSystem_XiPattern3`.
4. In tab *Receiver Interfaces* select the line for the communication component `<SID>_dAAE_FileSystem_XiPattern3`. In the table Receiver Interfaces and Mappings for name and namespace enter `XiPatternInterface1` and

`http://sap.com/xi/XI/System/Patterns`, respectively. In Software Component Version set SAP BASIS 7.31.

> The input help may not show the interface and namespace. Create the entry manually.

5. In the tab *Outbound Processing* select the line for the communication component `<SID>_dAAE_FileSystem_XiPattern3.` In the table Configuration select the communication channel `XiPatternChannel_FileReceiver`.
6. Save the integrated configuration.

Activating the Change List

1. To do this, select the *Change Lists* tab in the Integration Directory navigation area.
2. Choose *Open* → *Standard Change List*.
3. Position the cursor on the *Standard Change List* node and choose *Activate* ()) in the context menu.

4.3 Executing the Use Case

You perform the following steps to check that the process integration scenario is executed without errors.

As for the execution of variant 1, in directory *XiPattern1*, copy the file *ZXiPatternSenderFile1.xml* and rename it *XiPatternSenderFile1.xml*. The file will be moved from the directory within the next 60 seconds.

Since this variant has two target directories (receivers), check in both directory *XiPattern2* and *XiPattern3*, whether the file has arrived.

The content of the xml file should not change as no mapping is executed.

Check the message monitoring as described for variant 1. The message monitoring will display two messages, one for each of the receivers.

> In this variant, two messages are created from the outbound message (one for each receiver).

Again, the messages can not be seen in the Integration Server as the message is sent directly from the sender adapter to the receiver adapters.

5 Variant 3: One Sender and Two Receivers using de-central Advanced Adapter Engine – with Routing Condition

This variant extends variant 2 by adding a routing condition. Depending on the condition, the message is sent to one of the two receivers.

5.1 Design Objects Used

In this variant, the same interfaces are used as in variant 1. Note that no process integration scenario exists for this variant.

5.2 Configuring the Process Integration Scenario

When you configure this variant, you will extend the configuration settings from variant 1 and 2. You use the same configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE*.

Using the Configuration Scenario

In this variant, the configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE* created in variants 1 and 2 is extended.

For this scenario, again the integrated configuration object is used for configuration. The model configurator is not used in this case.

Changing the Integrated Configuration

1. In the Integration Directory main menu, open the integrated configuration object <SID>_dAAE_FileSystem_XiPattern1 / XiPatternInterface1 / http://sap.com/xi/XI/System/Patterns
2. Switch to change mode ().
3. In tab *Receiver*, select the cell for the receiver component *<SID>_dAAE_FileSystem_XiPattern2* in the column *Condition*.
4. Call the condition editor by choosing the input help icon ().
5. In the column *Left Operand*, call the expression editor by choosing the input help icon ().
6. In the expression editor, select the *XPath* radio button.

   The data structure of the message is displayed.
7. Expand the structure and select the element *CountryCode* by using the cursor.
8. Choose *OK*.
9. In the condition editor, choose the operator = in the column *Op*.
10. In the column *Right Operand*, enter *US*.
11. Choose *OK*.

You have specified the routing condition `CountryCode = US`.

At runtime, messages that have the value *US* entered for the element *CountryCode* are sent to receiver *<SID>_dAAE_FileSystem_XiPattern2*.

12. Following this procedure, create the routing condition `CountryCode = DE` for the configured receiver *<SID>_dAAE_FileSystem_XiPattern3*.
13. Save the integrated configuration.

Activating the Change List

1. To do this, select the *Change Lists* tab in the Integration Directory navigation area.
2. Choose *Open* → *Standard Change List*.
3. Position the cursor on the *Standard Change List* node and choose *Activate* ()) in the context menu.

5.3 Executing the Use Case

You perform the following steps to check that the process integration scenario is executed without errors.

In directory *XiPattern1*, copy the file *ZXiPatternSenderFile1.xml* and rename the copied file *XiPatternSenderFile1.xml*. The file will be moved from the directory within the next 60 seconds.

Check whether the file has arrived in directory *XiPattern2*. Open the file and check that the country code *US* is entered in the personal data.

There should be no file in directory *XiPattern3*.

Message monitoring contains one entry for this exchange of messages (see **Error! Reference source not found.**)..

> You can repeat this procedure for file *ZXiPatternSenderFile2.xml*. This file should arrive in directory *XiPattern3* (contains personal data with the country code *DE*).

Again, the messages can not be seen in the Integration Server as the message is sent directly from the sender adapter to the receiver adapters.

6 Variant 4: One Sender and Two Receivers using de-central Advanced Adapter Engine – With Mapping

In this variant the scenario from variant 3 is extended. The message is sent to two receivers. A mapping is executed when the message is forwarded to one of the two receivers.

6.1 Design Objects Used

In this variant the same interfaces are used as in variant 1. Note that no process integration scenario exists for this variant.

Mapping Objects

The mapping *XiPatternInterface1ToInterface2* is executed between the sender and receiver data structure for receiver *<SID>_dAAE_FileSystem_XiPattern3* during message exchange.

6.2 Configuring the Process Integration Scenario

When you configure this variant, you will extend the configuration settings from variant 3. You use the same configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE*.

Using the Configuration Scenario

In this variant the configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE* created in variant 1 and extended in variants 2 and 3 is extended to use a mapping for one of the receivers.

For this scenario, again the integrated configuration object is used for configuration. The model configurator is not used.

Changing the Integrated Configuration

1. In the Integration Directory main menu, open the integrated configuration object <SID>_dAAE_FileSystem_XiPattern1 / XiPatternInterface1 / http://sap.com/xi/XI/System/Patterns
2. Switch to change mode (✎).
3. In tab *Receiver Interfaces* select the line for the communication component <SID>_dAAE_FileSystem_XiPattern3. In table *Receiver Interfaces and Mappings* for mapping select the mapping *XiPatternInterface1ToInterface2* from software component version SAP BASIS 7.31*.*
4. In tab *Outbound Processing*, select the line for the communication component <SID>_dAAE_FileSystem_XiPattern3. In the table *Configuration*, select the communication channel XiPatternChannel_FileReceiver again as in this scenario another interface is used *(XiPatternInterface2).*
5. Save the Integrated configuration.

Activating the Change List

1. To do this, select the *Change Lists* tab in the Integration Directory navigation area.
2. Choose *Open* → *Standard Change List*.
3. Position the cursor on the *Standard Change List* node and choose *Activate* (⚙)) in the context menu.

6.3 Executing the Use Case

You perform the following steps to check that the process integration scenario is executed without errors.

In directory *XiPattern1*, copy the file *ZXiPatternSenderFile2.xml*. Rename the file to *XiPatternSenderFile2.xml*. The file will be moved from the directory within the next 60 seconds.

Check whether a file has arrived in directory *XiPattern3*. Open the file. The structure will have changed: the first and the second name of the person have been concatenated; the field for the telephone number has a new name.

Monitoring contains one entry for this exchange of messages.

You can repeat this procedure for file *ZXiPatternSenderFile1.xml*. This file is expected in directory *XiPattern2* (no mapping executed).

7 Variant 5: One Sender and Two Receivers using de-central Advanced Adapter Engine - with XML-Validation

This variant extends variant 4 by switching on the XML-Validation to check the schema of the file to be sent. The message is sent from the sender to two receivers directly using the local Advanced Adapter Engine if the schema is correct, the file will not be picked up if the structure does not fit.

7.1 Design Objects Used

In this variant the same interfaces are used as in variant 1. Note that no process integration scenario exists for this variant.

7.2 Configuring the Process Integration Scenario

When you configure this variant, you will extend the configuration settings from variant 4. You use the same configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE*.

Opening the Configuration Scenario

In this variant the configuration scenario *XiPatternOneStepScenario_dAAE_AAE2AAE* created in variant 1 and extended in variants 2 to 4 is extended to use XML-Validation to check the structure of the message.

For this scenario, again the integrated configuration object is used for configuration. The model configurator is not used.

Changing the Integrated Configuration

1. In the Integration Directory main menu, open the integrated configuration object <SID>_dAAE_FileSystem_XiPattern1 / XiPatternInterface1 / http://sap.com/xi/XI/System/Patterns
2. Switch to change mode (✎).
3. In tab *Inbound Processing* for Schema Validation select the radio button *Validation by Adapter*.
4. Save the Integrated configuration.

Activating the Change List

1. To do this, select the *Change Lists* tab in the Integration Directory navigation area.
2. Choose *Open* → *Standard Change List*.
3. Position the cursor on the *Standard Change List* node and choose *Activate* (⚙)) in the context menu.

7.3 Execution

You perform the following steps to check that the process integration scenario is executed without errors.

Execution without Validation Error

For the execution in directory *XiPattern1*, rename the file *ZXiPatternSenderFile1.xml* to *XiPatternSenderFile1.xml*. The file will be moved from the directory within the next 60 seconds.

Check in directory *XiPattern2* whether the file has arrived.

Furthermore, you can check the processing of the message in monitoring. To do this, proceed as follows:

1. Start the Monitoring by calling the *Configuration and Monitoring Home* on the de-central AAE: http(s)://<AAE-host>:<AAE-port>/pimon or use the *Runtime Workbench* as described in the Appendix..
2. Choose *Message Overview*.

3. Select *Database*.
4. Use appropriate filter criteria to restrict the number of XML messages displayed (for example, the processing period).
5. Choose *Go*.

   The system displays the selected messages in a table. You want to locate the messages that were sent during execution.
6. Select the message to be displayed by selecting the appropriate radio button and then choose *Open Message*.

   The tab *Payloads* displays the message contents for the various versions of the message during processing.

Message monitoring displays one message.

Execution with Validation Error

For the execution with wrong XML schema in directory *XiPattern1*, rename the file *ZXiPatternSenderFile3.xml* to *XiPatternSenderFile3.xml*. The file will not be moved from the directory within the next 60 seconds. No file should arrive in directory *XiPattern2.* The message-processing will not process the message because the message has the wrong XML-schema. If you check the XML file *ZXiPatternSenderFile3.xml,* you will see a tag with the name Email, this tag does not exist in the message type *XiPatternMessage1* used in the outbound interface *XiPatternInterface1*.

You cannot see the error in message monitoring because the adapter engine does not create a message out of the file because of the wrong structure. You can only see the error in channel monitoring. To do this, proceed as follows:

1. Start the Monitoring by calling the *Configuration and Monitoring Home* on the de-central AAE: http(s)://<AAE-host>:<AAE-port>/pimon or use the *Runtime Workbench* as described in the Appendix.
2. Select *Adapter Engine* → *Communication Channel Monitoring*.
3. Use the Link *Advanced* to get additional selection fields.
4. Set XiPatternChannel_FileSender as channel name and <SID>_dAAE_FileSystem_XiPattern1 as communication component to get only the channel used for this variant.
5. Choose *Use Filter*.

   The system displays the channel used for picking up the files from the sender directory. You will see that the channel is in error state because the XML-Validation found an error in the structure of the payload.

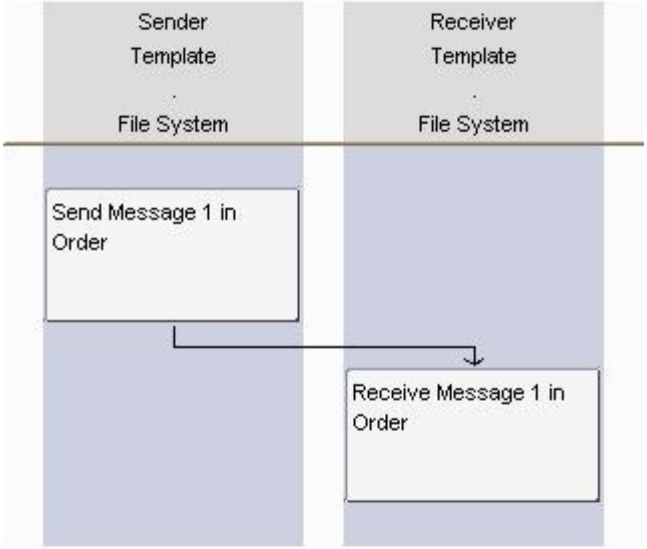
To end the processing delete the file *XiPatternSenderFile3.xml* from the directory.

8 Variant 6: One Sender and One Receiver using de-central Advanced Adapter Engine - using Exactly Once In Order

This variant enables you to configure and execute the simplest kind of message exchange: one sender on a de-central adapter engine (file directory *XiPatterns4*) sends a series of messages to one receiver on a de-central adapter engine (file directory *XiPatterns5*). No mapping is executed between the outbound and inbound interface in this variant. The messages are delivered in some particular order. Concretely, the file adapter allows either the last-modified timestamp or the filename to be used to order the input files in its directory.

8.1 Design Objects Used

You use the component view *File2File_NoMapping_InOrder* in process integration scenario *XiPatternOneStepScenario* to configure this variant. The following graphic shows the component view:



SAP NetWeaver Process Integration, Process Integration Scenario Editor

Interface Objects

In this variant, the data structures at both the sender and receiver side are the same and no mapping is required, the service interfaces *XiPatternInterface1O* is used for sender and receiver

Summary of Design Objects Used

The following table summarizes the design objects used.

Design Objects Used

Object Type (Name)	Description
Component View *File2File_NoMapping_InOrder* (Process Integration Scenario *XiPatternOneStepScenario*)	Describes the process flow and the message exchange for this variant.
Service interface (*XiPatternInterface1O*)	Describes the interface and data structure of the sender and receiver. The data structure is the same for sender and receiver.
Message Type (*XiPatternMessage1*)	
Data Type (*XiPatternDataType1*)	

8.2 Configuring the Process Integration Scenario

To configure this variant, a new configuration scenario is created.

Selecting the Process Integration Scenario

Perform the following steps to select the process integration scenario that you want to configure from the Enterprise Services Repository. Also, enter the name of the configuration scenario (in the Integration Directory) in which you want to group the configuration objects.

1. To call the model configurator, choose *Tools* → *Apply Model from ES Repository* in the Integration Builder menu bar.

   This calls a wizard, which you use to select a model (in this case, a process integration scenario) and create a configuration scenario in the Integration Directory.

2. Make sure that the model type *Process Integration Scenario* is selected (checkbox under *Type of ES Repository Model*).

3. Choose the input help (for the *Name* field) (□▾) and select *Process Integration Scenario* in the dropdown listbox.

   This calls an input help, which displays all process integration scenarios from the Enterprise Services Repository in a list.

4. Select the process integration scenario *XiPatternOneStepScenario* (software component version *SAP BASIS 7.31*, namespace *http://sap.com/xi/XI/System/Patterns*) from the Enterprise Services Repository and choose *Apply*.

5. Choose *Continue*.

6. Enter the name of the configuration scenario as: `XiPatternOneStepScenario_dAAE_EOIO`.

7. Choose *Finish* and then *Close*.

   The system calls the model configurator.

Selecting the Component View

Select component view *File2File_NoMapping_InOrder*.

Assigning Communication Component

Assign the communication component *<SID>_dAAE_FileSystem_XiPattern4* to the application component *Sender* and *<SID>_dAAE_FileSystem_XiPattern5* to the application component *Receiver*.

Configuring the Connection

You create the communication channel on the basis of the communication channel templates that the process integration scenario already contains.

1. In the *Configure Connections* screen, assign the sender a communication channel. To do so, place the cursor in the field *Communication Channel for the Sender Component*, in the push button bar choose □▾ → *Create Communication Channel With Template* (▨). In the subsequent wizard you can create a communication channel based on a communication channel template.

2. Choose *Continue*.

   The default communication channel template on the subsequent screen is that which was assigned to the sender application component in the process integration scenario (name *XiPatternChannel_FileSenderO*, software component version *SAP BASIS 7.30*, namespace *http://sap.com/xi/XI/System/Patterns*).

3. Choose *Continue*.

The default name for the communication channel on the next screen is *XiPatternChannel_FileSenderO*.

4. Confirm your selection by choosing *Finish*.

5. Following the same procedure (see step 4-7), assign the application component of the receiver a communication channel (*XiPatternChannel_FileReceiverO*) based on the default communication channel template (name *XiPatternChannel_FileReceiverO*, software component version *SAP BASIS 7.31*, namespace *http://sap.com/xi/XI/System/Patterns*).

Generating the Configuration Objects

To do so, proceed as described in variant 1.

When you have checked the generation log, close it and save the configuration scenario.

Manual Editing

In the following steps, you edit the generated configuration objects manually and enhance the configuration settings made during generation. In this variant, you must edit the communication channels after generation.

1. In the editor *Display Configuration Scenario* (for the open scenario *XiPatternOneStepScenario_EOIO*), on the tab page *Objects* open the communication channel (🔌) *XiPatternChannel_FileSenderO by* double-clicking.

2. Under *File Access Parameter* in the *Source Directory* field, enter the path in the file directory that you defined previously (see **Error! Reference source not found.**). In the case of the sender communication channel, enter the whole path for the file directory *XiPattern4*.
   Example: `C:/<selected path>/XiPatternsO/XiPattern4`

3. In the field *Adapter Engine* select the de-central adapter engine af.<SID>.<hostname>

   Example: `af.v4d.hostname`

4. Save the communication channel.

5. To check if the channel works you can directly jump to the channel monitoring using the menu path Communication Channel → Channel Monitoring

6. The PI Communication Channel Monitoring page will open up where you can do a self test for the channel. To do this click on the button *Ping Channel*.

7. The status page for the channel will show up with detailed information of the executed self test.

8. Following the same procedure, edit the communication channel *XiPatternChannel_FileReceiverO* (communication component *<SID>_dAAE_FileSystem_XiPattern5*). Select the de-central adapter engine and enter the path for the file directory *XiPattern5*.

9. Save the communication channel.

10. Check the status of the channel in the channel monitor.

11. Activate your change lists.

8.3 Execution

You perform the following steps to check that the process integration scenario is executed without errors.

1. Stop the channel *<SID>_dAAE_FileSystem_XiPattern4* | *XiPatternChannel_FileSender* in the *Configuration and Monitoring Home* on the de-central AAE: http(s)://<AAE-host>:<AAE-port>/pimon → Adapter Engine → Communication Channel Monitoring or use the *Channel Monitoring* in the *Runtime Workbench* to make sure no processing runs during the time the files are placed into the source folder.

2. On the host of the de-central Advanced Adapter Engine, open the file directory *XiPatternsO/XiPatterns5*. If the output file XiPatternReceiverFile.txt is already present, delete this file. The execution of this test will create this file if it is not present and append a series of output files to this file.

3. Open the file directory *XiPatternsO/XiPatterns4*

4. Unpack file *XiPatternSenderFilesEOIO.zip* in the current directory. This will place a series of input files *XiPatternSenderFile00001.xml, …, XiPatternSenderFile01000.xml*.

   The File/FTP adapter is configured so that the files are picked in the order sorted by the file names and moved from the source directory *XiPatterns4* to the target directory *XiPatterns5* and the files are concatenated into a single file *XiPatternReceiverFile.txt*.

5. Start the channel *<SID>_FileSystem_XiPattern4* | *XiPatternChannel_FileSender* in the *Configuration and Monitoring Home* on the de-central AAE: http(s)://<AAE-host>:<AAE-port>/pimon → Adapter Engine → Communication Channel Monitoring or use the *Channel Monitoring* in the *Runtime Workbench* to start processing the files.

6. Open the directory *XiPatternsO/XiPatterns5* and check that the files have been transferred.

7. Open the XML file *XiPatternReceiverFileAll.xml which includes the output file* and check whether the content is in the correct order.

   The content must be in the correct order with the incremental ID value and the entire content must be present.

8. Furthermore, you can check the processing of the messages in the message monitoring as in Variant 1.

9 Appendix

9.1 Monitoring Tools

9.1.1 Monitoring in Runtime Workbench

The message exchange can be monitored in the Configuration and Monitoring Home as described in the execution chapters and in the Runtime Workbench. To start the Runtime Workbench monitoring proceed as follows:

1. Start the Monitoring by choosing *Runtime Workbench* on the PI start page. (transaction **SXMB_IFR**)

2. Choose *Message Monitoring*.

3. Select *Adapter Engine <SID>* as the component and then choose *Display*.

4. Use appropriate filter criteria to restrict the number of XML messages displayed (for example, the processing period).

5. Choose *Start*.

   The system displays the selected messages in a table. You want to locate the messages that were sent during execution.

6. Select the message to be displayed by selecting the appropriate radio button and then choose *Details*.

   The tab page *Message Contents* displays the message contents including all message versions for the various phases of message processing.

9.1.2 Transaction SXMB_MONI in the PI System

The messages in the Integration Server can not only be monitored in the Runtime Workbench but as well in the SAP system of the Integration Server. To do this, proceed as follows:

1. In the Integration Server client, start *Exchange Infrastructure* → *Integration Engine: Monitoring* (transaction ***SXMB_MONI****)*
2. Select *Monitor for Processed XML Messages* and use the appropriate search criteria to restrict the number of XML messages displayed (for example time/date)
3. Choose *Start*

   The system displays the selected messages in a table. For more details select the line of the message and choose *Display*
4. The system displays the message in detail.

To display the payload before and after the mapping step you have to activate logging in the Integration server (RUNTIME parameter LOGGING in **SXMB_ADM** → Integration Server Configuration). IN this case, every step in the message processing will be persisted and you can check the payload for each of the steps. To compare the payload before mapping with the payload after mapping use the following procedure in the details screen of a message in transaction **SXMB_MONI**.

1. On the left side in the tree select *Inbound Message* → *Payloads* → *Main Document* and choose *Display contents in window 1.*

   The Inbound message payload is displayed.
2. Do the same procedure with *Receiver Determination --> Payloads --> Main Document* and choose *Display contents in window 2.*

   The Outbound message payload is displayed.

9.1.3 Monitoring in Solution Manager

The messages can also be monitored centrally in the Solution Manager System. To do this, proceed as follows:

1. Log into the connected SAP Solution Manager System and call the Solution Manager Workcenters using transaction **SOLMAN_WORKCENTER**.
2. Change to Tab *Technical Monitors* and follow the link to *PI Monitoring*.
3. Select the PI Domain you want to monitor (for example B4X).
4. Launch the Message Monitor either embedded or in a new window using the button *Message Monitor*.