Table of Contents

1  What’s New in SAP NetWeaver Process Integration .................................................. 3

2  Support Package Stack 19. ......................................................................................... 4

2.1  Java Support Package Manager (JSPM) obsolete (Changed). .............................. 4

2.1.1  Packaging and Deployment of Adapters. .................................................... 4

2.1.2  Providing External Drivers for the JDBC and JMS Adapters. ....................... 5

2.1.3  Support Package Stacks and Notes Implementation. ................................... 7

2.1.4  SAP-Specific Deployment as SCA Using JSPM. ........................................... 7

2.2  Archiving and Deleting (Changed). ................................................................. 7

2.2.1  Archiving and Deleting. ............................................................................... 8

2.3  Integration Builder (Enhanced). ........................................................................ 12

2.3.1  Integration Builder. ..................................................................................... 12

2.4  Jobs for Event-Driven Message Processing (Enhanced). .................................. 13

2.4.1  Jobs for Event-Driven Message Processing ................................................ 13

2.5  Adding Code Example from Adapter Development to Module Processor (Changed). .................................................. 16

2.5.1  Adding Code Example from Adapter Development to Module Processor. ........ 16

2.6  XSLT Mapping (Changed). .............................................................................. 17

2.6.1  XSLT Mapping. .......................................................................................... 17

2.7  Testing the Configuration (Enhanced). ............................................................ 20

2.7.1  Testing the Configuration. ........................................................................... 21
1 What’s New in SAP NetWeaver Process Integration

Release Notes SAP NetWeaver Process Integration 7.1

This section of the SAP Library gives you an overview of the new features and functions in SAP NetWeaver Process Integration 7.1. from Support Package 19 on. It includes Release Notes that describe what is new, enhanced, changed or deleted.

Note

SAP NetWeaver Process Integration 7.1, Support Package Stack 18 was the last support package stack with complete documentation publishing. As of Support Package Stack 19, only new or changed topics are published.
2 Support Package Stack 19

This documentation contains the Release Notes of Support Package Stack 19.

2.1 Java Support Package Manager (JSPM) obsolete (Changed)

The Java Support Package Manager (JSPM) is no longer supported. JSPM is replaced by the Software Update Manager (SUM).

Related Information

Packaging and Deployment of Adapters [page 4]
You can package and deploy adapters.

Providing External Drivers for the JDBC and JMS Adapters [page 5]
You can provide external drivers for JDBC and JMS adapters as Java archives (JARs).

Support Package Stacks and Notes Implementation [page 7]
You can implement Support Package Stacks and Notes.

SAP-Specific Deployment as SCA Using JSPM [page 7]
The Java Support Package Manager (JSPM) is no longer supported. The documentation is not longer valid.

2.1.1 Packaging and Deployment of Adapters

You can package and deploy adapters.

Use

Adapter deployment follows the JCA specification.

NOTE

The references refer to chapter 10 of the document J2EE Connector Architecture Specification, Final Version 1.0. You can download this document at java.sun.com/j2ee/connector.

Third-party adapters are deployed as follows:

- In the stand-alone variant, if all functions are implemented in the adapter and the default Adapter Framework is used
- In the bundled variant, if you provide your own Enterprise JavaBeans that are used in your own Adapter Framework module chain definition. A bundled deployment scenario of the Adapter Framework with a third-party adapter is not supported.
The JCA specification allows two mechanisms for the ConnectionFactory JNDI lookup, based on a serializable class or on a referenceable class. Although the Adapter Framework only uses one of the two classes, you must adhere to the JCA specification and implement both.

Integration

There is an example for each of the mechanisms described below:

- RAR (Resource Adapter Archive) deployment
  More information: com.sap.aii.adapter.sample.ra.rar in SAPXIAP<SP>_<HF>.sca
  <SP> stands for Support Package number; <HF> stands for hotfix number.

- EAR (Enterprise Archive) deployment
  Not supported at present

- SCA (Software Component Archive) deployment
  More information: sample_ra.sda in SAPXIAP<SP>_<HF>.sca

Features

The following deployment options are available:

- Stand-Alone Deployment as RAR
- Bundled Deployment as EAR

Activities

- You use the Software Update Manager (SUM) for the deployment. SUM is part of the Software Logistics Toolset shipment and is available as a download at http://service.sap.com >> Support Portal >> Software Downloads >> Search for Software Downloads >> Search Term: Software Update Manager
  You can find the SUM documentation on SAP Service Marketplace at http://service.sap.com/sitoolset >> Software Logistics Toolset 1.0 >> Software Update Manager (SUM)

- It is easier for development to use RAR files, especially when they are using the Development Studio deploy view. You can deploy the RAR file immediately, without having to use any other tools. SAP may provide tools in the future that enable you to convert RAR files to SCA files without the SAP-specific deployment descriptors being created automatically.

  Note

  Read SAP Note 1004000 to find out whether such tools are available.

For context information refer to the support package 18 version of this document.

2.1.2 Providing External Drivers for the JDBC and JMS Adapters

You can provide external drivers for JDBC and JMS adapters as Java archives (JARs).

Use

For a scenario involving communication with a database or a messaging system, you need external drivers for the JDBC and JMS adapters. These drivers must be provided as Java archives (JARs) by the provider of the database or messaging system, respectively.
You have to deploy these drivers to enable them to be used by the adapters on the SAP J2EE Server. For this purpose, the Adapter Engine installation provides the archive `com.sap.aii.adapter.lib.sda`, to which you have to add the respective driver files.

The SDA file format required for the deployment is basically a Zip-file format that contains additional text files, called descriptors, which describe the deployable library content (the JAR files packed in the SDA). Therefore, adding a JAR file to the archive means adding the JAR file with a Zip tool and adding its name to the `<jars>` element in the `provider.xml` file contained in the SDA. For the latter, you need to unzip the `provider.xml` file from the SDA, add the name to the correct section, and zip it again to the archive with the correct path information. To do so, proceed as described in the following sections.

For the JMS Driver only

Prepare the JMS provider file as follows:

1. Copy the file from the provider directory to a separate directory (for SonicMQ, for example, this file is called `client.jar` and is located in the lib subdirectory of the SonicMQ installation).
2. Open the file with a zip program and remove the standard JMS classes. These are located in directory `javax/jms`.
3. Save the modified provider file.

For the JDBC Driver and the JMS Driver

Add the driver to the `com.sap.aii.adapter.lib.sda` archive as follows:

1. Use a Zip program to extract the `provider.xml` descriptor from the `com.sap.aii.adapter.lib.sda` archive.
2. Edit the `provider.xml` descriptor by adding the line `<jar-name>driver-jar</jar-name>` for each provider-specific JAR file in the `<jars>` section. 

Example

If you want to install the MaxDB JDBC driver and the SonicMQ JMS driver, this section would look like this:

```
<jars>
    <jar-name>sapdbc.jar</jar-name>
    <jar-name>client.jar</jar-name>
</jars>
```

3. Return the changed `provider.xml` descriptor to the archive.

**Note**

Make sure that the original META-INF\ and server\ directories are retained under Path in the Zip archive (check for the latest Path entry).

4. Add all JARs (without path information) that you have defined in `provider.xml` to `com.sap.aii.adapter.lib.sda`.

You find `com.sap.aii.adapter.lib.sda` in directory `DATA_UNITS\JAVA_J2EE_OSINDEP_J2EE_INST` on your SAP NetWeaver 7.10 PI Java installation DVD.

5. Use the Software Update Manager (SUM) to deploy `com.sap.aii.adapter.lib.sda`. See the SUM documentation on SAP Service Marketplace at `http://service.sap.com/sitoolset`.

6. Restart AS Java. The drivers are now known to the JMS adapter or JDBC adapter and can be used as described.

For context information refer to the support package 18 version of this document.
2.1.3  Support Package Stacks and Notes Implementation

You can implement Support Package Stacks and Notes.

Updates can be performed using standard procedures. To update both the ABAP parts of PI and the Java parts, use the Software Update Manager (SUM).


You can find currently available support packages on SAP Service Marketplace at [service.sap.com/sp-stacks](http://service.sap.com/sp-stacks).

A description of how to apply support packages is available in SAP Note [1119255](http://service.sap.com). The SAP Note assistant is supported for the ABAP parts of PI.

For context information refer to the support package 18 version of this document.

2.1.4  SAP-Specific Deployment as SCA Using JSPM

The Java Support Package Manager (JSPM) is not longer supported. The documentation is not longer valid.

The following document was deleted: SAP-Specific Deployment as SCA Using JSPM

2.2  Archiving and Deleting (Changed)

You can archive and delete messages.

A wrong parameter name was corrected: SAP_BC_XMB. The correct parameter name is SAP_BC_XMB1.

Related Information

Archiving and Deleting [page 8]

You can archive and delete messages.
2.2.1 Archiving and Deleting

You can archive and delete messages.

Use

- Archiving and deleting messages:
  Messages that have been processed correctly are deleted by default. You must archive all messages that are not to be deleted.
  To archive messages, you must first define their corresponding interfaces and then schedule the archiving.

  **Note**
  Since correctly processed messages are flagged as either *To Be Archived* or *To Be Deleted* (and because this indicator cannot be changed), you must decide which messages you want to archive before the first message is processed.
  Messages that have been modified or cancelled manually are archived automatically.

You require two archiving jobs to archive messages:
- A job to write the messages to the archive
- A job to delete the archived message

Only one job is required to delete messages.
You can schedule all jobs periodically, but you must maintain the job sequence.
You can select and display archived messages.

- Deleting history entries for processed messages
  History entries are spots for observing message processing. They are generated by persisting at the end of a processing step for a message and contain the current status and the execution date.
  History entries remain in the database for an unspecified length of time and must be deleted at some stage so that the database table does not overflow. The deletion process only deletes history entries for messages that have already been archived or deleted. The history entry is kept in the database for at least seven days after a message has been deleted. This is necessary since history entries are also required for the quality of service *Exactly Once*.
  The default retention period is 30 days. You can change this value in the configuration of the Integration Engine by using the HISTORY subparameter of the PERSIST_DURATION parameter in the DELETION category.

Prerequisites

To archive messages, the following prerequisites must be fulfilled:

- You have defined the interfaces for the messages to be archived by choosing Configuration ➔ Define Interfaces for Archiving and Retention Periods in the Integration Engine menu.
  To archive messages, use the archiving object BC_XMB and use SAP_BC_XMB1 as the archiving infostructure. If you want to read the archive, the archiving infostructure must be activated.
You have used transaction Archive Administration (SARA) to set up archiving for the archiving object BC_XMB to archive messages.

You have used transaction Archive Information System (SARI) to activate the archive infrastructure SAP_BC_XMB1 to read archived messages.

To delete messages, the following prerequisites must be fulfilled:

- You have defined the deletion procedure required to delete messages by choosing Configuration Configure Delete Job from the Integration Engine menu.

  The Simple Deletion Procedure is set as the default. No steps are required to activate the deletion procedure. If you want to use the Switch Procedure, select the Switch Procedure Activated checkbox. You can do this at any point. The delete jobs react correspondingly, and with immediate effect. However, you can only deactivate the switch procedure when the original tables are active and the counter for the number of deleted records in the original tables is at zero. If this is not the case at the time of configuration, this is noted by the system and the switch procedure is deactivated the next time that table entries are copied from the table copies to the original tables.

  For more information about the simple deletion procedure and the switch procedure, see the documentation, which is displayed when you call the function.

To delete history entries, the following prerequisites must be fulfilled:

- You have already deleted or archived the messages that belong to the history entries because you cannot delete history entries for messages that are still in the database.

Scheduling Archiving Jobs

You schedule an archiving job, which in turn schedules the jobs to write the message to the archive and delete the archived messages. To do this, proceed as follows:

1. In the Integration Engine Administration menu, choose Schedule Archiving Job. The system displays the screen Integration Engine: Archiving.

2. Specify the following.

   **Archiving Specifications:**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Current client</td>
</tr>
<tr>
<td>Parallel Processing</td>
<td>Level of parallel processing</td>
</tr>
<tr>
<td></td>
<td>You can specify the level of parallel processing by using the configuration parameter ARCHIVE_PARALLEL in the ARCHIVE category.</td>
</tr>
</tbody>
</table>

   ✉️ **Note**

   If the value is greater than 1, parallel archiving takes place, which generally speeds up the archiving process.
### User Name

Your user name

You can also enter a different user name that has the corresponding authorizations.

### Start Time

The start time of the archiving action

The system displays a dialog box, where you define the start time and maintain the parameters in the corresponding fields.

If jobs are to be executed periodically, choose Period values to enter the start values.

To enter restrictions for the start time and date, choose Restrictions.

**Note**

The start time only applies to the execution of the first action. It is retained until the end of the archiving session.

### Spool Parameters

Controls the print output of the generated archiving log

The system displays a dialog box in which you maintain the general parameters Output Device, Number of copies and Number of pages.

You can choose Properties to maintain additional properties of the spool request. You do not generally change these properties for each spool request.

**Note**

Spool parameters that are already specified in the user defaults or in the user-specific print parameters are automatically applied. Otherwise, you only have to specify the spool parameters for the execution of the first action in an archiving session. They are retained until the end of the archiving session.

### Instructions

3. To schedule your job, choose Schedule Archiving.
4. To display the scheduled job in the job overview, choose Job Overview.
5. To use the SAP archive administration options, choose Archive Administration and then enter the archiving object BC-XMB in the Object name field.
Note

Do not schedule your write or delete jobs by using archive administration, but use the Integration Engine archiving functions.

The following options are also available:

- To read XML messages that have already been archived, choose **Read**. You can start the read program for your archiving object either in the background or in dialog mode. The latter corresponds to choosing ** Archived XML Messages (Search Using Archive)** in the **Integration Engine > Monitoring** menu.
- To display an overview of archiving sessions that have already run for your archiving object, choose **Management**.
- To display statistics for the individual archiving sessions, choose **Statistics**.

⚠️ Caution

For performance reasons, the statistics values are generally imprecise.

6. To migrate archive administration data, choose **Archive Migration**.

**Scheduling Delete Jobs**

To schedule delete jobs, proceed as follows:

1. In the **Integration Engine > Administration** menu, choose **Schedule Delete Jobs**.
2. Select the job(s) to be scheduled.
3. Specify the start time and date.
4. Specify the period you want to use and choose **Schedule**.

For an overview of all jobs, choose **Jobs**.

For context information refer to the support package 18 version of this document.

**Related Information:**

- Selecting Archived Messages
- Displaying Message Versions
- Defining Interfaces and Retention Periods for Archiving
- Displaying/Changing Configuration Data
- Archiving Procedure
- Archive Information System
- Reading Archive Files
- Archive Administration: Archiving Session Overview
- Statistics
- Migrating Archive Administration Data
2.3  Integration Builder (Enhanced)

Use the Integration Builder to define and manage the objects in the Integration Directory.

Information was added, how to access the Integration Builder.

Related Information

Integration Builder [page 12]

Use the Integration Builder to define and manage the objects in the Integration Directory.

2.3.1  Integration Builder

Use the Integration Builder to define and manage the objects in the Integration Directory.

Purpose

You use the Integration Builder to define and manage the objects in the Integration Directory.

To access the Integration Builder, choose \[ http://<host>:\<port>\dir\start/index.jsp \] Integration Builder.

Implementation Considerations

This section of the documentation describes the cross-object-type functions of the Integration Builder.

The fundamental tasks that you can perform with the Integration Builder are described in Tasks.

Functions that are specific to particular object types are described separately in the documentation for the individual object types.

Note

To call the online help from the SAP library, in the main menu, choose \[ Help \] Application Help.

To call the context-sensitive help for particular object types, in the menu bar of the object editor, choose \[ Help \] Application Help.

You call the Integration Builder from the start page of usage type Process Integration.

For context information refer to the support package 18 version of this document.

Related Information

- User Interface
- Tasks
- Defining Configuration Objects
- Working with the Development Environment
2.4 Jobs for Event-Driven Message Processing (Enhanced)

You can define jobs for scheduling the processing of messages or message packages.
Special authorizations were added for executing jobs immediately.

Related Information

Jobs for Event-Driven Message Processing [page 13]
You can define jobs for scheduling the processing of messages or message packages.

2.4.1 Jobs for Event-Driven Message Processing

You can define jobs for scheduling the processing of messages or message packages.

Use

This function enables you to define jobs for scheduling the processing of messages or message packages that have been sorted by a filter and not processed immediately.

Integration

Together with the functions for defining the required senders and receivers and the corresponding message filters, this function gives you the option of event-controlled message processing. You can stop specific messages and postpone processing until a later event occurs.

If the event-driven message processing functions of PI are not sufficient, you can use the more extensive job scheduling options provided by usage type AS-ABAP.

In this case you have to use the ABAP Editor (transaction SE38) and program SXMS_START_JOBS to schedule a job whose name corresponds to the job ID.

Before scheduling, the status of the job is To Be Scheduled; after scheduling, the status is Scheduled Externally.

Prerequisites

You have chosen Configure Event-Driven Message Processing from the Integration Engine menu and then Jobs. You are on the Schedule Message Processing screen.
Features

A job starts as soon as
- the specified criteria are met
- the job is active or scheduled externally
- the scheduler job, which schedules the jobs for message processing is running

A traffic light above the job list shows the status of the scheduler job:

<table>
<thead>
<tr>
<th>Traffic Light</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>The scheduler job is scheduled and is running</td>
</tr>
<tr>
<td>Yellow</td>
<td>The scheduler job is scheduled but overdue.</td>
</tr>
<tr>
<td>Red</td>
<td>The scheduler job is not scheduled.</td>
</tr>
</tbody>
</table>

If necessary, you can reschedule the scheduler. Choose Scheduler Overview.

The Number column in the job list shows how many messages have been stopped by message filters and assigned to the job. You can display these messages individually.

Activities

You can add, change, delete, update, and reset jobs. You can also (de)activate jobs and execute them immediately. You can display details for a job.

To execute a job immediately, you require a special authorization:

```
AUTHORITY-CHECK OBJECT 'S_XMB_AUTH'
ID 'SXMBAREA' FIELD 'JOB'
ID 'ACTVT' FIELD '16'.
```

This authorization and the common authorization to release are part of the SAP_XI_ADMINISTRATOR_ABAP role (ABAP composite role SAP_XI_ADMINISTRATOR).

The job is executed immediately in the background with the name of the job ID.

Adding a Job

When adding a new job, specify the following criteria:
- Job ID (must not begin with the letter S)
- Start date and start time (time stamp) of the job
- Latest possible start date (end date) and latest possible start time (end time) of the job
- Job repetition period
  The specified start time and date are recalculated after each period.
- Time unit of the repetition period
- Brief description of the job

All entries are optional except for the job ID.
Changing a Job

You can change all the criteria except for the job ID by using change mode.

If a job ID was created when the message filter was defined, you can add further specifications for the job in change mode.

(De)activating a Job

There are two ways to (de)activate a job:

- Use the Activate and Deactivate functions. The job is then controlled by the scheduler job.
- Control the (de)activation of a job by using an RFC-enabled function module (SXMS_JOB_ACTIVATE or SXMS_JOB_DEACTIVATE).
  In this case we recommend that you do not make any time specifications during job definition, so that the chronological sequence can be driven by an external event.

Scheduling a Scheduler Job

To start the periodic scheduler job that performs a client-specific check of the criteria that are used to start a job for event-driven message processing, choose Schedule Scheduler.

When you schedule the scheduler job (for example, if the traffic light for the scheduler status is red) you can enter the corresponding data in the dialog box.

If the scheduler job is already scheduled (the traffic light for the scheduler status is green), a corresponding message appears.

Displaying the Scheduler Log

To display the entire scheduler log, choose Display Scheduler Logs. This takes you to a standard application log, where you can search in the message texts for errors or for IDs of processed messages, for example.

Displaying Messages

If the number of messages stopped by a message filter and assigned to the job is greater than 0, you can list the messages individually by double-clicking the number displayed under Messages per Job. This displays the message ID and the quality of service (QoS). You can display each message in detail.

You can also navigate to message monitoring and Display All Messages or display individual messages in detail (by selecting the message ID).

For context information refer to the support package 18 version of this document.

Related Information

- Defining Sender/Receiver IDs
- Message Selection Filter
- Event-Driven Message Processing
- Quality of Service
- Monitoring XML Messages
2.5 Adding Code Example from Adapter Development to Module Processor (Changed)

You can use an example module for adapter and module development provided by SAP as a basis for your module development.

A wrong parameter was corrected. The Convert parameter is replaced by Mode.

Related Information

Adding Code Example from Adapter Development to Module Processor [page 16]

You can use an example module for adapter and module development provided by SAP as a basis for your module development.

2.5.1 Adding Code Example from Adapter Development to Module Processor

You can use an example module for adapter and module development provided by SAP as a basis for your module development.

SAP provides an example module for adapter and module development, which you can use as a basis for your module development. The example module provided by SAP converts the character that indicates the end of a line in Microsoft Windows, CarriageReturn+LineFeed (CRLF), to the character that indicates the end of a line in Unix, LineFeed (LF), or the other way around.

Add the Module in the Processing Sequence

1. Enter the module name SAP XI Sample/ConvertCRLFfromToLF before the module name of the adapter.
2. Select the module type Local Enterprise Bean.
   The system proposes a Module Key.

Add Parameters in the Module Configuration

Under Parameter Name, enter mode.

You can enter the following parameter values:

- none
  No conversion
- CRLFtoLF
  It is converted from Microsoft Windows CRLF to Unix LF.
- No, or any, value.
  It is converted from Unix LF to Microsoft Windows CRLF.

For context information refer to the support package 18 version of this document.

Related Information
2.6 XSLT Mapping (Changed)

You can transform a XML structure into another one.

Up to now it was stated that the SAP XML Toolkit is not going to be supported in subsequent releases. This has been changed: The SAP XML Toolkit will continue to be supported in all releases to come. New customers however are advised to use Java Development Kit (JDK) for all new XSLT mapping programs.

Related Information

XSLT Mapping [page 17]

You can transform a XML structure into another one.

2.6.1 XSLT Mapping

You can transform a XML structure into another one.

Use

Interface descriptions are in the form of XML documents. XSL Transformation (XSLT) is a member of the XML family of languages. It describes how one XML structure is transformed into another XML structure.

Features

In releases lower than SAP NetWeaver Process Integration (PI) 7.1, XSLT programs are executed using the SAP XML Toolkit. The SAP XML Toolkit will be replaced by version 5 of the Java Development Kit (JDK). JDK 5 has the following advantages over the SAP XML Toolkit:

- Quicker execution of XSLT programs
- More advanced troubleshooting
- Greater scope of functions

The SAP XML Toolkit will continue to be supported in all releases to come. New customers however are advised to use Java Development Kit (JDK) for all new XSLT mapping programs. Therefore, SAP recommends that you use JDK 5 for all new XSLT mapping programs, and that you test any existing XSLT programs with JDK 5 and modify...
them accordingly. To use the JDK 5 as the runtime for XSLT mapping programs, do not select Use SAP XML Toolkit in the operation mapping that you reference the XSLT mapping program from

Note
In SAP NetWeaver 2007, the PCK will only have JDK 5.

XPath and `<xsl:include>`

You can define mappings using XSLT together with XPath. XPath is also a specification of the XML family. Using XPath you can address any node in an XML document. XSLT implements XPath expressions to select substructures of an XML document. Using templates in XSLT you can define the mapping rules for the selected substructures.

You can use the XSLT tags `<xsl:include>` and `<xsl:import>` to include predefined templates for substructures in a complete mapping definition. In this way you can reuse mappings for data types.

Runtime Constants

In the same way as in Java mappings, the mapping runtime in XSLT programs also returns parameters that you can use to evaluate information in the XSLT program that is not known until runtime.

Mapping Runtime Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MessageClass</td>
<td>Classification of message. Possible values:</td>
</tr>
<tr>
<td></td>
<td>● ApplicationMessage: Asynchronous or synchronous request message</td>
</tr>
<tr>
<td></td>
<td>● ApplicationResponse: Response to a request message</td>
</tr>
<tr>
<td></td>
<td>● SystemAck, ApplicationAck, SystemError, ApplicationError: Acknowledgment Messages</td>
</tr>
<tr>
<td>$VersionMajor</td>
<td>XI message protocol version. Example: For the XI 3.0 message protocol VERSION_MAJOR = 3 and VERSION_MINOR = 0.</td>
</tr>
<tr>
<td>$VersionMinor</td>
<td></td>
</tr>
<tr>
<td>$ProcessingMode</td>
<td>The mode of a message can be synchronous or asynchronous. Correspondingly, these constants can have the value synchronous or asynchronous.</td>
</tr>
<tr>
<td>$MessageId</td>
<td>The message ID. It can change during communication:</td>
</tr>
<tr>
<td></td>
<td>● Response messages get a new message ID.</td>
</tr>
<tr>
<td></td>
<td>● If new messages result from a message (the message is copied at multiple receivers), the new messages get new message IDs.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$RefToMessageId</td>
<td>The ID of a referenced message that belongs semantically to this message. For example, a response message uses this field to note which request message it belongs to.</td>
</tr>
<tr>
<td>$ConversationId</td>
<td>This field is not mandatory in the message. It enables an ID to be used to group messages that belong together. This field is not intended to be used for message serialization and has nothing to do with the serialization context (ABAP proxy runtime, Java proxy runtime).</td>
</tr>
<tr>
<td>$TimeSent</td>
<td>Time stamp specifying when the message was sent by the sender. The format of the time stamp is as follows: YYYY-MM-DDTHH:MM:SSZ. The letter 'T' separates the date from the time, which is generally specified in UTC. If it is a local time, the closing 'Z' is omitted.</td>
</tr>
<tr>
<td>$Interface</td>
<td>Sender interface name. As of SAP XI 3.0, use this constant instead of the constant $SEND_NAME used previously.</td>
</tr>
<tr>
<td>$InterfaceNamespace</td>
<td>Sender interface namespace. As of SAP XI 3.0, use this constant instead of the constant $SEND_NAMESPACE used previously.</td>
</tr>
<tr>
<td>$SenderParty</td>
<td>Communication party that sent the message.</td>
</tr>
<tr>
<td>$SenderPartyAgency</td>
<td>Issuing agency for the message sender.</td>
</tr>
<tr>
<td>$SenderPartyScheme</td>
<td>Identification scheme used by the sender.</td>
</tr>
<tr>
<td>$SenderService</td>
<td>Service on the sender side that sent the message. For example, the name of a business system. As of SAP XI 3.0, use this constant instead of the constant $SEND_SYSTEM used previously.</td>
</tr>
<tr>
<td>$ReceiverName</td>
<td>Receiver interface name.</td>
</tr>
<tr>
<td>$ReceiverNamespace</td>
<td>Receiver interface namespace.</td>
</tr>
<tr>
<td>$ReceiverParty</td>
<td>Communication party to receive the message.</td>
</tr>
<tr>
<td>$ReceiverPartyAgency</td>
<td>Issuing agency for the message receiver.</td>
</tr>
<tr>
<td>$ReceiverPartyScheme</td>
<td>Identification scheme used by the receiver.</td>
</tr>
</tbody>
</table>
$ReceiverService  Service on the receiver side that receives the message. 
*For example, the name of a business system.*
As of SAP XI 3.0, use this constant instead of the constant `RECEIVER_SYSTEM` used previously.

$MappingTrace  Returns an *AbstractTrace Object* that you can use to write messages to monitoring in the mapping program.

If you want to access one of the constants in the XSLT program, you first have to declare the constant as a parameter, for example:

```xml
<xsl:param name="MessageId"/>
```

**Java Enhancements**

Moreover, using an XSLT definition you can call external Java methods to convert XML structures. This procedure gives you more flexibility when defining mappings.

**Note**

For more information about XSL/XSLT and XPath, see the Web page of the [http://www.w3c.org/](http://www.w3c.org/).

**Example**

The function for booking a flight contains a parameter for the flight class. The outbound interface represents flight classes as symbolic values; in the inbound interface, flight classes have numerical values. The XSLT section converts the symbolic representation of the source system into the numerical representation of the target system.

The XPath expression `match="//hh:class" select="<hh:class>"` selects the node `<hh:class>` in the outbound interface. The XSLT statement `<xsl:choose>` then defines the mapping rule for how the Process Integration runtime converts the symbolic values in the source system into the numerical values in the target system.

For context information refer to the [support package 18 version](#) of this document.

**Related Information**

[XSLT Mapping with Java Enhancement](#)

### 2.7 Testing the Configuration (Enhanced)

You can test the configuration by simulating the message processing.

The ABAP composite role `SAP_XICONFIGURATOR` was added as a prerequisite to call the configuration test.
Related Information

Testing the Configuration [page 21]
You can test the configuration by simulating the message processing.

2.7.1 Testing the Configuration

You can test the configuration by simulating the message processing.

You can simulate the processing of a message on the basis of existing configuration data. You can enter the header and payload of the message as the input parameters. You can display and analyze the status of the message after each individual sub-step and once the message has been fully processed.

Note
The test will only include active configuration objects.

Note
At present, the configuration test does not support adapter-specific attachments.

During the configuration test, the individual pipeline services are called, and the change to the message in each case is simulated in the individual processing steps.

Prerequisites

To call the configuration test, in the Integration Builder main menu, choose Tools Test Configuration
Therefor you need the ABAP composite role SAP_XI_CONFIGURATOR.

Features

The tool for the configuration test comprises the following screen areas:

Table 1: Screen Areas and Functions

<table>
<thead>
<tr>
<th>Screen Area</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu bar</td>
<td>Calls the functions for starting and resetting the test.</td>
</tr>
</tbody>
</table>
| Test message and test results       | • Test Message tab page
                                                Entries for the header and payload of the test message.  
• Results tab page
                                                Displays the result messages from the test run for the entire pipeline or for individual processing steps. |
| Progress display                    | Display the steps that have already been processed.                      |
                                                Calls the functions for starting and resetting the test. |
<table>
<thead>
<tr>
<th>Screen Area</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Objects and Error Log</td>
<td>Detailed analysis of errors:</td>
</tr>
<tr>
<td></td>
<td>•  <strong>Configuration Objects</strong> tab page</td>
</tr>
<tr>
<td></td>
<td>Displays the configuration objects that are relevant for the selected messages.</td>
</tr>
<tr>
<td></td>
<td>•  <strong>Error Log</strong> tab page</td>
</tr>
<tr>
<td></td>
<td>Display the detailed error log (trace).</td>
</tr>
</tbody>
</table>

### Process Flow

**Define Input Parameters**

On the **Test Message** tab page, enter the header and, if applicable, the payload of the test message (in XML format).

**Note**

It only makes sense for you to enter a payload if you want to test some configuration settings that are independent of the application data. This includes mappings and routing conditions, both of are dependent on the contents of the payload.

You have the following options for specifying the fields of the header:

- Input help
- Drag and drop
  
  When you use the cursor to drag a logical-routing or collaboration-agreement object from the navigation area in the Integration Directory to the **Header** area on the **Test Message** tab page, the header fields of the message are filled with the values of the object key of the configuration object.

You have the following options for specifying the payload of the message:

- Manual entry
- Copy a message from the test tool of the mapping editor
- Copy a message from message monitoring

**Start the Test**

You start the test in the test-progress display. You can do the following:

- Start the test run for the whole pipeline (**Start**)
- Start the test run for a single processing step (**SingleStep**)

You can reset the test by choosing **Reset**.

The graphical test-progress display shows which steps in message processing have already been simulated. There are 3 results possible:

1. The step has executed all the messages successfully
2. All the messages had to be stopped due to an error
3. One or more messages (but not all) had to be stopped due to an error. In the case of the latter, the test can be continued; however, SAP recommends that you analyze the error.

**Display the Result Message(s)**

**Display the List of Result Message(s)**
The **Results** tab page displays all the test messages for either the whole test run or just the sub-step specified in the test-progress display.

For example, multiple result messages are created by a receiver determination with multiple configured receivers. The table lists all the messages that were created after the executed processing step. Here, you have the option of specifying which information is displayed for each message. To do so, choose **Maintain Table** and then select the columns that you want to display in the table. The following columns are available:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Displays the status of the result message.</td>
</tr>
<tr>
<td></td>
<td>● Message processed successfully</td>
</tr>
<tr>
<td></td>
<td>● Message not processed further due to an error. Check the <strong>Error Log</strong> tab page (see below) to analyze the error further.</td>
</tr>
<tr>
<td></td>
<td>● Message processed successfully by entire pipeline (once the whole pipeline has been tested).</td>
</tr>
<tr>
<td>Stopped At</td>
<td>Only if the message was not processed further: Displays the configuration object (icon) in which the error occurred. For more information about the configuration object and the error, see the <strong>Error Log</strong> tab page (see below).</td>
</tr>
<tr>
<td>Sender</td>
<td>Displays the header values of the result message.</td>
</tr>
<tr>
<td>Receiver</td>
<td>Displays the header values of the result message.</td>
</tr>
<tr>
<td>Interface</td>
<td>Displays the header values of the result message.</td>
</tr>
</tbody>
</table>

**Display the Payload of a Result Message**

In the table, select the message whose payload you want to display. The payload of the selected message is then displayed in the **Payload** frame.

**Troubleshooting**

**Display Relevant Configuration Objects**

The **Configuration Objects** tab page lists the configuration objects that define the processing of the message for the steps executed. The object key for each configuration object is displayed. To open the configuration object, double-click the relevant line of the table.

If a processing step creates multiple messages, you can use a filter function to restrict the display to leave only those configuration objects that are relevant for a particular message. To do so, activate the filter on the **Configuration Objects** tab page and select a message in the table (**Results** tab page).

**Display Detailed Error Log**

The **Error Log** tab page displays an error log for the test.

The error log contains the following:
A list of the configuration objects involved, in the order in which the message was processed. To open the configuration object, double-click the object key.

A trace entry for each processing step
You can expand the trace and display the individual trace entries.

Here too, you can open the listed configuration objects by navigating forwards.

For context information refer to the support package 18 version of this document.

Related Information

- Pipelines and Pipeline Services
- Test Environment
- Message Monitoring
- Legend
Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP’s gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP’s current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of wilful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP’s gross negligence or wilful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).