THE BEST-RUN E-BUSINESSES RUN SAP

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

Documentation in the SAP Service Marketplace
You can find this documentation at the following address:
http://service.sap.com/xxxxxxx

© Copyright 2005 SAP AG. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

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

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

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

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

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

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

Java is a registered trademark of Sun Microsystems, Inc.

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

MaxDB is a trademark of MySQL AB, Sweden.

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

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Text</td>
<td>Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation</td>
</tr>
<tr>
<td>Example text</td>
<td>Emphasized words or phrases in body text, titles of graphics and tables</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td>Example text</td>
<td>Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, source code as well as names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td>Example text</td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td>&lt;Example text&gt;</td>
<td>Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Keys on the keyboard, for example, function keys (such as ( f_2 )) or the ENTER key.</td>
</tr>
</tbody>
</table>

Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️</td>
<td>Caution</td>
</tr>
<tr>
<td>🍃</td>
<td>Example</td>
</tr>
<tr>
<td>🍂</td>
<td>Note</td>
</tr>
<tr>
<td>🔄</td>
<td>Recommendation</td>
</tr>
<tr>
<td>⌨️</td>
<td>Syntax</td>
</tr>
</tbody>
</table>
Contents

1 Introduction................................................................................................................................. 9

2 Prerequisites ................................................................................................................................. 10

3 Preparation ................................................................................................................................ 11

4 Configuration of a Central Integration Server .......................................................................... 13
   4.1 Configuring the Role of the Integration Server ................................................................. 14
   4.2 Activating the HTTP Service on the Integration Server .................................................. 15
   4.3 Defining HTTP Destinations for Business Systems ......................................................... 15
   4.4 Registering Queues Used by the Integration Server ......................................................... 16
   4.5 Checking the Connection Between the Integration Server and the System Landscape Directory ................................................................................................................................. 16
      4.5.1 Checking the SAP J2EE Engine Configuration ............................................................... 16
   4.6 Enabling an Automatic Customizing of the Business Process Engine ........................................... 16

5 Configuring the Change Management Service ........................................................................ 17
   5.1 Deploying the CMS Component ....................................................................................... 17
   5.2 Maintaining Integration Builder Service Users in CMS ..................................................... 17
   5.3 Configuring CMS Domain, CMS Server, and SLD .............................................................. 18
   5.4 Activating CMS in the Integration Builder ......................................................................... 19
      5.4.1 Integration Repository ................................................................................................. 19
      5.4.2 Integration Directory ................................................................................................. 20
   5.5 Defining the Tracks in CMS ................................................................................................. 20

6 Configuring Message Archiving for the Adapter Engine and the PCK ........................................ 21
   6.1 Configuring the Destination for the XML DAS Administration ......................................... 21
   6.2 Defining an Archive Store ................................................................................................. 21
   6.3 Synchronizing and Displaying the Home Path .................................................................... 22

7 Further Optional Configuration Steps ....................................................................................... 23
   7.1 Configuring and Starting the SLD Bridge ........................................................................... 23
   7.2 Configuring the Integration Builder Documentation Editor ............................................. 23
      7.2.1 Defining New Base Locations ....................................................................................... 24
      7.2.2 Editing or Deleting Base Locations ............................................................................... 24
      7.2.3 Importing/Exporting the Base Location ....................................................................... 25
      7.2.4 Examples ..................................................................................................................... 25

8 Connecting Business Systems with an Integration Engine to the Central Integration Server .............. 26
8.1 Activating an HTTP Service on the Integration Engine (Receiver Side) 29
8.2 Registering Queues Used by the Local Integration Engine .......... 30
8.3 Creating a Connection Between a Business System and the System Landscape Directory (SLD) ............................................................ 31
  8.3.1 Maintaining the HTTP Address of the SLD Server .................. 31
  8.3.2 Testing the Connection Between a Business System and the System Landscape Directory .......................................................... 35
8.4 Creating a User for the Runtime Workbench ................................ 35
8.5 Creating a Destination for the Integration Server for Receiver Pre-Identification (Optional) ........................................................................ 35
8.6 Creating a Destination for the Integration Server for Maintaining the Value Mapping Table (Optional) .................................................. 36

9 Integration of Business Systems Using the IDoc Adapter ....37
  9.1 Settings for the IDoc Sender System (R/3) ............................... 37
  9.2 Settings of the Integration Server System .................................. 38
  9.3 Settings for an SAP Receiver System .......................................... 42
  9.4 Further Hints and Information .................................................. 42
  9.5 Defining an IDoc Communication Channel .................................. 43

10 Integration of Business Systems Using the Plain J2SE Adapter Engine .............................................................................................................. 45
  10.1 Use .......................................................................................... 45
  10.2 Configuring the File/FTP, JDBC, JMS, and SOAP Adapters ....... 46

11 Integration using the Adapter Engine (J2EE) ......................... 46
  11.1 External Drivers for the JDBC and JMS Adapters ...................... 47

12 Communication and Security ..................................................... 49
  12.1 Single Sign-On Configuration for SAP XI ............................... 49
    12.1.1 Steps Required for XI Web Components ......................... 49
    12.1.2 Additional Steps Required for the Runtime Workbench ........ 50
  12.2 Single Sign-On Configuration for the PCK ............................ 51
  12.3 HTTPS Configuration for Messaging ........................................ 51
    12.3.1 Inbound (Sending Messages to the Adapter Engine or PCK) ........ 52
    12.3.2 Outbound (Sending Messages to an External System from the Integration Server through the Adapter Engine or PCK) .................. 53
    12.3.3 Strict hostname checking ...................................................... 53
  12.4 Security Configuration at Message Level .............................. 53
    12.4.1 Configuring Web Service Security in the ABAP Stack ............ 54
    12.4.2 Configuring Web Service Security in the Java Stack ............. 55
    12.4.3 Verifying the Web Service Security Configuration ............... 56

13 Java Proxy Runtime (JPR) ......................................................... 57
  13.1 Settings in the Exchange Profile ............................................. 57
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.2</td>
<td>Entries in the System Landscape Directory</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>13.3</td>
<td>Security Roles for the Java Proxy Runtime</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>13.4</td>
<td>Configuring an Inbound Service for the Java Proxy Runtime</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>User Management and User Roles</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>14.1</td>
<td>Dialog Users of SAP Exchange Infrastructure</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>14.1.1</td>
<td>Description of the Dialog User Roles</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>14.1.2</td>
<td>Making Changes to Dialog User Roles</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>14.2</td>
<td>Service Users of SAP Exchange Infrastructure</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>14.2.1</td>
<td>Description of the Service User Roles</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>14.2.2</td>
<td>Service Users in the Exchange Profile</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>14.3</td>
<td>Users with Data-Dependent Authorizations</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>14.3.1</td>
<td>Creating Users and Roles (User Groups)</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>14.3.2</td>
<td>Assigning Roles to User Groups</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>14.3.3</td>
<td>Assigning Unrestricted Roles to Predefined User Groups</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activating Data-Dependent Authorization Checks</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Central Monitoring</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>15.1</td>
<td>General Prerequisites</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>15.2</td>
<td>Configuring End-to-End Message Monitoring</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>15.3</td>
<td>Configuring Message-Based Alerting</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>15.4</td>
<td>Configuring the CCMS Alert Monitor for SAP Exchange Infrastructure</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>15.4.1</td>
<td>Creating a System Group for SAP Exchange Infrastructure</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>15.4.2</td>
<td>Enabling Performance Alerts for Specific Sender-Receiver Combinations</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>15.4.3</td>
<td>Configuring the CCMS Heartbeat Monitor for Java Components</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>15.4.4</td>
<td>Configuring CCMS Alerts for the Adapter Engine</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Exchange Profile Parameters</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>Appendix</td>
<td>Service Properties that may Require Changes</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2EE Service Properties</td>
<td>81</td>
<td></td>
</tr>
</tbody>
</table>
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
The purpose of this configuration guide is to describe the configuration steps required to set up SAP Exchange Infrastructure 3.0 within a system landscape that has already been installed using the corresponding Master Guide.

The SAP Exchange Infrastructure 3.0 Configuration Guide
• Covers processes and process steps across any number of mySAP.com application components as well as third-party products used by one or more business partners
• Describes the configuration settings that need to be made during implementation in each of the components of the communication scenarios
• Provides the sequence of configuration steps and their interdependencies

Intended Audience
The SAP Exchange Infrastructure 3.0 Configuration Guide is intended to be used by both technology and application consultants.

Structure
The structure of this configuration guide follows the sequence of steps required to configure and run SAP Exchange Infrastructure 3.0.

Additional Documentation
List of related documentation

<table>
<thead>
<tr>
<th>Title</th>
<th>Purpose</th>
<th>Where to find</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Guide – SAP NetWeaver '04</td>
<td>Starting point for implementing SAP NetWeaver™</td>
<td>SAP Service Marketplace at service.sap.com/instguides nw04</td>
</tr>
<tr>
<td>Installation Guide – SAP Exchange Infrastructure 3.0</td>
<td>Installing required SAP Exchange Infrastructure components</td>
<td>SAP Service Marketplace at service.sap.com/instguides nw04</td>
</tr>
</tbody>
</table>
SAP NetWeaver Library documentation: SAP Exchange Infrastructure

The SAP NetWeaver Library is the basis for understanding this configuration guide. It provides detailed information on each function.

SAP Help Portal at help.sap.com, under SAP NetWeaver

Related SAP Notes

Read the related notes before beginning the configuration. These notes contain the most recent information regarding the configuration, as well as corrections to the configuration documentation.

Make sure that you have the latest version of each note. You can find the SAP Notes on SAP Service Marketplace at service.sap.com/notes or by using the SAP Net – R/3 Frontend.

List of Related SAP Notes

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>723268</td>
<td>SAP Exchange Infrastructure 3.0 Configuration</td>
<td>This SAP Note is crucial for the configuration. It contains:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Any information relating to the SAP Exchange Infrastructure that was not available when this configuration guide was published.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Corrections to this SAP Exchange Infrastructure Configuration Guide.</td>
</tr>
</tbody>
</table>

The SAP Exchange Infrastructure Configuration Guide is updated on a regular basis on SAP Service Marketplace at service.sap.com/instguidesNW04.

Make sure that you have the latest version of the SAP Exchange Infrastructure Configuration Guide by checking SAP Service Marketplace before you begin the configuration.
1 Introduction

SAP Exchange Infrastructure (XI) 3.0 consists of the following components:

- Integration Server (IS)
  The Integration Server is a centralized Integration Engine (IE). The Integration Engine is the runtime component that integrates the IDoc adapter and the Plain HTTP adapter.
- Local Integration Engines (IE)
- Integration Builder Repository (IR) and Directory (ID)
- Runtime Workbench (RWB)
- System Landscape Directory (SLD)
- Adapter Engine (J2EE-based) and Adapter Framework
- Plain J2SE Adapter Engine
- Java Proxy Runtime
- Java Web Start

For more information about installation scenarios, see the Installation Guide – SAP Exchange Infrastructure 3.0.

The configuration guide contains all activities for configuring the central Integration Server, the local Integration Engines, and the adapters to process XML messages. The figure below provides an overview of the relevant software components.
## 2 Prerequisites

### List of required steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Documentation</th>
</tr>
</thead>
</table>
| 1.   | You have installed the Exchange Infrastructure.  
      | *Master Guide – SAP NetWeaver ’04*  
      | *Installation Guide – SAP Web AS 6.40 ABAP and Java on <Operating System>*  
      | *Installation Guide – SAP Exchange Infrastructure 3.0* |
| 2.   | You have installed the XI Add-On in each involved business system that is based on SAP Web AS 6.20.  
      | SAP Note 439915; if you generate ABAP proxies, see SAP Notes 675441 and 721160.  
      | Use SAP Web AS 6.20 kernel patch level 1253 if you want to generate new ABAP proxies in your business systems based on SAP Web AS 6.20. |
| 3.   | You have configured the System Landscape Directory (SLD).  
      | *SAP System Landscape Directory User Manual for SAP Web AS 6.40*  
      | For information about configuring the SLD, see also the SAP NetWeaver Library documentation under:  
      | *SAP Exchange Infrastructure → Design and Configuration Time → Configuration → System Landscape Directory in SAP Exchange Infrastructure* |
| 4.   | You have imported the required content for the Integration Repository (interfaces, mappings, and so on).  
      | *SAP Exchange Infrastructure → Design and Configuration Time → Software Logistics for XI Objects → Transporting XI Objects* |
| 5.   | You have executed the program RSXMB_ACTIVATE_ICF_SERVICES  
      | to activate all required ICF services.  
      | Use the Object Navigator (transaction SE80) to execute this program. |
3 Preparation

1. Before you can start to configure SAP Exchange Infrastructure, you have to assign yourself (and all those involved in the configuration process) the following roles on each host (SAP Web AS) that is involved:
   - SAP_XI_ADMINISTRATOR (mandatory)
   - SAP_XI_CONFIGURATOR (mandatory)
   - SAP_SLD_ADMINISTRATOR (mandatory for the SLD configuration)
   - SAP_SLD_CONFIGURATOR (mandatory for the SLD configuration)

   For more information about SLD roles, see the SAP System Landscape Directory User Manual for SAP Web AS 6.40.

   For test purposes, we recommend that you also assign the other roles that are available.

   User menus for the roles are available in the start menu of SAP Exchange Infrastructure (SAP Web AS, ABAP side). All the role-specific user menus contain a reference (transaction SXMB_IFR) to the XI start page which runs in the Web browser. From this XI start page you have access to the individual J2EE applications, the System Landscape Directory, the Runtime Workbench, as well as to the online documentation.

   For more information about users and roles, see chapter 14.

2. The following service users are created during installation. Check whether they have been created correctly on each involved host (if not, this might be due to an installation error).

   Service Users and Assigned Roles

<table>
<thead>
<tr>
<th>Service user</th>
<th>Description</th>
<th>Assigned role</th>
</tr>
</thead>
<tbody>
<tr>
<td>XIREPUSER</td>
<td>User for the Integration Repository</td>
<td>SAP_XI_IR_SERV_USER</td>
</tr>
<tr>
<td>XIDIRUSER</td>
<td>User for the Integration Directory</td>
<td>SAP_XI_ID_SERV_USER</td>
</tr>
<tr>
<td>XIAPPLUSER</td>
<td>User for sender applications</td>
<td>SAP_XI_APPL_SERV_USER</td>
</tr>
<tr>
<td>XILDUSER</td>
<td>User for the System Landscape Directory</td>
<td>SAP_BC_AI_LANDSCAPE_DB_RFC</td>
</tr>
<tr>
<td>XIRWBUSER</td>
<td>User for the Runtime Workbench</td>
<td>SAP_XI_RWB_SERV_USER</td>
</tr>
<tr>
<td>XIISUSER</td>
<td>User for the Integration Server</td>
<td>SAP_XI_IS_SERV_USER</td>
</tr>
<tr>
<td>XIAFUSER</td>
<td>User for the Adapter Engine (communication between SLD, Integration Server, and Adapter Framework)</td>
<td>SAP_XI_AF_SERV_USER_MAIN</td>
</tr>
</tbody>
</table>
Passwords are specified for these users during installation. The names of the users as well as the password may be changed as required. Ensure, however, that they are always assigned the roles as listed above.

If you make changes to user names and passwords, you have to make them in transaction SU01 first and then again in the exchange profile. For more information, see SAP Note 721584.

⚠️

For all J2EE applications (in other words, the Integration Repository, Integration Directory, Runtime Workbench, System Landscape Directory, and sender Java applications), name and password changes to any of the above users will not take effect until the system is restarted.

For sender ABAP applications, any name and password changes must also be made in the involved HTTP destinations.
4 Configuration of a Central Integration Server

Before configuring your central Integration Server, make sure that it is registered in the System Landscape Directory (SLD). For this purpose, the required Common Information Model (CIM) objects should have been instantiated in the SLD during the post-installation steps described in the Installation Guide – SAP Exchange Infrastructure 3.0.

- Use the Technical Landscape Wizard under System Landscape → Technical Landscape to describe your Integration Server technically.
- Use the Business System Wizard under System Landscape → XI Landscape to define the business system with the role of an Integration Server.


Furthermore, ensure that you created the following fixed RFC destinations during the post-installation steps:

- **AI_DIRECTORY_JCOSERVER** from the Integration Server to the Integration Builder. The program ID corresponds to the entry under JCo RFC provider for the SAP J2EE Engine (used for refreshing the runtime cache).
- **AI_RUNTIME_JCOSERVER** from the Integration Server to the mapping runtime. The program ID corresponds to the entry under JCo RFC provider for the SAP J2EE Engine (used for the mapping runtime, value mapping, and the exchange profile).
- **INTEGRATION_DIRECTORY_HMI** of type HTTP, used for the cache update.
  - Path prefix /dir/CacheRefresh
  - HTTP 1.0
  - Accept Cookies: Yes (All)

Always use user XIISUSER.
4.1 Configuring the Role of the Integration Server

As of Release 6.20, the SAP Web AS is able to exchange XML-based messages using the Exchange Infrastructure. For this purpose, it contains an Integration Engine as part of the Exchange Infrastructure. The Integration Engine can assume different roles, for example, the role of an Integration Server.

The Integration Server is configured as a central Integration Engine and is not able to execute application logic. The Integration Server receives XML messages, determines the receivers, performs mappings, and forwards the XML messages to the corresponding receiver systems.

Make sure that you assigned the role of the central Integration Server to an Integration Engine during the post-installation steps described in the Installation Guide – SAP Exchange Infrastructure 3.0.
Within the same function (transaction SXMB_ADM or Exchange Infrastructure: Administration → Integration Engine – Administration → Integration Engine Configuration) you can specify additional configuration data (settings for monitoring and error handling, for example). Check all parameter default values (especially for logging and tracing) to ensure that they meet your requirements. To do this:

1. Select the relevant category under Specific Configuration Data.
2. Choose Change Specific Configuration Data.
   
   The system displays the Configuration screen where you enter the configuration parameters that you require for the selected category.

For more information about the configuration, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Runtime → Integration Engine → Configuring the Integration Engine → Displaying/Changing Configuration Data.

### 4.2 Activating the HTTP Service on the Integration Server

To enable the Integration Server to receive XML messages using HTTP, use transaction SMICM (Goto → Services) to verify that you activated the corresponding HTTP service during the post-installation steps described in the Installation Guide – SAP Exchange Infrastructure 3.0.

You can activate all required HTTP services by executing the report RSXMB_ACTIVATE_ICF_SERVICES (see also SAP Note 517484). HTTP services are deactivated by default. Therefore, each time you change an HTTP service, you have to activate it again.

Once you have activated the service, the logon data (user, password, client, and language) is dynamically transferred in the HTTP header fields during an HTTP call of the service.

For security reasons we recommend that the logon data is transferred dynamically using an HTTP destination. This method ensures more secure communication than the use of explicit URLs.

### 4.3 Defining HTTP Destinations for Business Systems

You maintain the HTTP destinations for your business systems in the Integration Builder under Directory (Configuration).

For information about defining the HTTP destinations, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Design and Configuration Time → Configuration → Defining Collaboration Profiles → Communication Channel.
4.4 Registering Queues Used by the Integration Server

Use transaction SXMB_ADM (or choose Exchange Infrastructure: Administration → Integration Engine – Administration → Manage Queues) to make sure that you registered the queues for asynchronous message processing used by the Integration Server during the post-installation steps described in the Installation Guide – SAP Exchange Infrastructure 3.0.

For more information about these queues, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Runtime → Integration Engine → Processing XML Messages → Queues for Asynchronous Message Processing.

4.5 Checking the Connection Between the Integration Server and the System Landscape Directory

Call transaction SLDCHECK to make sure that you established a connection between the Integration Server and the System Landscape Directory during the post-installation steps described in the Installation Guide – SAP Exchange Infrastructure 3.0. The system displays a log that lists the current configuration data, the function test results, errors, and notes for checking. Use this log to verify that the connection works properly.

Furthermore, make sure that you have defined the following RFC destinations for the connection between the Integration Server and the SLD:

- LCRSAPRFC
- SAPSLDAPI

The program ID (LCRSAPRFC_<SID> or SAPSLDAPI_<SID>, respectively) must correspond to an entry under JCo RFC provider for the SAP J2EE Engine.

4.5.1 Checking the SAP J2EE Engine Configuration

Make sure that you maintained the SAP J2EE connection parameters during the post-installation steps described in the Installation Guide – SAP Exchange Infrastructure 3.0.

Also ensure that you created an RFC destination with the same program ID and gateway options that you used on the ABAP side for the LCRSAPRFC and SAPSLDAPI destinations (LCRSAPRFC_<SID> and SAPSLDAPI_<SID>, for example).

Entries are case-sensitive here.

Save the configuration data of the RFC destination in your local file system by choosing Export bundle to file (Save icon). To recover the destination data after a new installation of the SAP J2EE Engine, choose Import bundle from file (Open icon).

4.6 Enabling an Automatic Customizing of the Business Process Engine

If you want to use the Business Process Engine, make sure that you enable an automatic customizing of this engine by executing transaction SWF_XI_CUSTOMIZING. Executing this transaction will set all the required technical settings.
5 Configuring the Change Management Service

Using the Change Management Service with XI is optional. You can use it to transport Integration Builder objects. If you want to use CMS, you have to perform the following configuration steps beforehand.

You can configure CMS for either repository objects or directory objects independently from each other.

The following sections only cover a basic and initial setup of CMS for use with XI. For more information, see

- Installation Guide – SAP NetWeaver JDI on SAP Service Marketplace at service.sap.com/instguides under SAP NetWeaver → Release 04 → Installation → Dev Env
- How to Use CMS in XI3.0 on SAP Service Marketplace at service.sap.com/nw-howtoguides under Exchange Infrastructure
- SAP Library for SAP NetWeaver at SAP Exchange Infrastructure → Design and Configuration → Software Logistics for XI Objects → Transport of XI Objects → Transport using the Change Management Service
- SAP Note 743542

5.1 Deploying the CMS Component

The CMS component is part of the JDI component. You find the sources for the JDI deployment on SAP Service Marketplace at service.sap.com under SAP Support Portal → SAP Software Distribution Centre → Download → Support Packages and Patches → Entry by Application Group → SAP NetWeaver → SAP NETWEAVER → SAP NETWEAVER 04.

Choose NWDI and select the following SDAs:
- JDI 6.40
- JDI OFFLINE 6.40
- JDI BUILD TOOL 6.40 (optional for CMS use, but belongs to JDI)

You can deploy all these SDAs together in one step.

5.2 Maintaining Integration Builder Service Users in CMS

To maintain the Integration Builder service user in CMS, perform the following steps:

1. Access the User Management area of the SAP J2EE Engine of your CMS system and define the following roles and actions:
   - Role CMSDeveloper with actions CMS.Display and CMS.Export
   - Role CMSAdministrator with action CMS.Administrate

2. Choose the User Administration tab page of the SAP J2EE Engine of your CMS system and define the repository service user XIREPUSER and the directory service user XIDIRUSER as CMS users with role CMSDeveloper.
XIDIRUSER and XIREPUSER are used by XI to send data to CMS. Therefore, their passwords must be the same in CMS and in XI.

Make sure that the passwords you define for these two users in CMS match the passwords in your XI system. To do so, log on with both users to the J2EE login screen to make sure that the passwords are correct and that they do not need to be changed anymore.

3. Define a CMS Administrator user with the role CMSAdministrator.

It is assumed that the CMS Administrator performs the customizing and the later transports. If you need to split administration tasks between different users (for example, approval, transports) you have to define additional roles and users.

All users you created during the CMS setup must change their passwords at initial logon or must have set the flag No password change required.

Be aware that passwords are case sensitive.

For more information, see Administration of SAP NetWeaver Java Development Infrastructure → User Management in SAP NW JDI → User Management in the Change Management Service.

5.3 Configuring CMS Domain, CMS Server, and SLD

Use the Landscape Configurator to configure

- The CMS domain
- The CMS server
- The System Landscape Directory (as External Server)

The configuration of a CMS domain is described under Administration of SAP NetWeaver Java Development Infrastructure → Change Management Service → Landscape Configurator → Configuring a Domain.

To configure the CMS server, perform the following steps:

1. Enter the CMS URL (server name and port of the CMS system).
2. Create a CMS user. The user allows CMS to connect to the XI systems.

   This user (for example LSADMIN) must be maintained as a service user in all XI systems (including the SLD) with the same password. It requires the following ABAP roles:
   - SAP_XI_CMS_SERV_USER
   - SAP_SLD_ORGANIZER

   Restart the J2EE Engine after you have added the roles to the CMS user, or wait until the group assignment has been transferred to the J2EE Engine.
Within CMS, it is assumed that there is only one central SLD for all Integration Builder systems of the CMS domain. If more than one SLD is used by the Integration Builder systems, you should choose the master SLD for the domain definition.

A CMS track can only be defined if all software component versions relevant for the track are available within the SLD of the CMS domain.

If you have installed the SLD on a standalone J2EE Engine (and not on an SAP Web AS including XI), you have to perform the following steps:

1. Use the J2EE Visual Administrator and open the component sap-com/com.sap.lcr*sld.
2. Add the CMS user (for example LSADMIN) to the security role LcrInstance WriterAll or to the group SAP_SLD_ORGANIZER.

5.4 Activating CMS in the Integration Builder

Set or enhance the following exchange profile parameters to activate CMS for use with the Integration Builder. The CMS transport option will then be displayed in the Integration Repository and in the Integration Directory.

Normally, these parameters must be set only in systems you plan to insert into the tracks. In all other systems, the transport should be organized by CMS.

5.4.1 Integration Repository

To be able to use CMS for repository objects, set the following parameters in your development and consolidation systems; do not change this parameters in production systems:

- com.sap.aii.ibrep.core.cms.enableClTransport=true
  Determines that the option of transporting change lists with CMS is provided on the user interface.

- com.sap.aii.ibrep.core.cms.enableTransportWizard=true
  Determines that the CMS transport is provided by the export wizard.

In addition, add the following value to com.sap.aii.ib.client.properties:

- com.sap.aii.ibrep.core.cms.*
  Do not replace existing values here. Just add the new one.
5.4.2 Integration Directory

To be able to use CMS for directory objects, set the following parameter in your development system:

- com.sap.aii.ibdir.core.cms.enableTransportWizard=true
  
  Determines that the CMS transport is provided by the export wizard.

- The transport of change lists with CMS is not supported for the Integration Directory.

In addition, add the following value to `com.sap.aii.ib.client.properties`:

- com.sap.aii.ibdir.core.cms.*

Do not replace existing values here. Just add the new one.

In addition, set the following parameter in your consolidation system:

- com.sap.aii.ibdir.core.cms.enableClTransport=true
  
  Determines that the option of transporting change lists with CMS is provided on the user interface.

Do not change any of these parameters in production systems.

5.5 Defining the Tracks in CMS

For information on defining a development track in CMS, see SAP Library for SAP NetWeaver at Application Platform → Java Technology in SAP WAS → Administration Manual → Administration of SAP NetWeaver Java Development Infrastructure → Administration of the Change Management Service → Creating a Development Track.

Specify the tracks as follows:

- For the Integration Repository, specify `http://<servername>:<port>/rep/`
- For the Integration Directory, specify `http://<servername>:<port>/dir/`
6 Configuring Message Archiving for the Adapter Engine and the PCK

To enable message archiving for the Adapter Engine and for the Partner Connectivity Kit (PCK), the following configuration steps are required for the SAP XML Data Archiving Service (DAS) of the SAP J2EE Engine 6.40.

- Configuration of the XML DAS Administration destination
- Definition of the archive store
- Synchronization and display of the home path of archiving sets

6.1 Configuring the Destination for the XML DAS Administration

Use the Destinations service of the J2EE Visual Administrator to configure an HTTP destination for the XML DAS Administration by specifying the following values.

**Settings for the HTTP Destination**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>DASdefault</td>
</tr>
<tr>
<td>URL</td>
<td>http://&lt;j2ee_host&gt;:&lt;j2ee_port&gt;/DataArchiving Service/DAS</td>
</tr>
<tr>
<td>Authentication</td>
<td>BASIC</td>
</tr>
<tr>
<td>User name</td>
<td>XMLDAS</td>
</tr>
<tr>
<td>Password</td>
<td>XMLDAS</td>
</tr>
</tbody>
</table>

This is only an example. You can configure any destination according to the XML DAS documentation (see SAP Service Marketplace at service.sap.com/data-archiving and SAP Library for SAP NetWeaver at Application Platform (SAP Web Application Server) → Java Technology in SAP Web Application Server → Administration Manual → Server Administration → Administration of the XML Data Archiving Service).

6.2 Defining an Archive Store

To define an archive store, perform the following steps:

1. Launch the XML DAS using the URL and logon data maintained for the HTTP destination above.

This example defines a local file system archive. You can define any archive store according to the XML DAS documentation (see SAP Service Marketplace at service.sap.com/data-archiving and SAP Library for SAP NetWeaver at Application Platform (SAP Web Application Server) → Java Technology in SAP Web Application Server → Administration Manual → Server Administration → Administration of the XML Data Archiving Service).
2. Choose Define Archive Stores and then choose New and specify the following parameters.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive Store</td>
<td>Name of your archive store, for example, Archive Store</td>
</tr>
<tr>
<td>Storage System</td>
<td>Name of your storage system, for example, AS1</td>
</tr>
<tr>
<td>Store Type</td>
<td>Select File System</td>
</tr>
<tr>
<td>WebDAV Root or Win Root</td>
<td>If your SAP J2EE Engine runs on a Windows operating system, enter the archive store root path as, for example, C:\Archive. This folder must have been created on the file system beforehand.</td>
</tr>
<tr>
<td>Unix Root</td>
<td>If your SAP J2EE Engine runs on a Unix or Linux operating system, enter the archive store root path as, for example, /usr/sap/Y6D/home/archive. This folder must have been created on the file system beforehand.</td>
</tr>
<tr>
<td>Proxy Host</td>
<td>Leave this parameter empty.</td>
</tr>
<tr>
<td>Proxy Port</td>
<td>Leave this parameter empty.</td>
</tr>
</tbody>
</table>

3. Choose Insert Archive Store, return to Home and test your new archive store by choosing Test Archive Stores.

### 6.3 Synchronizing and Displaying the Home Path

To synchronize and display the home path of archiving sets, perform the following steps:

1. Choose Synchronize Home Path and enter the following parameters:

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Path</td>
<td>Enter the path /&lt;SYSID&gt;/xi_af_msg/ specifying the SYSID of your SAP J2EE Engine installation (for example, C11).</td>
</tr>
<tr>
<td>Action</td>
<td>Select Insert New Home Collection.</td>
</tr>
<tr>
<td>Context</td>
<td>Optional</td>
</tr>
<tr>
<td>Archive Store</td>
<td>Select the archive store you created above.</td>
</tr>
</tbody>
</table>


3. Choose List Archive Paths to verify that your home path is assigned to your archive store.

For more information about message archiving, see SAP Library for SAP NetWeaver at Process Integration → SAP Exchange Infrastructure → SAP XI: Runtime → Central Monitoring → Component Monitoring → Archiving Messages.
7 Further Optional Configuration Steps

7.1 Configuring and Starting the SLD Bridge

As an alternative to manual configuration of technical systems in the SLD and in order to automatically receive reported data that is sent by these systems, you have to configure and start the SLD bridge as well as the data suppliers that run in these systems. The SLD bridge transforms the system data sent by data suppliers to the SLD server into CIM-compliant format.

For information on how to make settings for the data supplier bridge and the individual data suppliers, see the System Landscape Directory User Manual for Web AS 6.40. The manual is included in the SAP NetWeaver Library documentation and also accessible on the SLD UI pages by choosing Help.

7.2 Configuring the Integration Builder Documentation Editor

When documenting objects in the Integration Repository, you have the option of setting hyperlinks to external documentation. For this purpose, you can enter either absolute paths (internet addresses) or relative paths in the documentation editor (HTML editor). If you use relative paths, you must define a base location for the URL so that the hyperlinks reference the correct target document.

You can define the base location centrally for all software component versions or you can define different base locations for different software component versions.

You specify base locations in the administration tool of SAP Exchange Infrastructure (XI).

You have saved documentation on your corporate Internet page (http://www.<mycompany>.com) and want to link to one of these documents from the documentation editor in the Integration Builder.

See also the examples below.
7.2.1 Defining New Base Locations

1. To call the administration tool, on the Integration Builder start page, choose Administration from the menu bar.
2. Log on to the administration tool.
4. Choose Add to Entry to List. The Change/Add Base Location dialog is called.
5. In the Software Component field, select the software component version for which you want to specify the base location.
6. Use the table below to enter the remaining information.

Entries for Specifying the Base Location

<table>
<thead>
<tr>
<th>Input Field</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Docu Storage Type</td>
<td>Specify the type of hyperlink that the settings are to refer to.</td>
</tr>
<tr>
<td></td>
<td>o Knowledge Warehouse (File System) for hyperlinks to Knowledge Warehouse documentation that is saved on a file server or on the SAP Help Portal</td>
</tr>
<tr>
<td></td>
<td>o HTTP Server for all other types of hyperlinks</td>
</tr>
<tr>
<td>Base URL</td>
<td>Specify the base location.</td>
</tr>
<tr>
<td></td>
<td>o For hyperlinks to Knowledge Warehouse documentation that is saved on the SAP Help Portal, enter the part of the URL preceding the language ID (for example, EN). The language of the target document is the same as the logon language (with which you have logged on to the Integration Builder)</td>
</tr>
<tr>
<td></td>
<td>o In the case of all other hyperlinks, enter the first part of the URL (up to the start of the relative path).</td>
</tr>
<tr>
<td></td>
<td>For examples about how the basis address and actual URL are related, see below.</td>
</tr>
<tr>
<td>Additional Parameters</td>
<td>You can specify further standard parameters for the URL in the fields Name and Value.</td>
</tr>
</tbody>
</table>

7. Choose Submit Changes.

7.2.2 Editing or Deleting Base Locations

In the Manage Base Locations for External Documentation dialog
- Choose Edit to edit an existing base location
- Choose Delete to delete an existing base location

You must restart the Integration Builder (Integration Repository) for the settings made for the base location to take effect.
### 7.2.3 Importing/Exporting the Base Location

If you want to reuse the settings made for the base location in another Integration Repository (target system), you can export them from the source system in a directory as an XML file, and then import this XML file into the target system.

To export the settings, proceed as follows (in the administration tool of the source system):

1. Call the administration tool in the system in which you want to make the settings for the base location.
2. Choose *Import/Export DocuBases*.
3. In the *Import/Export External Documentation Locations* dialog, position the cursor on *Export* and in the context menu, choose *Save Target As*.
4. Enter the target directory.
   
The settings are saved as an XML file in the specified target directory.

To import the settings, proceed as follows (in the administration tool of the target system):

1. Open the XML file in which the settings are saved.
2. Copy the contents of the XML file to the input field (*Paste XML File Here...*).
3. Choose *Import*.

### 7.2.4 Examples

#### Example of URLs

<table>
<thead>
<tr>
<th>Settings in the Administration Tool</th>
<th>Syntax in the Documentation Editor</th>
<th>Determined URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Docu Storage Type:</strong> Knowledge Warehouse (File System)</td>
<td><code>&lt;a href=&quot;DOCU:KW:887ADB7A030B424B8EF29B99461E52A8&quot;&gt;</code></td>
<td><a href="http://help.sap.com/saphelp_crm40sr1/helpdata/EN/2A/ADA4D0F0E99843E3AA305AA68E6E63/fram">http://help.sap.com/saphelp_crm40sr1/helpdata/EN/2A/ADA4D0F0E99843E3AA305AA68E6E63/fram</a> eset.htm (Integration Builder logged onto in English)</td>
</tr>
<tr>
<td><strong>Base-URL:</strong> <a href="http://help.sap.com/saphelp_nw04/helpdata">http://help.sap.com/saphelp_nw04/helpdata</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Base-URL:</strong> <a href="http://www">http://www</a>.&lt;mycompany&gt;.com</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8 Connecting Business Systems with an Integration Engine to the Central Integration Server

Before configuring business systems with an Integration Engine, ensure the following:

- For business systems based on SAP Web AS 6.20:
  
  You have installed the XI Add-On in each of these business systems as described in the Installation Guide – SAP Exchange Infrastructure 2.0.

- If you want to generate new ABAP proxies on your business systems based on SAP Web AS 6.20:
  
  Use SAP Web AS 6.20 kernel patch level 1253. See SAP Notes 675441 and 721160.

- The business systems and your central Integration Server are maintained in the System Landscape Directory (SLD).

For the central Integration Server, the following Common Information Model (CIM) objects must have been instantiated in the SLD:

- Use the Technical Landscape under System Landscape to describe your Integration Server technically.

  This is not necessary if your Integration Server is configured as a data supplier for the SLD.

- Use the XI Landscape under System Landscape to define the business system with the role of an Integration Engine and define the corresponding Integration Server.


The Integration Engine is the runtime environment of the Exchange Infrastructure. It controls the messaging process (receiving, processing, and forwarding) to and from the central Integration Server.

To configure a business system as a local Integration Engine, proceed as follows:

1. In your user menu, choose Exchange Infrastructure: Administration → Integration Engine – Administration (transaction SXMB_ADM).

2. To determine the role of the business system, choose Integration Engine Configuration.

   The authorizations for maintaining the configuration data are contained in the role SAP_XI_ADMINISTRATOR. Assign this role to system administrators only.

3. Choose Edit → Change Global Configuration Data.

4. Under Global Configuration Data, select Application System as the Role of Business System.

5. In the Corresponding Integ. Server field, enter an HTTP destination (pointing to the corresponding Integration Server) with the following syntax: dest://<HTTPDESTINATION>.
You have to create this HTTP destination in transaction SM59 as a destination of type H. You can maintain all the settings that transaction SM59 offers, namely HTTP, HTTPS, user, and password.

This method therefore enables more secure communication than the use of explicit URLs. We recommend that you create a user in the Integration Server for each Integration Engine that sends messages to the Integration Server. This means that the Integration Server can recognize from the user which Integration Engine sent the message.

6. Under Specific Configuration Data you can specify additional configuration data (settings for monitoring and error handling, for example). To do this:
   a. Select the relevant category under Specific Configuration Data.
   b. Choose Change Specific Configuration Data.

The system displays the Configuration screen where you can enter the configuration parameters that you require for the corresponding category.

For more information about the configuration, see the SAP NetWeaver Library documentation under Exchange Infrastructure → Runtime → Integration Engine → Configuring the Integration Engine → Displaying/Changing Configuration Data.

To create the HTTP destination, perform the following steps:

1. Call transaction SM59 and create an HTTP destination (destination of type H).

   The system displays the HTTP connection to R/3 System may not be secure message. This does not apply in this case and you can ignore the message.

2. On the Technical Settings tab page, enter the HTTP address that the local Integration Engine uses to send messages to the Integration Server.

   For security reasons, it is recommended that you use HTTPS. For more information, see the SAP XI Security Guide under HTTP and SSL.

The HTTP address consists of the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Host</td>
<td>pwdf0740, for example</td>
</tr>
<tr>
<td>Service No. (HTTP or HTTPS port)</td>
<td>8040, for example</td>
</tr>
<tr>
<td>PathPrefix (path in the HTTP service tree of the SAP Web AS)</td>
<td>/sap/xi/engine/?type=entry</td>
</tr>
</tbody>
</table>

If the system displays the Query String Not Allowed message, you can ignore it.
To find out the number of the HTTP or HTTPS port (service number), see the ICM Monitor of the relevant Integration Server (call transaction SMICM and then Goto → Services).

3. On the Logon/Security tab page, enter the following logon data:
   - User: XIAPPLUSER (for example); this user should represent the sending business system (see also the SAP XI Security Guide under Service Users for Message Exchange).
   - Password: the password you have specified for this user
   - Client
   - Language

   The user that you specify here must have the role SAP_XI_APPL_SERV_USER (service user role for application systems). For security reasons, it is recommended that you create different users in the Integration Server, one for each business system.
8.1 Activating an HTTP Service on the Integration Engine (Receiver Side)

For the Integration Engine to be able to receive XML messages using HTTP, you have to activate an HTTP service.

The XIAPPLUSER user and password that are specified here as an example (or any other user that you may specify instead) must exist on the Integration Server with the role SAP_XI_APPL_SERV_USER (service user role for application systems).
The HTTP services are deactivated by default. After every change you have to activate them again.

1. Call the maintenance function of the HTTP services (transaction SICF).
2. Open `default_host → sap → xi → engine`.
   Double-click `engine` to see the default settings for the service `engine`.
3. Activate the service.
   To do this, select the service and choose `Activate Service` in the context menu.

The logon data (user, password, client, and language) is dynamically transferred in the HTTP header fields during an HTTP call of the service. For more information, see the SAP NetWeaver Library documentation under `Exchange Infrastructure → Design and Configuration Time → Configuration → Defining Collaboration Profiles → Communication Channel`.

Since transaction SICF is a cross-client transaction, you have to create an HTTP service for each client (for each business system) in this case. Proceed as follows:

1. Select `engine`, and choose `Create Host/Service`.
2. Enter a new name (`engine_srm_010`, for example).
3. Select `Reference to an existing service`.
4. Enter a description and choose the `Alias Trgt` tab page.
5. Select `engine` and save your alias.
6. Choose `Local Object`.
7. Activate the alias.
   Select the alias and choose `Activate Link` in the context menu.

### 8.2 Registering Queues Used by the Local Integration Engine

1. To register queues for asynchronous message processing, go to your user menu and choose `Exchange Infrastructure: Administration → Integration Engine – Administration` (transaction SXMB_ADM).
2. Choose `Manage Queues`.
3. Save the default settings.
   Depending on the role of the Integration Engine (here: a local Integration Engine), the relevant queues are proposed.
   If errors occur (because the queue name has already been used, for example), deregister all the proposed queues so that you can reregister them afterwards. To do this, choose `Goto → QRFC Monitor`.

For more information about queues, see the SAP NetWeaver Library documentation under `Exchange Infrastructure → Integration Engine → Processing XML Messages → Queues for Asynchronous Message Processing`.
8.3 Creating a Connection Between a Business System and the System Landscape Directory (SLD)

You only require a connection to the SLD when you use the ABAP proxy generation or runtime.

8.3.1 Maintaining the HTTP Address of the SLD Server

Business systems with an Integration Engine require a connection to the SLD server to obtain their name from the SLD. The business system name is used in the header of the message sent to the Integration Server. The connection to the SLD is established by creating an RFC destination (as described below) and calling a registered server program (which is defined on the SAP J2EE RFC engine of the Integration Server). The server program is called with the HTTP address as a parameter that is maintained with transaction SLDAPICUST.

For security reasons, you can also secure your SLD connection with HTTPS. For more information, see SAP Note 766215.

You have to use transaction SLDAPICUST in the business system to define the SLD access data. This data consists of the host and port of the SLD as well as a user and password (XIAPPLUSER, for example). You can use the entries that you maintained on your Integration Server.

8.3.1.1 Overview

The RFC destinations LCRSAPRFC and SAPSLDAPI are used for the SLD connection. They use the SLD access data maintained with transaction SLDAPICUST. The RFC destination LCRSAPRFC is used to read the exchange profile; SAPSLDAPI is used by the ABAP API.

The RFC destinations are also used to retrieve the information about where to find the Integration Builder required for the proxy generation. This information is located in the XI exchange profile.

You have to create both RFC destinations in your business system. They use the same registered server program, which is defined as part of the RFC engine settings of the SAP J2EE Engine on the Integration Server host. This means that all business systems can use the same server program ID (SAPSLDAPI_UNICODE or SAPSLDAPI_NONUNICODE) for their RFC destinations LCRSAPRFC and SAPSLDAPI.

In the case of Unicode business systems (program ID SAPSLDAPI_UNICODE), the Unicode flag is set in the RFC engine settings. In the case of non-Unicode business systems (program ID SAPSLDAPI_NONUNICODE), the Unicode flag is deactivated.

8.3.1.2 Unicode Business Systems

If your business system is a Unicode system, the RFC destinations LCRSAPRFC and SAPSLDAPI are used and need to be defined for the connection between the business system and the SLD.

8.3.1.2.1 Maintaining the RFC Destination LCRSAPRFC

1. Execute transaction SM59 in the business system.
2. Choose Create.
3. Enter at least the following:
   - RFC destination: LCRSAPRFC
   - Connection type: T
   - Description: <your description>

4. Choose ENTER.

5. Choose the Technical settings tab page and do the following:
   a. In the Program ID field under Registered Server Program, enter LCRSAPRFC_UNICODE.
      
      This program ID must correspond to a program ID specified in the RFC engine settings of the SAP J2EE Engine on the Integration Server host.

      The program ID in the RFC engine settings is case-sensitive.

   b. Under Gateway Options enter the Gateway host and Gateway service of your Integration Server host.

      To find this information on the Integration Server host, call transaction SMGW, choose Goto → Parameters → Display, and search for the entries for the gateway hostname and gateway service attributes.

6. Choose the Special Options tab page and select the Unicode flag under Character Width in Target System.

7. Save your settings.

8.3.1.2.2 Maintaining the SAP J2EE Connection Parameters for LCRSAPRFC

   The basic procedure of configuring the connection has changed since patch level 8 of the SAP J2EE engine. If you work with patch level 8 or above, see the documentation Configuring the Connection of the SAP Web AS to the SAP J2EE Engine, which is available on SAP Service Marketplace at service.sap.com/security under Security in Detail → Hot Topic: J2EE.

1. Start your J2EE Engine Administrator either locally by choosing Start → Programs or on your Integration Server by executing the file go from within the directory: ...

2. Choose Cluster → Server → Services → JCo RFC provider

3. Under RFC destination specify the following:
   - Program ID: LCRSAPRFC_UNICODE

   You can reuse this program ID for all Unicode business systems.

   - Gateway Host: <Integration Server host>

   - Gateway Service: <Integration Server gateway service>

   - Number of process: 3

4. Under Repository specify the following:
   - Application Server: <Integration Server host>
5. Choose Set.

8.3.1.2.3 Maintaining the RFC Destination SAPSLDAPI

1. Execute transaction SM59 in the business system.
2. Choose Create.
3. Enter at least the following:
   o RFC destination: SAPSLDAPI
   o Connection type: T
   o Description: <your description>
4. Choose ENTER.
5. Choose the Technical settings tab page and do the following:
   a. In the Program ID field under Registered Server Program, enter SAPSLDAPI_UNICODE.
   b. Under Gateway Options enter the Gateway host and Gateway service of your Integration Server host.
      To find this information on the Integration Server host, call transaction SMGW, choose Goto → Parameters → Display, and search for the entries for the gateway hostname and gateway service attributes.
6. Choose the Special Options tab page and select the Unicode flag under Character Width in Target System.
7. Save your settings.

8.3.1.2.4 Maintaining the SAP J2EE Connection Parameters for SAPSLDAPI

The RFC destination SAPSLDAPI also uses the registered server program SAPSLDAPI_UNICODE. Therefore, you do not need to maintain anything here, since you have already maintained the connection parameters for LCRSAPRFC.

8.3.1.3 Non-Unicode Business Systems

If your business system is a non-Unicode system, the RFC destinations LCRSAPRFC and SAPSLDAPI are used and need to be defined for the connection between the business system and the SLD.

8.3.1.3.1 Maintaining the RFC Destination LCRSAPRFC

1. Execute transaction SM59 in the business system.
2. Choose Create.
3. Enter at least the following:
   o RFC destination: LCRSAPRFC
   o Connection type: T
Choose ENTER.

5. Choose the Technical settings tab page and do the following:

a. In the Program ID field under Registered Server Program, enter LCRSAPRFC_NONUNICODE.

This program ID must correspond to a program ID specified in the RFC engine settings of the SAP J2EE Engine on the Integration Server host.

The program ID in the RFC engine settings is case-sensitive.

b. Under Gateway Options enter the Gateway host and Gateway service of your Integration Server host.

To find this information on the Integration Server host, call transaction SMGW, choose Goto → Parameters → Display, and search for the entries for the gateway hostname and gateway service attributes.

6. Save your settings.

8.3.1.3.2 Maintaining the SAP J2EE Connection Parameters for LCRSAPRFC

The basic procedure of configuring the connection has changed since patch level 8 of the SAP J2EE engine. If you work with patch level 8 or above, see the documentation Configuring the Connection of the SAP Web AS to the SAP J2EE Engine, which is available on SAP Service Marketplace at service.sap.com/security under Security in Detail → Hot Topic: J2EE.

1. Start your J2EE Engine Administrator either locally by choosing Start → Programs or on your Integration Server by executing the file go from within the directory: ...

2. Choose Cluster → Server → Services → JCo RFC provider

3. Under RFC destination specify the following:

   a. Program ID: LCRSAPRFC_NONUNICODE

   You can reuse this program ID for all Unicode business systems.

   b. Gateway Host: <Integration Server host>

   c. Gateway Service: <Integration Server gateway service>

   d. Number of process: 3

4. Under Repository specify the following:

   a. Application Server: <Integration Server host>

   b. System number: <Integration Server system number>

   c. Client: <client where the Integration Server is defined>

   d. User: SAPJSF (defined during the installation of XI)

   e. Password: ask system administrator
5. Choose Set.

8.3.1.3.3 Maintaining the RFC Destination SAPSLDAPI

1. Execute transaction SM59 in the business system.
2. Choose Create.
3. Enter at least the following:
   - RFC destination: SAPSLDAPI
   - Connection type: T
   - Description: <your description>
4. Choose ENTER.
5. Choose the Technical settings tab page and do the following:
   a. In the Program ID field under Registered Server Program, enter SAPSLDAPI_NONUNICODE.
   b. Under Gateway Options enter the Gateway host and Gateway service of your Integration Server host.
      To find this information on the Integration Server host, call transaction SMGW, choose Goto → Parameters → Display, and search for the entries for the gateway hostname and gateway service attributes.
6. Save your settings.

8.3.1.3.4 Maintaining the SAP J2EE Connection Parameters for SAPSLDAPI

The RFC destination SAPSLDAPI also uses the registered server program SAPSLDAPI_NONUNICODE. Therefore, you do not need to maintain anything here, since you have already maintained the connection parameters for LCRSAPRFC.

8.3.2 Testing the Connection Between a Business System and the System Landscape Directory

Call transaction SLDCHECK. The system displays a log that lists the current configuration data, the function test results, errors, and notes for checking.

In the event of communication errors between your local Integration Engine and the System Landscape Directory. For more information, see SAP Note 578904.

8.4 Creating a User for the Runtime Workbench

The Runtime Workbench monitors business systems by using the default user XIRWBUSER with the role SAP_XI_RWB_SERV_USER. Therefore, create a service user with this role in your business system.

8.5 Creating a Destination for the Integration Server for Receiver Pre-Identification (Optional)

Application development teams can use the option of pre-identifying the message receiver available in the Integration Engine. This can improve performance when sending large...
messages. The application development team decides whether to use this function, which enables it to identify the potential receiver of a message before the message is created.

To identify a receiver in advance, you have to create a fixed RFC destination **AI_INTEGRATION_SERVER**, which is used to call the Integration Server. The application program can then use the routing rules of the Integration Directory to decide whether a message will find a receiver, and then trigger the creation of a message.

The Integration Engine requires the fixed RFC destination **AI_INTEGRATION_SERVER** to identify receivers in advance. Proceed as follows:

1. Call transaction SM59, and create the type 3 RFC destination **AI_INTEGRATION_SERVER** (R/3 connection).
2. Enter the Integration Server as the Target Host.
3. Enter **XIAPPLUSER** and password as the user.

The XIAPPLUSER user and password are specified here as an example. This user, or any other user you may specify instead, must exist on the Integration Server with the role **SAP_XI_APPL_SERV_USER** (service user role for application systems).

For more information, see

- The documentation that can be displayed by choosing the Info button in transaction Integration Engine: Administration (SXMB_ADM), menu item Integration Engine Configuration, configuration data category Runtime.
- The SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Runtime → Connectivity → Proxy Runtime → ABAP Proxy Runtime → XI-Specific ABAP Proxy Runtime Protocols → Routing → Receiver Pre-Identification.

### 8.6 Creating a Destination for the Integration Server for Maintaining the Value Mapping Table (Optional)

If your application requires maintenance of the value mapping table on the Integration Server, you have to use the same RFC destination (**AI_INTEGRATION_SERVER**) as for the receiver pre-identification described in the previous section.

For more information, see

- The documentation that can be displayed by choosing the Info button in transaction Integration Engine: Administration (SXMB_ADM), menu item Integration Engine Configuration, configuration data category Runtime.
- The SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Design and Configuration Time → Configuration → Additional Functions → Value Mapping.
9 Integration of Business Systems Using the IDoc Adapter

You can use the IDoc adapter to connect SAP systems (as of release 3.1) and non-SAP systems with the Integration Server using native IDoc structures.

An IDoc-based communication using the Integration Server works as follows:

- The IDoc control record contains information, for example, the logical system name (LS), port definitions, business partner information, partner role information, and more ALE-specific parameters. This IDoc control record information is used to build SOAP message header information for the Integration Server in the IDoc adapter. The configuration in the Integration Directory allows the creation of the XI header from an IDoc control header on the Integration Server inbound side. It is also used on the Integration Server outbound side to create an IDoc control header.

- The original IDoc control record is available in the payload of the XI message, which means that it is not modified by the IDoc adapter.

- The tasks of the IDoc adapter are:
  - To create an IDoc-XML message out of an RFC call on the inbound side. The metadata of the IDoc structure is used for this purpose. This metadata is retrieved from the DDIC of the sender and receiver systems.
  - To create a RFC call out of an IDoc-XML message on the outbound side.
  - To create XI header information by mapping ALE logical system names to business system names stored in the Integration Directory and (optionally) defined in the SLD.
  - To create an IDoc control record out of an XI message on the outbound side.

9.1 Settings for the IDoc Sender System (R/3)

The following describes the handling of IDocs with partner function ‘LS’ (logical system) in the IDoc control header. IDocs with other partner functions like ‘LI’ or ‘KU’ are described in a separate topic.

To be able to send and receive IDocs, you have to set up the classic ALE customizing.

Assuming you have a running ALE scenario, you have to switch the used ALE port to a new port pointing to the Integration Server.

1. Call transaction SM59 to maintain an RFC destination from the IDoc sender system to the Integration Server. The type of this RFC destination is usually an R/3 connection, for example RFC_RCV_089. Specify a service user with the role SAP_XI_APPL_SERV_USER (such as XIAPPLUSER) as the user.

2. Use ALE customizing transaction WE21 to create a new port (for example RCV_089) of type transactional RFC. Use the RFC destination created above.

3. Use ALE customizing transaction WE20 to maintain the partner profile and partner number, for example of type logical system: RCVCLNT089/LS.

Also maintain the message type (and basic type) to be sent as Outbound parameters, and register the port maintained in step 2 as the Receiver port.
9.2 Settings of the Integration Server System

The following describes the handling of IDocs with partner function ‘LS’ (logical system) in the IDoc control header. IDocs with other partner functions like ‘LI’ or ‘KU’ are described in a separate topic.

This section is based on the following example scenario: A business system sends an IDoc to XI by using the IDoc adapter, and XI sends an IDoc to the receiving business system by using the IDoc adapter. This section describes what has to be done for the inbound side (XI receives IDocs) and for the outbound side (XI sends IDocs).

1. Use transaction SXMB_ADM to verify that the role of the Integration Engine is set to Integration Server as part of the global configuration data. This setting ensures that incoming IDocs are redirected to IDoc adapter processing instead of undergoing normal R/3-based IDoc handling.

For more information, see the SAP NetWeaver Library documentation under Exchange Infrastructure → Runtime → Integration Engine → Configuring the Integration Engine → Displaying/Changing Configuration Data.

2. Use transaction SM59 to maintain an RFC destination for the IDoc sender/receiver system. This RFC destination is used to retrieve the IDoc metadata from the sender system. The IDoc adapter needs these metadata to create the corresponding IDoc-XML message from the RFC stream.

Specify a user with the authorization to call function module RFC_READ_TABLE. Depending on the release of the system that contains the metadata, different function groups (FUGRs) have to be accessed to read this metadata, and the corresponding access authorizations have to be granted to these function groups (Authorization Object: S_RFC, ACTVT: 16).

Relevant function groups depending on SAP releases

<table>
<thead>
<tr>
<th>SAP Release</th>
<th>Function Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1H</td>
<td>SDTX, EDI8, RFC1, EDIT</td>
</tr>
<tr>
<td>4.0A to 6.10</td>
<td>SDTX, EDIMEXT</td>
</tr>
<tr>
<td>6.20</td>
<td>IDXN</td>
</tr>
</tbody>
</table>

For more information on the availability of the individual function modules in function group EDIMEXT, see SAP Note 212011.

As of SAP Web AS 6.20, the role SAP_XI_IS_SERV_USER exists, which includes the authorization for function group IDXN.
3. Use transaction IDX1 to assign a port (RFC destination) to the system that contains the metadata of the IDoc types. This system is defined by the sender port, for example SAPSND (SAP<SID>), and the client, in the IDoc control record.

An SAP system with system ID XYZ and client 300 sends an IDoc to the Integration Server. The sender port in the IDoc control header is SAPXYZ. You use transaction IDX1 on the Integration Server to maintain which RFC destination to use for IDocs with port SAPXYZ and client 300 in order to retrieve the IDoc metadata.

There is a mechanism on the Integration Server that uses this RFC destination to retrieve and cache the metadata at runtime if it is not yet available on the Integration Server.

For more information, see the SAP NetWeaver Library documentation under Exchange Infrastructure → Runtime → Connectivity → Adapters → IDoc Adapter → IDoc Processing with the IDoc Adapter → Maintaining Ports.

Since IDoc metadata is cross-client, you should only assign one port for each system. If several ports are assigned, ensure that they are all working.

If you are not able to load the metadata from this system (because of administrative or security-related restrictions, for example), you can also load it from a reference system (for example, a test system) and use transaction IDX2 to assign it to your production system afterwards.

For non-SAP systems, either the ports in transaction IDX1 have to point to a reference R/3 system, or you copy the metadata with transaction IDX2.

4. Use the SLD to maintain the sender and receiver business systems of your system landscape.

You have to define a technical system (and a client in the case of SAP systems) to which your business system belongs. When you define a client, you also have to specify the corresponding logical system name (see the following figure).
The technical system configuration is not required if your business system is configured as a data supplier for the SLD.

5. You have to define the business systems in the Integration Directory as services without party.

In SAP Exchange Infrastructure 3.0, the IDoc adapter only uses the service definition (business system) with the corresponding adapter-specific identifiers in the Integration Directory. The maintenance of your IDoc sender system in the SLD is therefore not sufficient (see figure below).

It is recommended that you assign your business system definition retrieved from the SLD to a service in the Integration Directory as shown in the following figure.
Using this information about the system ID, client, and logical system name for a specific business system, the IDoc adapter is able to specify the corresponding service in the XML header. Routing is then based on service names.
Details:

- Logical system, SAP system ID, and client are needed for an IDoc receiver SAP system.
- SAP system ID and client are needed for an SAP system.
- Logical system is needed for a non-SAP system.

This means that the business systems used in XI for the routing definitions in the IDoc-XML message header are retrieved from the adapter-specific identifiers of the service definitions in the Integration Directory, where for each business system client, the corresponding system ID, client, and logical system name is defined.

Use the import function to retrieve the adapter-specific identifiers for a service (business system) to avoid double maintenance in the SLD and Integration Directory.

When an IDoc type is sent to the Integration Server for the first time, the corresponding metadata is loaded from the source system. If metadata for an IDoc type changes, it must be deleted by executing the program IDX_RESET_METADATA (or by using transaction IDX2). The automatic mechanism described above will then ensure that metadata is reloaded as soon as an IDoc of this type is sent.

For more information, see the SAP NetWeaver Library documentation under Exchange Infrastructure → Runtime → Connectivity → Adapters → IDoc Adapter → IDoc Processing with the IDoc Adapter → Loading, Displaying and Deleting Metadata.

### 9.3 Settings for an SAP Receiver System

To retrieve IDocs, you must perform the classic ALE customizing.

### 9.4 Further Hints and Information

- When you define an IDoc communication channel, make sure that you specify the correct version of the IDoc interface to be used, that is:
  - Version 2 for function module INBOUND_IDOC_PROCESS (for example, for 8 character IDoc types, that is R/3 Releases 3.0 and 3.1)
  - Version 3 for function module IDOC_INBOUNDASYNCHRONOUS (for example, for 30-character IDoc types, that is R/3 Releases 4.0 and higher)

- When you load metadata from a sender or receiver system, ensure that all required IDoc types and IDoc segments have been released (this can be checked with transaction IDoc Type Development (WE30) in the corresponding system).

- You can monitor the status of all XML messages by using transaction Integration Engine – Monitoring (SXMB_MONI).

- Transaction SAP Adapter – Monitoring (IDX5) provides you with an overview of all messages sent to or from the Integration Server as IDocs. Double-clicking a message provides you with detailed information on this message in the Integration Server.

- Transaction Display IDoc (WE02) can be used in the sender and receiver SAP system to evaluate if and how the particular IDoc is sent or received.
With transaction Asynchronous RFC Error Log (SM58) you can search for a missing IDoc in the sender system. Enter an asterisk (*) in the User Name field for this purpose.

9.5 Defining an IDoc Communication Channel

Use the Integration Builder (Configuration) to define a communication channel for the IDoc adapter as shown in the following figure.

Specify the following:

- **RFC Destination**: C19CLNT401
  - Destination to the IDoc receiver system, defined on the Integration Server with transaction SM59. With this destination, the IDoc adapter sends the IDoc by using RFC.

- **Port**: SAPC10
  - Port defined with transaction IDX2 on the Integration Server. The IDoc adapter uses this port to retrieve the IDoc metadata. The port definition again uses an RFC destination to obtain the IDoc structure by using RFC. For example, you can define that system A is used to retrieve the IDoc metadata and that system B is the IDoc receiver system. This has to be done if the IDoc receiver system is a non-SAP system.
  
  The receiver port in the generated IDoc control header is always empty.

- **SAP Release**: 46C
  - Determines which function module is used to retrieve the corresponding IDoc metadata.
An empty field is recommended and is the default setting, because then the newest version is always used. A valid value consists of three digits, for example 45A, and represents a version equal to or lower than the one you specified in the SAP Release field.

If you enter an invalid value, for example 4, the following runtime error is displayed:

Segment versions missing, table EDISDEF is empty

- Queueing

Function module IDOC_INBOUND_IN_QUEUE is called (available only in SAP Web AS release 6.40 or higher) if the quality of service in the message header is EOIO. Otherwise, function module IDOC_INBOUNDASYNCHRONOUS is called.

Use transaction SM58 to monitor native RFC communication on the Integration Server. This type of communication protocol is used in two places here:

- To send an IDoc from an SAP system to the Integration Server
- To send an IDoc from the Integration Server to an SAP system
10 Integration of Business Systems Using the Plain J2SE Adapter Engine

There are different adapters that you can use to connect the Integration Engine to external systems (legacy systems, for example). You can use these adapters to convert XML-based and HTTP-based messages to the specific protocols and formats of systems of this kind and the other way round.

The individual adapters can be installed and configured on the same technical base, either together or separately.

The following adapters are available:
- File/FTP adapter
- JDBC adapter
- JMS adapter
- SOAP adapter

10.1 Use

Depending on the configuration, the adapters can be used to do the following:
- Exchange data with external systems using a file interface (File/FTP adapter)
- Access databases using JDBC (JDBC adapter)
- Communicate with messaging systems using the JMS API (JMS adapter)
You need to differentiate here between adapters that are located before inbound processing in the Integration Engine (sender adapter) and those that are located after outbound processing in the Integration Engine (receiver adapter).

Each adapter has a special module that is started when the adapter is connected. You must install and configure this module correspondingly.

- You can also run multiple modules of this kind in parallel. This enables you to use different adapters or different configurations of the same adapter simultaneously. It may be necessary for you to change the installation parameters to enhance the installation.

### 10.2 Configuring the File/FTP, JDBC, JMS, and SOAP Adapters

For information about configuring the adapters, as well as an explanation of the parameters, see the SAP NetWeaver Library documentation Exchange Infrastructure → Runtime → Connectivity → Adapter Engine → Plain J2SE Adapter Engine → Configuration. After you have installed the adapters, this part of the documentation is also available as a PDF document in the file system of the adapters.

The configuration options of the Plain J2SE Adapter Engine can be accessed from the Exchange Infrastructure start page (or from http://<host>:8200, where <host> is the Integration Server in an all-in-one installation).

### 11 Integration using the Adapter Engine (J2EE)

The J2EE Adapter Engine (referred to in the following as simply the Adapter Engine) is a runtime component, which hosts resource adapters for integrating applications and systems with SAP Exchange Infrastructure.

It can be deployed centrally, as part of the Integration Server (standard), or non-centrally on any SAP J2EE Server.

In both cases, adapters that run within this Adapter Engine can be monitored and configured centrally. The following adapters are provided:

- RFC
- SAP Business Connector
- File/FTP
- JDBC
- JMS
- SOAP
- Marketplace
- Mail
- RNIF
- CIDX

Also provided is the Java Proxy Runtime (JPR) for communication between Java proxies and the Integration Server.

The Adapter Engine contains the Adapter Framework for messaging, queuing, security handling services, and connectivity to the Integration Server. You can use this framework to plug in your own resource adapters or those of partners.
The central services of the Adapter Framework and most of the above mentioned adapters are implemented as J2EE services that can be maintained with the J2EE Visual Administrator. The corresponding service properties are listed in the Appendix of this document, together with some typical cases where changes to some of these properties may be required.

11.1 External Drivers for the JDBC and JMS Adapters

For the JDBC and JMS adapters you need external drivers that have to 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 in the SAP J2EE Server. For this purpose, the Adapter Engine installation provides the archive ai://aii_af_jmsproviderlib.sda, to which you have to add the required driver(s). To do so, proceed as follows:

11.1.1 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 example with SonicMQ, 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.

11.1.1.2 For the JDBC Driver and the JMS Driver

Add the driver to archive ai://aii_af_jmsproviderlib.sda with the help of a zip program. Proceed as follows:

1. Modify the file provider.xml from ai://aii_af_jmsproviderlib.sda. To do this, extract provider.xml using a zip program and edit it as follows:
   a. Add the line <jar-name>driver-jar</jar-name> to each jar in the <jars> section.

If you want to install the SAP DB JDBC driver and the SonicMQ JMS driver, this section looks like this:

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

b. Return the changed provider.xml to the archive.

Make sure that the original directory server\ is retained under Path in the zip archive (check for the last Path entry).
2. Add all jars (without path information) that you have defined in `provider.xml` to `aii_af.jmsproviderlib.sda`.

3. Use the SDM to deploy `aii_af.jmsproviderlib.sda` in standalone (offline) mode. If the empty archive was already deployed during installation, then in step 1 of the deployment choose *Update deployed SDAs/SCAs that have any version*. Otherwise the subsequent deployment will be terminated with the message *Already been deployed*. (For more information about the deployment procedure, see the SDM documentation.)

4. Restart the J2EE Server. The drivers are now known to the JMS adapter or JDBC adapter, and can be used as described.

If you use MQSeries 53x.xxx, you have to include the following Java archives (.jar files) in archive `aii_af.jmsproviderlib.sda`:

- `com.ibm.mq.jar`
- `com.ibm.mqbind.jar`
- `com.ibm.mqjms.jar`
- `connector.jar`

You can find the last archive in the J2EE client directory.
12 Communication and Security

12.1 Single Sign-On Configuration for SAP XI

12.1.1 Steps Required for XI Web Components

To ensure that Single Sign-On works properly between the XI Web components, you must change their authentication template from basic to ticket. To do so, you have to perform the following steps:

1. Use the J2EE Visual Administrator and navigate to Server → Services → Security Provider.

2. Choose tab page Runtime → Policy Configuration and specify the login module ticket from the Authentication template for each of the following XI Web components:

   The XI Web components are:
   - sap.com/com.sap.xi.repository*rep
   - sap.com/com.sap.xi.directory*dir
   - sap.com/com.sap.xi.services*run
   - sap.com/com.sap.xi.rwb*rwb_mdt
   - sap.com/com.sap.xi.mdt*mdt
   - sap.com/com.sap.xi.rwb*rwb
   - sap.com/com.sap.lcr*sld
   - sap.com/com.sap.rprof.remoteProfile*exchangeProfile
   - sap.com/com.sap.aii.af.app*AdapterFramework

   All these changes are effective immediately and will still be effective after subsequent redeployments. See also SAP Note 768456.

3. Access the Exchange Profile and set the following property to true:

   com.sap.aii.ib.core.sso.enabled

4. Refresh the All Properties.

5. Refresh the XI start page.

   From now on, the logon dialog will be displayed only once and then no longer for each available component.


For more information about Single Sign-On for XI Java components, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Overview → Roles and Tool Access.
12.1.2 Additional Steps Required for the Runtime Workbench

Since the Runtime Workbench communicates with the SAP Web AS ABAP stack, the Java logon ticket key pair must be modified, and the corresponding certificate must be exported from the J2EEEngine and imported to the ABAP stack.

1. Use the J2EE Visual Administrator to change the client value of the Java logon ticket as described in the SAP NetWeaver Library documentation under Security → Authentication on the SAP Web Application Server ABAP → User Authentication and Single Sign-On → Using Logon Tickets.

   In the J2EE Visual Administrator, navigate to Server → Services → Configuration Adapter, expand the nodes cluster_data → server → cfg → services, change to edit mode, open the property sheet com.sap.security.core.ume.service, change the value of login.ticket_client to a client number that is not used in the ABAP stack, for example 001, and restart the SAP J2EE Engine.

2. Create a new SAPLogonTicketKeypair certificate with a distinguished name (DN) other than the one used in the ABAP stack.

   Use the J2EE Visual Administrator and navigate to Server → Services → Key Storage. In the frame on the right-hand side, select TicketKeystore under Views on the Runtime tab page, and mark the entry SAPLogonTicketKeypair-cert. Choose Delete and delete the entry. Do the same with the entry SAPLogonTicketKeypair. Then choose Create and create a new SAPLogonTicketKeypair entry with a DN other than the one used in the ABAP stack.

   Make sure that you
   - mark Store Certificate
   - use Key Length 1024
   - select Algorithm DSA
   - specify your <SID> as Common Name
   - fill the values for the other keys as appropriate

3. Export the SAPLogonTicketKeypair certificate of the SAP J2EE Engine.

   Use the J2EE Visual Administrator and navigate to Server → Services → Key Storage. In the right frame select TicketKeystore and mark the entry SAPLogonTicketKeypair-cert. Choose Export and export the certificate in either X.509 or Base64 Encoded format.

4. Import the J2EE certificate into the SAP Web AS ABAP stack.

   Log on to the Integration Server (for example with client 100) and call transaction STRUSTSSO2. In the Certificate frame, choose Import Certificate and select the previously exported J2EE SAPLogonTicketKeypair-cert. Use binary format for the X.509 and Base64 format for the Base64 Encoded formatted export. Choose Add to Certificate List and Add to ACL. While adding the certificate to the access control list (ACL), specify the system ID (which is the certificate’s common name, that is, the value for CN=) and the client (the client specified as login.ticket_client in the UME Provider service, 001 in our example).

5. Switch to fully qualified host names.

   To ensure that Single Sign-On works properly, all services must be called with the fully qualified host name. Proceed as follows:
a. On the SAP Web AS ABAP Engine, change the profile parameter
   \texttt{icm/server\_port\_<n>}, to reflect the fully qualified host name in the HOST section.

b. Change the host name to a fully qualified one for the following parameters in the
   exchange profile:
   \begin{itemize}
   \item \texttt{com.sap.aii.rwb.server.centralmonitoring.r3.ashost} (under \textit{Runtime Workbench})
   \item \texttt{com.sap.aii.connect.repository.name} (under \textit{Connections})
   \item \texttt{com.sap.aii.connect.rwb.name} (under \textit{Connections})
   \end{itemize}

See also SAP Note 757373 for further information.

\section*{12.2 Single Sign-On Configuration for the PCK}

Single sign-on to the Partner Connectivity Kit (PCK) Configuration Web Start application is only possible with SAP NetWeaver 2004 SP Stack 11 or higher. For more information refer to SAP Note 810554.

\begin{enumerate}
\item Change the PCK authentication template.
   For single sign-on, it is necessary that the PCK uses the authentication template \textit{Ticket}.
   \begin{enumerate}
   \item In the Visual Administrator, choose \textit{Server} \rightarrow \textit{Services} \rightarrow \textit{Security Provider}.
   \item In the right-hand frame, select the following component on the \textit{Runtime} \rightarrow \textit{Policy Configuration} tab page:
   \texttt{sap.com/com.sap.xi.pck*\_pck}
   \item On the \textit{Authentication} tab page, change the authentication template from \textit{Basic} to \textit{Ticket} by selecting \textit{Ticket} in the dropdown menu.
   \end{enumerate}
\item Assign the security role.
   On the \textit{Security Roles} tab page, assign the role \textit{singlesignon} to the PCK user.
\item Adapt \textit{aii.properties}.
   \begin{enumerate}
   \item Edit the file \textit{aii.properties}, located at \texttt{usr/sap/<SID>/JC<ISysNr>/j2ee/cluster/server0}.
   \item Add the value \texttt{com.sap.aii.ib.core.*} to \texttt{com.sap.aii.ib.client.properties}.
   \item Add the property \texttt{com.sap.aii.core.sso.enabled = true}.
   \item Restart the browser session.
   \end{enumerate}
\end{enumerate}

\section*{12.3 HTTPS Configuration for Messaging}

When configuring HTTPS for message exchange, you must handle the ABAP and Java parts differently.

For the ABAP proxies, you use transaction SM59 to configure the destinations pointing to the Integration Server to use HTTPS.

For the J2EE-based Adapter Engines, the Java proxies, and the Partner Connectivity Kit (PCK), HTTPS communication is set generically using the following parameter in the exchange profile (see also the SAP Security Guide XI under \textit{Network and Communication Security} \rightarrow \textit{HTTP and SSL}).
com.sap.aii.connect.secure_connections = messaging

To be able to send and receive messages using HTTPS/SSL:

- The SAP Web AS J2EE Engine must first be configured to support HTTPS/SSL.
  
  This process includes downloading and deploying the SAP Java Cryptographic Toolkit (IAIK) from SAP Service Marketplace, as well as creating a private key and public certificate for the server. This whole procedure is described in detail in the SAP Library for SAP NetWeaver under Security → Network and Transport Layer Security → Transport Layer Security on the SAP J2EE Engine.

- The SAP Web AS ABAP Engine must be configured to support HTTPS/SSL.
  
  For more information, see SAP Library for SAP NetWeaver under Security → Network and Transport Layer Security → Using the Secure Sockets Layer Protocol with the SAP Web AS ABAP.

With HTTPS, you can use basic authentication with username and password, as well as authentication using client certificates. If you want to use client certificate authentication, additional private keys and public certificates are required for each user.

### 12.3.1 Inbound (Sending Messages to the Adapter Engine or PCK)

No configuration is required in the adapter-specific sender channel configuration (inbound) of the Integration Directory (except for the RNIF and CIDX adapters; see the SAP NetWeaver Library documentation of these adapters under SAP Exchange Infrastructure → Runtime → Connectivity → Adapters → RNIF Adapter or CIDX Adapter, respectively).

The authentication/authorization is performed by the SAP J2EE Engine and therefore needs to be configured with the J2EE Administration Tool. This configuration is described in the SAP J2EE Engine Administration Manual and is outlined in the following section.

When a message is to be sent to the Adapter Engine or PCK (and ultimately to XI), the SAP J2EE Engine serves as the SSL Server and presents its server certificate to the client as part of the SSL handshake procedure.

- Client-side configuration (required):
  
  The public certificate of the trusted authority (CA) that signed the public certificate of the SSL server needs to be imported to the list of trusted certificates of the SSL client. This allows the SSL client to accept the certificate of the server in the SSL handshake.

- Server-side configuration (optional):
  
  If basic authentication is used, no additional configuration is required on server side.

  If client certificate authentication is requested or required by selection of the corresponding option in the SSL service and configuration of the ClientCertLoginModule in the SecurityProvider service (using the J2EE Administration Tool), additional configuration steps are required.

  If the server certificate check on the client side is successful, the client sends its public certificate to the server as part of the SSL handshake (when requested). The server needs to map the certificate to a user for authentication and will then check the authorization based on the security roles of the user.

  In order to allow the J2EE engine to map the client certificate to a user the following steps are required:
The CA cert of the client certificate needs to be imported to the list of trusted certificates (TrustedCAs keystore view in the keystore service) and the client cert needs to be imported to an arbitrary keystore view.

The client certificate has to be mapped to an existing user with role SAP_XI_APPL_SERV_USER by using the J2EE Administration Tool, service SecurityProvider, tab page UserManagement.

12.3.2 Outbound (Sending Messages to an External System from the Integration Server through the Adapter Engine or PCK)

In this scenario the SAP J2EE Engine acts as the SSL client and therefore the public certificate issuer (CA) cert of the SSL server needs to be imported to the TrustedCAs keystore view of the J2EE keystore service, using the J2EE Administration Tool.

This is sufficient for the SSL handshake to succeed, in the case of basic authentication. Whether basic authentication or client certification is to be used is configured in the receiver channel configuration (outbound) of the specific adapter in the Integration Directory (if supported).

For the Adapter Engine or PCK it is possible to select the client certificate authentication. When selected, the name (alias) of the private key and public certificate needs to be specified as well as the keystore view, in which the key and cert can be found (as imported to the J2EE keystore).

The private key and certificate of the user can either be present as one entry (with one name) in the specified keystore view (if both were imported in one step from one .p12 file), or as two entries (if the key was imported separately from the client certificate as .p8 and .crt (or .cert - Base64 encoded) files).

12.3.3 Strict hostname checking

The optional strict hostname checking feature of the SSL handshake can be selected for the Adapter Framework, using the Boolean Adapter Framework service property messaging.ssl.serverNameCheck. When enabled, the client will verify that the CN (common name) of the public certificate of the server exactly matches the hostname of the server as part of the SSL handshake procedure.

12.4 Security Configuration at Message Level

The configuration of security at message level differs depending on the used protocol and communication partner.

To enable message-level security within the Adapter Engine or the Partner Connectivity Kit (PCK), you need to do the following:

- Use the J2EE Visual Administrator and verify under Libraries → IAIKSecurity whether the cryptographic library IAIK has been deployed on the Integration Server.

  If not, download and deploy the SAP Java Cryptographic Toolkit (IAIK) from SAP Service Marketplace. For more information, see the SAP Library for SAP NetWeaver under Security → Network and Transport Layer Security → Transport Layer Security on the SAP J2EE Engine → Configuring the Use of SSL on the SAP J2EE Engine → Deploying the SAP Java Cryptographic Toolkit.
• Verify that the appropriate JCE Unlimited Strength Jurisdiction Policy file has been downloaded from java.sun.com and applied to your Java Runtime Environment (JRE).

• For the RNIF and CIDX adapters, the adapter framework service user must be assigned the Keystore Administrator role for all views in the key storage that are referenced in sender and receiver agreements for RosettaNet or CIDX trading partners. This has to be done in the involved Adapter Engines.

To enable message-level security within the Integration Server (using the XI protocol), the configuration includes the following:

• Configuration of Web service security on the SAP Web Application Server (ABAP stack)
• Configuration of Web service security on the SAP Web Application Server (Java stack)
• Verification of the Web service security configuration by using the Runtime Workbench

For further information about message-level security, see the corresponding section in the SAP XI Security Guide and also the SAP NetWeaver Library documentation of the concerned adapters under SAP Exchange Infrastructure → Runtime → Connectivity → Adapters.

12.4.1 Configuring Web Service Security in the ABAP Stack

To configure Web service security in the ABAP stack of the SAP Web Application Server (AS), you have to do the following:

• Configure the RFC destination for the Web service security services
• Configure the logical port for the Web service security services

12.4.1.1 Configuring the RFC Destination

Perform the following steps to configure the RFC destination for the Web service security services:

1. Use transaction SM59 to create a new RFC destination of type HTTP Connections to Ext. Server.
2. Specify connection type G and enter an arbitrary description.
3. Specify the following technical settings:
   • Target Host: Host name of the Integration Server (Java stack)
   • Service No.: Port number of the Integration Server (Java stack)
   • Path Prefix: Path prefix of the Integration Server (Java stack)

   Depending on the logon procedure or security settings you are going to specify in Step 4, you must set the path prefix as follows:

   ▪ If the logon procedure is Basic Authentication and the SSL option is inactive, enter the path prefix /wssproc/plain?style=document.
   ▪ If the logon procedure is Basic Authentication or Send SAP logon ticket, and the SSL option is active, enter the path prefix /wssproc/ssl?style=document.
   ▪ If the logon procedure is SSL Client Certificate or Send SAP logon ticket, and the SSL option is active, enter the path prefix /wssproc/cert?style=document.
4. Select the logon/security settings that correspond to the path prefix specified in Step3.

12.4.1.2 Configuring the Logical Port

Perform the following steps to configure the logical port for the Web service security services:

1. Use transaction LPCONFIG to configure the logical port by specifying the following attributes.

**Settings for the Logical Port:**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy class</td>
<td>CO_WSSEWSSPROCESSOR_VI_DOCUMENT</td>
</tr>
<tr>
<td>Logical port</td>
<td>BASIC</td>
</tr>
<tr>
<td>Description</td>
<td>Any description</td>
</tr>
<tr>
<td>Default port</td>
<td>Select the check box</td>
</tr>
<tr>
<td>Runtime</td>
<td>Select Web Service Infrastructure</td>
</tr>
<tr>
<td>Call parameter: HTTP Destination</td>
<td>Enter the RFC destination you configured above</td>
</tr>
</tbody>
</table>

2. Save and activate the logical port.

12.4.2 Configuring Web Service Security in the Java Stack

To configure Web service security in the Java stack of the SAP Web Application Server (AS), you have to do the following:

- Assign the security roles
- Check the cryptographic library

12.4.2.1 Assigning the Security Roles

For both the PCK and the central Adapter Engine, use the J2EE Visual Administrator to assign security roles as follows:

1. Select the service Security Provider on the corresponding server.

2. Under Policy Configuration select the Component sap.com/tc-sec-wssec-app*wssprocess.jar

3. Assign the security role WSSESecurityProcessing to the user you specified in the above maintained RFC destination for logging on to the Integration Server.

4. If the private key and public root certificate of any keystore view are not contained in the TrustedCAs or DEFAULT view, assign the security role KeystoreAdministrator of this keystore view component to the user you specified in the above maintained RFC destination for logging on to the Integration Server.

12.4.2.2 Checking the Cryptographic Library

Verify the following:

- The cryptographic library IAIK has been deployed on the Integration Server.
- The JCE Unlimited Strength Jurisdiction Policy file has been applied to the corresponding Java Runtime Environment (JRE).
12.4.3 Verifying the Web Service Security Configuration

To verify the configuration of your Web service security, perform the following steps:

1. Start the Runtime Workbench from the XI tools start page under Integration Monitoring.
2. Choose to display all components under Component Monitoring.
3. Switch to tree representation and expand the Integration Server.
4. Choose Integration Engine and check whether the self test result for the availability of the Web service security has a green status. If not, check whether your configuration is correct.

💡

The assignment of the security role KeystoreAdministrator is not checked at this point.
13 Java Proxy Runtime (JPR)

The Java Proxy Runtime (JPR) handles the connectivity between Java proxies and the Integration Server. It converts data contained in Java proxy objects to XML messages and the other way round at runtime. For this purpose it requires information from both the exchange profile and the System Landscape Directory (SLD).

For more information about JPR, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Runtime → Connectivity → Proxy Runtime → Java Proxy Runtime.

As of SAP NetWeaver 2004, JPR is installed with and integrated into the J2EE Adapter Engine. The Adapter Engine guarantees the quality of service Exactly Once (In Order) required for asynchronous messages and provides services for queuing, persistence, and monitoring of XI messages (as well as other message formats).

As a result, JPR can no longer be used with J2SE, that is, with standalone Java applications. The JPR is integrated into the SAP J2EE Engine as follows:

- In the outbound case, an application uses proxy beans to send request messages and possibly receive response messages. The proxies in turn call the JPR to connect to the Adapter Engine which sends the messages to the Integration Server.

- In the inbound case, a proxy server application is provided, which consists of an EJB bean handling the processing of incoming XI messages. This bean acts as a listener to the Adapter Engine, which receives the messages from the Integration Server, and in turn calls the application bean registered for the particular inbound interface.

Currently, the SAP J2EE Engine is the only J2EE server supported by JPR.

13.1 Settings in the Exchange Profile

The JPR uses the values of the following parameters from the exchange profile to determine system-wide settings. These values should have been set during installation of the Adapter Engine.

You access the maintenance screen at http://<host>:<port>/exchangeProfile, where host and port are the hostname and connection port of the J2EE engine of your business system. Under Connection, the connection parameters of your Integration Server should appear. From there, all necessary profile data is copied to the Secure store of the J2EE engine.

See also the Checking the Connection Parameters in the Exchange Profile section in the chapter Activities for the Adapter Engine of the Installation Guide – SAP Exchange Infrastructure Release 3.0.

13.2 Entries in the System Landscape Directory

In addition to the deployment of your client proxy classes on the J2EE engine of your business system, you have to create a business system instance in the System Landscape Directory (SLD) when implementing an outbound Java proxy scenario. This information is used by JPR to determine the sender service when creating an outbound message, and by the Adapter Engine to locate the address of the Integration Server when sending the message.
For more information about the SLD, see the *SAP System Landscape Directory User Manual for SAP Web AS 6.40* guide.

A JPR instance used by Java applications is represented in the SLD by a technical system of type *Web AS Java* and must be associated with a business system. The technical system is specified by a system name and the name of the host of the J2EE engine.

During installation of the Adapter Engine a technical system of type *Web AS Java* is created in the SLD. It contains the following values:

- Web AS ABAP: none *(Standalone J2EE)*
- System Name (SID): C11
- System Home: `<your hostname>`

Use the built-in wizard of the SLD to create a business system instance by specifying the following:

- An arbitrary name
- The type of the associated technical system (*Web AS Java*)
- The name of the associated technical system (to be selected from the dropdown list; C11 on `<your hostname>` in the above example)
- The related Integration Server (to be selected from the dropdown list)

This configuration data is accessible by JPR and the Adapter Engine only after the CPA cache on your J2EE Engine has been updated.

### 13.3 Security Roles for the Java Proxy Runtime

Use the *Security Provider* service of the J2EE Engine Visual Administrator to verify that the following security roles are assigned to the appropriate user groups as follows:

#### Security Role Assignments

<table>
<thead>
<tr>
<th>Security Role</th>
<th>User Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>xi_jpr_register_bindings</td>
<td>SAP_XI_ADMINISTRATOR_J2EE</td>
</tr>
<tr>
<td></td>
<td>SAP_XI_DEVELOPER_J2EE</td>
</tr>
<tr>
<td>xi_jpr_list_bindings</td>
<td>SAP_XI_ADMINISTRATOR_J2EE</td>
</tr>
<tr>
<td></td>
<td>SAP_XI_DEVELOPER_J2EE</td>
</tr>
<tr>
<td></td>
<td>SAP_XI_CONFIGURATOR_J2EE</td>
</tr>
<tr>
<td>xi_jpr_admin</td>
<td>SAP_XI_ADMINISTRATOR_J2EE</td>
</tr>
<tr>
<td></td>
<td>SAP_XI_CONFIGURATOR_J2EE</td>
</tr>
</tbody>
</table>

Verify that these user groups are also available on the J2EE side.
13.4 Configuring an Inbound Service for the Java Proxy Runtime

This is an additional configuration step, which is only required if the JPR is to be used by inbound services, that is, by EJB beans implementing inbound interfaces on a J2EE application server.

It is described in the SAP NetWeaver Library under SAP Exchange Infrastructure → Runtime → Connectivity → Proxy Runtime → Java Proxy Runtime → Java Proxies as Receiver → Configuring the Channel for Java Proxy Receivers.
14 User Management and User Roles


In the following, a distinction is made between dialog users and service users that already have user roles for their maintenance.

If you perform a client copy for a new business system, ensure that you also copy the user roles.

In addition, you can create more specific user roles by defining data-dependent authorizations in the Integration Builder and assigning these authorizations to roles you create in user management. For more information about data-dependent authorizations and user roles, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Design and Configuration Time → Integration Builder → Additional Functions → Design Functions → User Roles.

14.1 Dialog Users of SAP Exchange Infrastructure

The following dialog user roles are provided in SAP Exchange Infrastructure (XI):

- SAP_XI_DISPLAY_USER
- SAP_XI_DEVELOPER
- SAP_XI_CONFIGURATOR
- SAP_XI_CONTENT_ORGANIZER
- SAP_XI_MONITOR
- SAP_XI_ADMINISTRATOR
- SAP_SLD_ADMINISTRATOR
- SAP_SLD_CONFIGURATOR

These roles are equipped with role menus, which are available and displayed as user menus on the initial SAP Easy Access screen.

The roles contain task-specific authorizations and are described in more detail below.

Make sure that the authorization profiles of these roles have been generated (call transaction PFCG on your SAP Web AS host and choose Utilities → Mass generation; enter SAP_XI* and select Generate automatically).

Also make sure that the user master records containing these roles have been compared (call transaction PFCG on your SAP Web AS host and choose Utilities → Mass comparison; enter SAP_XI* and choose Execute).

The users entered on the ABAP side of the SAP Web AS are automatically available on the J2EE side for logging on to the Integration Builder, System Landscape Directory, Runtime Workbench, and Marketplace adapter. This means that user authentication takes place on the ABAP side.
After installing the SAP Web AS, the roles are also automatically available on the J2EE side. Here they appear in the form of User Groups on the J2EE administration user interface. The authorizations required for the J2EE components (J2EE security roles) are assigned during deployment.

### 14.1.1 Description of the Dialog User Roles

Each of the roles has authorizations from multiple components of SAP Exchange Infrastructure. The following components exist:

- Integration Server (ABAP)
- Integration Repository (J2EE)
- Integration Directory (J2EE)
- System Landscape Directory (J2EE)
- Runtime Workbench (J2EE)
- Marketplace adapter (J2EE)
- Exchange profile (J2EE)

The Integration Repository and the Integration Directory are displayed and maintained using the Integration Builder.

All dialog roles contain at least display authorizations in all the components listed above. The individual dialog roles are described in more detail below:

#### 14.1.1.1 SAP_XI_DISPLAY_USER

The XI display user is able to display the entire content of the Integration Repository and Directory, the System Landscape Directory, the Integration Engine/Server, and so on, within SAP Exchange Infrastructure.

#### 14.1.1.2 SAP_XI_DEVELOPER

The XI developer is responsible for all activities within SAP Exchange Infrastructure that affect the design and development of integration processes.

In particular, this means the design of scenarios, interfaces, and mappings in the Integration Repository using the Integration Builder. The tasks of an XI developer also include generating proxies (ABAP and Java) and implementing them in the associated business systems.

#### 14.1.1.3 SAP_XI_CONFIGURATOR

The XI configurator is responsible for all activities within SAP Exchange Infrastructure that affect the configuration of integration processes (integration content).

In particular, this means

- The settings for logical routing, mapping, and technical routing in the Integration Directory
- The maintenance of the System Landscape Directory
- The maintenance of the configuration data for the IDoc adapter (metadata on the ABAP side)
- The configuration settings of the Adapter Engine
14.1.1.4 SAP_XI_CONTENT_ORGANIZER

The XI content organizer is responsible for activities within SAP Exchange Infrastructure that affect the organization and structuring of the contents of the System Landscape Directory. More precisely, this means maintaining and importing software components in the System Landscape Directory. These are tasks that are usually not performed by an XI developer or an XI configurator.

14.1.1.5 SAP_XI_MONITOR

The XI monitor is responsible for all activities within SAP Exchange Infrastructure that affect monitoring. In particular, this means

- Monitoring XML message processing and message throughput
- Troubleshooting and status tracking of XML messages
- Monitoring the processed IDocs and RFCs in the corresponding adapters (on the ABAP side)
- Monitoring the messages sent or received by the Marketplace adapter

By using the authorization object S_XMB_MONI, you can grant authorizations for monitoring depending on partner, service, and interface. For more information, see the documentation for this authorization object.

14.1.1.6 SAP_XI_ADMINISTRATOR

The XI administrator is responsible for all activities within SAP Exchange Infrastructure that affect the technical configuration and administration. In particular, this means the settings for the Integration Engine/Server, the Integration Builder, the System Landscape Directory, the Runtime Workbench, the IDoc adapter, and the Marketplace adapter. The administrator is also responsible for maintaining the profile data of SAP Exchange Infrastructure in the exchange profile.

14.1.1.7 SAP_SLD_CONFIGURATOR

The SLD configurator is responsible for all activities that affect the configuration of the System Landscape Directory (SLD).

14.1.1.8 SAP_SLD_ADMINISTRATOR

The SLD administrator is responsible for all activities that affect the administration of the System Landscape Directory (SLD).
14.1.2 Making Changes to Dialog User Roles

You can make adjustments to the dialog user roles. In particular, this may be necessary if the authorizations on the ABAP side are to be defined specifically.

14.1.2.1 Structure of the Dialog User Roles

The given dialog user roles are composite roles that each consist of two single roles. For example, the role SAP_XI_DEVELOPER consists of the following two single roles:

- SAP_XI_DEVELOPER_ABAP (contains ABAP authorizations)
- SAP_XI_DEVELOPER_J2EE (contains J2EE authorizations)

The suffixes are also used for other roles in the same way.

14.1.2.2 Procedure

To make changes to authorizations on the ABAP side, proceed as follows for each dialog user role:

1. Call transaction PFCG.
2. Copy the single role SAP_XI_<role>_ABAP into the customer namespace (replace the prefix SAP).
3. Create a new composite role in the customer namespace (for example, *_XI_<role>).
4. Assign the new single role (suffix _ABAP) to the new composite role.
5. Assign the new SAP single role (suffix _J2EE) to the new composite role.
6. Make changes to the new single role (suffix _ABAP).
7. Generate an authorization profile for the new single role (suffix _ABAP).

This is summarized and illustrated in the diagram below.

![Diagram showing structure of composite and single roles]

Basically what is important is that the single roles with a J2EE suffix are still assigned to the current composite role. These J2EE single roles are known to the J2EE applications and are required for automatic authorization assignments during deployment.

14.2 Service Users of SAP Exchange Infrastructure

Some central service users are required for dialog-free communication between the central components of SAP Exchange Infrastructure and between application systems and SAP Exchange Infrastructure.

After installing SAP Exchange Infrastructure, the following service users are already stored in the client of the Integration Server and in the exchange profile. They are listed below together with the roles assigned to them.
### Service users available after installation

<table>
<thead>
<tr>
<th>Service User</th>
<th>Description</th>
<th>Assigned Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>XIREPUSER</td>
<td>User for the Integration Repository</td>
<td>SAP_XI_IR_SERV_USER</td>
</tr>
<tr>
<td>XIDIRUSER</td>
<td>User for the Integration Directory</td>
<td>SAP_XI_ID_SERV_USER</td>
</tr>
<tr>
<td>XIAPPLUSER</td>
<td>User for sender applications</td>
<td>SAP_XI_APPL_SERV_USER</td>
</tr>
<tr>
<td>XIRWBUSER</td>
<td>User for the Runtime Workbench</td>
<td>SAP_XI_RWB_SERV_USER_MAIN</td>
</tr>
<tr>
<td>XILDUSER</td>
<td>User for the System Landscape Directory</td>
<td>SAP_BC_AI_LANDSCAPE_DB_RF</td>
</tr>
<tr>
<td>XIISUSER</td>
<td>User for the Integration Server</td>
<td>SAP_XI_IS_SERV_USER_MAIN</td>
</tr>
<tr>
<td>XIAFUSER</td>
<td>User for the Adapter Engine (communication between SLD, Integration Server, and Adapter Framework)</td>
<td>SAP_XI_AF_SERV_USER_MAIN</td>
</tr>
<tr>
<td>LSADMIN</td>
<td>User for the Change Management Server</td>
<td>SAP_XI_CMS_SERV_USER</td>
</tr>
</tbody>
</table>

Passwords are specified for these users during installation. The names of the users as well as the password may be changed as required. Ensure, however, that they are always assigned the roles listed above.

The roles were created for each of the components of SAP Exchange Infrastructure, so that each component only needs this single service user.

Within SAP Exchange Infrastructure, the roles provide you with all the authorizations required by the respective component for dialog-free access to the other components of SAP Exchange Infrastructure. They are therefore also available on the J2EE side. To access the J2EE components, corresponding authorization assignments (security roles) on the J2EE server are required. These are automatically performed when the J2EE components are deployed.

### 14.2.1 Description of the Service User Roles

The following service user roles are available:

#### 14.2.1.1 SAP_XI_IR_SERV_USER

Service user role for the Integration Repository.

Within SAP Exchange Infrastructure, this role supplies all the authorizations required by the Integration Repository for dialog-free access to the other components of the Exchange Infrastructure.

#### 14.2.1.2 SAP_XI_ID_SERV_USER

Service user role for the Integration Directory.

Within SAP Exchange Infrastructure, this role supplies all the authorizations required by the Integration Directory for dialog-free access to the other components of the Exchange Infrastructure.
14.2.1.3 SAP_XI_APPL_SERV_USER
Service user role for application systems that are sender business systems.
This role supplies all the authorizations required by application systems (ABAP and J2EE) for
dialog-free access to the components of SAP Exchange Infrastructure.

14.2.1.4 SAP_XI_IS_SERV_USER_MAIN
Service user role for the Integration Server.
This role supplies all XI specific authorizations required by the Integration Server for dialog-
free access to business systems (based on SAP Web AS 6.40 plus XI 3.0 Add-On).
Service users that have to be created in business systems for this purpose generally need
additional authorizations that are specific to the service to be accessed.

14.2.1.5 SAP_XI_RWB_SERV_USER_MAIN
Service user role for the Runtime Workbench.
Within SAP Exchange Infrastructure, this role supplies all the authorizations required by the
Runtime Workbench for dialog-free access to the other components of SAP Exchange
Infrastructure.

14.2.1.6 SAP_BC_AI_LANDSCAPE_DB_RFC
Service user role for the System Landscape Directory (SLD).
This role supplies all the authorizations required by the SLD for dialog-free access to the
database of the SAP Web Application Server. This role is usually assigned to exactly one
service user or communication user, which must be defined in the administration of the SLD
for database consistency.

14.2.1.7 SAP_XI_CMS_SERV_USER
Service user role for the Change Management Server (CMS).
Within SAP Exchange Infrastructure, this role supplies all the authorizations required by the
Change Management Server for dialog-free access to the other components of SAP
Exchange Infrastructure, mainly the Integration Repository and Integration Directory.

14.2.2 Service Users in the Exchange Profile
After the installation, the service users already exist together with a password in the
exchange profile (properties of SAP Exchange Infrastructure). They are retrieved from here
by the J2EE components of SAP Exchange Infrastructure when the system is started.
14.3 Users with Data-Dependent Authorizations

Perform the following steps to create special user roles with authorizations restricted to specific data. As a prerequisite, you must have created the corresponding authorizations beforehand (see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Design and Configuration Time → Integration Builder → Additional Functions → Design Functions → User Roles).

14.3.1 Creating Users and Roles (User Groups)

Perform the following steps to create users and roles in the ABAP part of the SAP Web AS.

Since ABAP roles are mapped to J2EE user groups, the term role used in an ABAP context means user group in J2EE.

1. Use transaction SU01 to create a user, for example XI_TA_DEV.
2. Use transaction PFCG to create a role, for example XI_TST_AUTH_DEV.
   For example, for a restricted developer role, copy SAP_XI_DEVELOPER completely to sample role XI_TST_AUTH_DEV, but disable the copying of contained *ABAP and *J2EE roles.
3. Delete all users from the role created in the previous step.
4. Use transaction PFCG to assign the user created in step 1 to the role created in step 2.

14.3.2 Assigning Roles to User Groups

Perform the following steps to assign ABAP roles to J2EE user groups in the Java part of the SAP Web AS.

1. Open the user management administration console from the SAP J2EE Engine start page.
2. Choose the role maintenance function, select the new role, and choose Assign groups.
3. To obtain a list of groups, choose

   4. Scroll to the user group (corresponding to the ABAP role) of interest and select it.
   5. Confirm your selection.
   6. Check and confirm the assignment on the following screen.

14.3.3 Assigning Unrestricted Roles to Predefined User Groups

You have to assign an unrestricted tool-specific role, for example XiRep_Unrestricted, to predefined user groups without data-dependent restrictions. For these users, only standard J2EE security applies. Otherwise, users of these groups do not have any permission once the additional data-dependent authorization checks are activated.
Perform the following steps to assign a predefined unrestricted role to standard user groups.

1. Open the user management administration console from the SAP J2EE Engine start page.

2. Choose the role maintenance function, select the unrestricted role (for example XiRep_Unrestricted), and choose **Assign groups**.

3. To obtain a list of groups, choose

4. Select the relevant XI user groups (ABAP roles), omit SAP_XI_DEMOAPP, and confirm your selection.

### 14.3.4 Activating Data-Dependent Authorization Checks

Perform the following steps to activate the additional, data-dependent authorization checks.

1. To access the exchange profile from the XI start page, choose **Administration → Exchange Profile**.

2. Go to **IntegrationBuilder → IntegrationBuilder.Repository** and select the property `com.sap.aii.util.server.auth.activation`.

![Exchange Profile](image)
3. Set the property `com.sap.aii.util.server.auth.activation` to `true` and save your settings.

4. Choose All Properties to display and refresh the properties of the Integration Builder.

5. Choose All Properties again and verify that the property `com.sap.aii.util.server.auth.activation` in the list of properties has the correct value.
15 Central Monitoring

This section describes the prerequisites and configuration steps required to enable central monitoring of a distributed SAP Exchange Infrastructure (XI) landscape. The central XI monitoring:

- Makes use of basic SAP Web Application Server (AS) monitoring services such as PMI, CCMS, and Alert Framework
- Requires that these basic services are provided by one central monitoring server
- Queries the System Landscape Directory (SLD) for information about distributed XI components and their communication addresses
- Provides central Web browser access to monitoring UIs through the Runtime Workbench

15.1 General Prerequisites

The following general prerequisites apply (see also SAP Note 768456):

- Central monitoring server must be defined in the exchange profile
  The instance parameter icm/host_name_full needs to be specified with a fully-qualified host name (see also SAP Note 654982).
  The following properties need to be set in the exchange profile. See also chapter 16. They define the SAP Web AS that provides the CCMS monitor, PMI server, and Alert server for XI.
    - com.sap.aii.rwb.server.centralmonitoring.r3.ashost
      Specify the same fully-qualified host name as for instance parameter icm/host_name_full.
    - com.sap.aii.rwb.server.centralmonitoring.r3.client
    - com.sap.aii.rwb.server.centralmonitoring.r3.sysnr
    - com.sap.aii.rwb.server.centralmonitoring.httpport
      Specify the port of the ABAP Web server, not the J2EE port.
- Technical user for the Runtime Workbench must be defined on all XI components
  The technical user for the Runtime Workbench (XIRWBUSER) as defined in the exchange profile needs to be available on all XI components that are to be monitored centrally. This service user is defined by the following properties, which are initially set during installation:
    - com.sap.aii.rwb.serviceuser.name
    - com.sap.aii.rwb.serviceuser.pwd
    - com.sap.aii.rwb.serviceuser.language
- ITS plug-in running on the Integration Server and central monitoring server
  To allow navigation from the Runtime Workbench to ABAP monitoring transactions, for example to RZ20, the ITS plug-in needs to run on the Integration Server and on the central monitoring server.
Use transaction SICF to check whether the ITS service is activated. You may also have to set profile parameter `em/global_area_mb` to 128 to allow proper ITS operation.

- **Activation of required SICF services**

  In addition to ITS, additional SICF services have to be activated on the central monitoring server in order to allow proper operation of end-to-end monitoring and alerting. You activate the required services by executing the report RSXMB_ACTIVATE_ICF_SERVICES (see also SAP Note 517484).

  Alternatively, you can manually activate the services listed below by using transaction SICF.

**SICF Services to be Activated:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Path to SICF Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITS plugin</td>
<td>/sap/bc/gui/sap/its/webgui</td>
</tr>
<tr>
<td>End-to-end monitoring</td>
<td>/sap/bc/bsp/sap/spi_monitor</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/spi_admin</td>
</tr>
<tr>
<td></td>
<td>/sap/public/bsp/sap/public/graphics/jnet_handler</td>
</tr>
<tr>
<td></td>
<td>/sap/public/bsp/sap/public/graphics/mimes</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/public/graphics</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/graphics/net</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/graph_bsp_test</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/graph_bsp_test/Mimes</td>
</tr>
<tr>
<td>Alerting</td>
<td>/sap/bc/bsp/sap/alertinbox</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/sxms_alertrules</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/system</td>
</tr>
<tr>
<td></td>
<td>/sap/bc/bsp/sap/public/bc</td>
</tr>
<tr>
<td></td>
<td>/sap/public/bsp/sap/public/bc</td>
</tr>
<tr>
<td></td>
<td>/sap/public/bsp/sap/system</td>
</tr>
<tr>
<td></td>
<td>/sap/public/bsp/sap/htmlb</td>
</tr>
<tr>
<td></td>
<td>/sap/public(bc/bsp/sap/icons</td>
</tr>
<tr>
<td></td>
<td>/sap/public(bc/bsp/sap/icons_rtl</td>
</tr>
<tr>
<td></td>
<td>/sap/public(bc/bsp/sap/ur</td>
</tr>
</tbody>
</table>

- **Consistent content of the SLD**

  The Runtime Workbench and other services, for example the directory cache refresh, read component and addressing information from the SLD. There are two complementary ways to provide the SLD with XI-specific content:
15.2 Configuring End-to-End Message Monitoring

In order to view messages in end-to-end monitoring of the Runtime Workbench, the following basic configurations are necessary:

- The systems to be monitored need to be configured in the Runtime Workbench under Configuration (for a detailed description, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Runtime → Central Monitoring → End-to-End Monitoring → Selecting and Configuring Components).

- Ensure that on each Adapter Engine to be monitored, a destination called pmistore has been defined using the service user for the Runtime Workbench (XIRWBUSER) for authentication.

  For more information, see the Installation Guide – SAP Exchange Infrastructure 3.0, chapter Creating HTTP Destination pmistore in the SAP J2EE Environment.

- Ensure that the service user for the Runtime Workbench (XIRWBUSER) has been defined in each component that is to be included in end-to-end monitoring.

- Ensure that enough batch processes are available on the central monitoring server.

15.3 Configuring Message-Based Alerting

You configure message-based alerts in the Runtime Workbench under Alert Configuration. For a detailed description of the configuration process, see the SAP NetWeaver Library documentation under SAP Exchange Infrastructure → Runtime → Central Monitoring → Alert Configuration.

It is a prerequisite that the SICF services sxms_alertrules and alertinbox are activated on the central monitoring server as described above.

Message-based alerting is only possible for XI components that are configured for end-to-end monitoring with at least monitoring level Medium.
15.4 Configuring the CCMS Alert Monitor for SAP Exchange Infrastructure

15.4.1 Creating a System Group for SAP Exchange Infrastructure

You have to define the system group `ExchangeInfrastructure_Systems` to obtain monitoring information when you call the CCMS alert monitor `Exchange Infrastructure` from the SAP CCMS Monitor Templates within transaction RZ20.

The definition of the `Exchange Infrastructure` monitor includes a rule for the uppermost node of the monitor tree. This rule determines the systems of system group `ExchangeInfrastructure_Systems` that are to be displayed.

This system group must be created and contained in the system from which all systems of your XI landscape are to be centrally monitored, that is, in the central Monitoring Server.

The system group is maintained with transaction RZ21. Two steps are required:

1. Choose `Create remote monitoring entry` from the `Technical infrastructure` menu and maintain all systems of your XI landscape.

2. Choose `Maintain system groups` from the `Technical infrastructure` menu and define the system group `ExchangeInfrastructure_Systems` consisting of the remote systems created before.

   If the local system in which you configure your monitor is part of this group, you can include it in the group by specifying its system ID. You do not need any remote monitoring for this system.

15.4.2 Enabling Performance Alerts for Specific Sender-Receiver Combinations

If you want to create performance-related alerts for specific sender-receiver combinations and display these alerts in the CCMS alert monitor, you have to use the table maintenance transaction SM30 to maintain appropriate logical sender-receiver assignments in table ALXMBPFALERT. For each of these assignments, you specify threshold values for the triggering of alerts. For more information, see SAP Note 730193.

15.4.3 Configuring the CCMS Heartbeat Monitor for Java Components

<table>
<thead>
<tr>
<th>Objective</th>
<th>You have installed an XI Java component, for example, an Adapter Engine. You want to monitor its heartbeat in the XI monitor of CCMS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
<td>Execute the steps as described in the remainder of this section. The heartbeat will be implemented based on the GRMG framework of CCMS.</td>
</tr>
</tbody>
</table>

For more information about GRMG, see the SAP NetWeaver Library documentation under Solution Lifecycle Management → Solution Monitoring → Monitoring in the CCMS → The Alert Monitor → Monitoring with the Generic Request and Message Generator → Customizing and Operating GRMG Monitoring.
Any Java component to be monitored has to be defined as a GRMG scenario. You do this by writing a corresponding XML specification and importing it with transaction GRMG.

A template for such a CRMG customizing file is provided with SAP Note 634771. Use this template and refer to this note for more information.

When writing your XML:

- Note that it is always the text between the desc tags that appears in the CCMS user interface.
- Ensure that you fill in the other tags as well (just like in the example), since the GRMG framework needs all these entries.

Once you have finished and saved your XML specification, proceed as follows:

1. Upload your XML document by calling transaction GRMG.
   For a better display in the alert monitor, we recommend that you do this in the system that belongs to the Integration Server. However, you can also do it in the central monitoring system. In this case, you have to add this system to the system group ExchangeInfrastructure_Systems.

2. Log on to the central server of the system to which you have uploaded your XML document (the server that hosts the enqueue service).

3. Call transaction GRMG to start the scenarios.

Also use transaction GRMG if you want to stop the CCMS heartbeat monitor configured for a particular Java component while keeping other heartbeat monitors.

### 15.4.4 Configuring CCMS Alerts for the Adapter Engine

If processing of a message on the Adapter Engine fails, the message state is changed to faulty. To notify administrators immediately of such processing failures, they can be propagated as alerts to CCMS. As CCMS is the SAP standard interface to third-party monitoring tools, these failure events may also become visible in non-SAP monitoring tools.

**Configuration Steps**

To make use of Adapter Engine alerts, you must configure the connection to CCMS as follows:

1. Make sure that the J2EE engine on which the Adapter Engine is running is configured to be monitored by the central CCMS system.
   To do so, execute transaction RZ20 on your central monitor server. There, the J2EE engine should be visible in the CCMS monitor under SAP J2EE Monitor Templates → All Systems. If not, follow the steps described in the Supportability Setup Guide for SAP NetWeaver 04 on SAP Service Marketplace at service.sap.com/nw04operations → Cross-NW.

2. Check whether the Adapter Engine is configured to report errors to CCMS.
   To do so, start the Visual Administrator and select service SAP XI AF Core. The value of property CCMSEnabled must be set to true (in lower case letters). If it is not, change the property value to true and restart the service.
   To verify that the setting is effective, select the Visual Administrator Monitoring service. Under Root → Applications, you should see a node for the Adapter Engine.
Alerting

Once you have performed the configuration steps in the previous section, errors that occur during message processing in the Adapter Engine should show up in both the Visual Administrator of the J2EE engine on which the Adapter Engine runs, and transaction RZ20 on the central monitor server:

- Open the Visual Administrator and select the Monitoring service.
  Here you should see that the number of errors of the appropriate error category and error code increases when messages fail on the Adapter Engine.
- Execute transaction RZ20 on the central monitor server and select the Exchange Infrastructure monitor.
  You should find a node for the Adapter Engine here and see the same error information as in the Visual Administrator.

Accuracy

The default refresh rate for the monitor is 30 minutes. If you want to have shorter refresh rates, you can change them in the Visual Administrator:

1. Select the Monitoring service.
2. Select an arbitrary Number of errors entry in the monitoring tree of the Adapter Engine (leaf node).
3. Choose Configuration in the right-hand pane.
4. Choose Edit in the dialog box that appears.
5. Change the polling period to 5 minutes, for instance.
6. Choose Save and wait until saving is finished (be patient, since this may take a while).

16 Exchange Profile Parameters

The exchange profile is an XML document, which is stored in the main database of SAP Exchange Infrastructure. The parameters contained in this document define some basic technical settings. Most of these parameters are initialized automatically during the installation phase, but in some cases the administrator may need to maintain them.

The available profile sections (red squares) are displayed in a tree structure.

You access the maintenance screen at http://<host:port>/exchangeProfile, where host and port are the hostname and connection port of your Integration Server.

The system prompts you to log on with a user name and a password. A user in J2EE security role administer is required. This J2EE role is mapped to SAP role SAP_XI_ADMINISTRATOR_J2EE on SAP Web AS 6.40 as the default setting.

This URL provides a Web-based interface for editing the exchange profile.

After any modifications, you must restart the Integration Server in order for the changes to take effect.

Use the Connection link to connect to the main database of SAP Exchange Infrastructure. Enter the RFC connection parameters for the Integration Server; the standard user is XILDUSER (see chapter 0). After the connection parameters are set, the profile main page is
displayed again, now showing the database content as a tree structure. In addition, the profile main page provides links to upload a new XML file containing profile parameters, or to download the current configuration as an XML document.

The following list shows the parameters that would be relevant in a customer installation, along with a brief explanation.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>com.sap.aii.docu.url</td>
<td>Points to the root URL where the tool documentation files are installed (if no documentation is available use &quot;null&quot;).</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.docu.languages</td>
<td>Comma-separated list of the language codes of the installed external tool documentation.</td>
</tr>
<tr>
<td>Connections</td>
<td>com.sap.aii.connect.repository.name</td>
<td>Connection parameters of the repository server:</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.repository.httpport</td>
<td>1. Host name</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.repository.httpsport</td>
<td>2. HTTP port number</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.repository.rmiport</td>
<td>3. HTTPS port number (optional)</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.directory.name</td>
<td>4. RMI port number (RMI is used for communication with the WebStart Client)</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.directory.httpport</td>
<td>Connection parameters of the directory server</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.directory.httpsport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.directory.rmiport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.landscape.name</td>
<td>Connection parameters of the System Landscape Directory server (listening for WBEM requests). WBEM is the communication standard under CIM. They are both defined by the Distributed Management Task Force (see <a href="http://www.dmtf.org">www.dmtf.org</a>).</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.landscape.httpport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.landscape.httpsport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.cr.name</td>
<td>Connection parameters of the Component Repository (usually equivalent to the System Landscape Directory)</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.cr.httpport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.cr.httpsport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.rwb.name</td>
<td>Connection parameters of the Runtime Workbench</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.rwb.httpport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.rwb.httpsport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.rwb.r3.sysnr</td>
<td>Connection parameters for the ABAP part of the Runtime Workbench</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.rwb.r3.client</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.name</td>
<td>Connection parameters of the Integration Server</td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.httpport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.httpport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.r3.httpport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.r3.httpsport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.r3.sysnr</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.integrationserver.r3.client</td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.sap.aii.connect.secure_connections</td>
<td>List of XI components that use SSL communication instead of plain HTTP</td>
</tr>
</tbody>
</table>
## IntegrationBuilder

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii.ib.client.login.languages</td>
<td>Comma-separated list of installed languages.</td>
</tr>
<tr>
<td>com.sap.aii.ib.client.content.languages</td>
<td>Comma-separated list of languages used for content documentation.</td>
</tr>
<tr>
<td>jco.connection.</td>
<td>JCo connection from the J2EE engine to &quot;its&quot; R3 instance (used by the locking service):</td>
</tr>
<tr>
<td>com.sap.aii.ib.server.connect.webas.r3.ashost</td>
<td>SAP application server (see JCo documentation of property jco.client.ashost). This is automatically filled during installation.</td>
</tr>
<tr>
<td>com.sap.aii.ib.server.connect.webas.r3.sysnr</td>
<td>SAP system number (see JCo documentation of property jco.client.sysnr). This is automatically filled during installation.</td>
</tr>
<tr>
<td>com.sap.aii.ib.server.connect.webas.r3.usefastrfc</td>
<td>This flag determines whether fast RFC should be used for connecting to the R/3 part of the SAP Web AS.</td>
</tr>
<tr>
<td>com.sap.aii.ib.server.lockauth.activation</td>
<td>Activates the Locking and Authorization Service (this should always be &quot;true&quot;).</td>
</tr>
</tbody>
</table>

## IntegrationBuilder.Repository

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii.repository.serviceuser.name</td>
<td>This is the service user used by the Integration Repository.</td>
</tr>
<tr>
<td>com.sap.aii.repository.serviceuser.pwd</td>
<td>(See XIREPUSER on page 11)</td>
</tr>
<tr>
<td>com.sap.aii.repository.serviceuser.language</td>
<td>This is the language used by the Integration Repository.</td>
</tr>
<tr>
<td>com.sap.aii.repository.serviceuser.certificate</td>
<td>(See XIREPUSER on page 11)</td>
</tr>
<tr>
<td>com.sap.aii.repository.mapping.additionallytypes</td>
<td>Besides the standard mapping types like &quot;class&quot; for Java mapping programs or &quot;xsl&quot; for XSLT programs, you can register additional mapping types of SAP partners. These additional mapping types enable you to call mapping programs of SAP partners, by calling a BAdI registered under the same mapping type. In the Integration Builder, you can select the additional mapping types in the Interface Mapping Editor when you create an interface mapping. You can enter a link to a mapping program of an SAP partner in the Interface Mapping, which is passed to the BAdI during the execution. You have to specify a key and a caption for an additional mapping type. The key field is restricted to 20 characters and should only contain the characters a,...,z and A,...,Z, the numbers 0,...,9, and the underscore &quot;.&quot;. The caption should not exceed 25 characters. The caption is separated from the key by &quot;</td>
</tr>
</tbody>
</table>

| com.sap.aii.repository.mapping.additionallytypes | Besides the standard mapping types like "class" for Java mapping programs or "xsl" for XSLT programs, you can register additional mapping types of SAP partners. These additional mapping types enable you to call mapping programs of SAP partners, by calling a BAdI registered under the same mapping type. In the Integration Builder, you can select the additional mapping types in the Interface Mapping Editor when you create an interface mapping. You can enter a link to a mapping program of an SAP partner in the Interface Mapping, which is passed to the BAdI during the execution. You have to specify a key and a caption for an additional mapping type. The key field is restricted to 20 characters and should only contain the characters a,...,z and A,...,Z, the numbers 0,...,9, and the underscore ".". The caption should not exceed 25 characters. The caption is separated from the key by "|"). Different additional mapping types are separated by "|". Example: PA_TYPE|Partner Mapping;pa_trafo|Partner Transformation Note: Use the value "none", if no additional mapping types are used. |
### IntegrationBuilder.Directory

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii.directory.serviceuser.name</td>
<td></td>
<td>This is the service user used by the Integration Directory.</td>
</tr>
<tr>
<td>com.sap.aii.directory.serviceuser.pwd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.directory.serviceuser.language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.directory.serviceuser.certificate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See XIDIRUSER on page 11

### IntegrationServer

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii.integrationserver.serviceuser.name</td>
<td></td>
<td>This is the service user used by the Integration Server.</td>
</tr>
<tr>
<td>com.sap.aii.integrationserver.serviceuser.pwd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.integrationserver.serviceuser.language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.integrationserver.serviceuser.certificate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See XIISUSER on page 11

### IntegrationServer.MappingRuntime

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii.mapping.transformerFactory</td>
<td></td>
<td>Customers may want to use a different XML/XSLT engine such as XALAN/Xerces for processing mappings.</td>
</tr>
<tr>
<td>com.sap.aii.mapping.parserFactory</td>
<td></td>
<td>See above.</td>
</tr>
</tbody>
</table>

### ApplicationSystem

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii. applicationsystem.serviceuser.name</td>
<td></td>
<td>This is the service user used by application systems.</td>
</tr>
<tr>
<td>com.sap.aii. applicationsystem.serviceuser.pwd</td>
<td></td>
<td>This user should exist on the main Integration Server and also be associated with the appropriate RFC destination on application systems using SM59. (See XIAPPLUSER on page 63)</td>
</tr>
<tr>
<td>com.sap.aii. applicationsystem.serviceuser.language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii. applicationsystem.serviceuser.certificate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RuntimeWorkbench</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.serviceuser.name</td>
<td>This is the service user used by the Runtime Workbench. (See XIRWBUSER on page 63)</td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.serviceuser.pwd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.serviceuser.language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.serviceuser.certificate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.server.centralmonitoring.r3.client</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.server.centralmonitoring.r3.sysnr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.sap.aii.rwb.server.centralmonitoring.httpport</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AdapterFramework</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap.aii. adapterframework.serviceuser.name</td>
<td></td>
</tr>
<tr>
<td>com.sap.aii. adapterframework.serviceuser.pwd</td>
<td>This user is used by the Adapter Framework when connecting to other servers. (See XIAFUSER on page 11)</td>
</tr>
<tr>
<td>com.sap.aii. adapterframework.serviceuser.language</td>
<td></td>
</tr>
<tr>
<td>com.sap.aii. adapterframework.serviceuser.certificate</td>
<td></td>
</tr>
</tbody>
</table>

When you compare this list with the complete exchange profile, you can see that most of these parameters are in fact for SAP internal use only, and therefore should not be changed during a customer installation.

16.1.1.1 Syntax of Parameter `com.sap.aii.connect.secure_connections`

The parameter `com.sap.aii.connect.secure_connections` has the following syntax:

```
<secure_connections> ::= <entry> ("," <entry>)*
<entry> ::= <usecase> ("(" <component_list> ")")?
<usecase> ::= "messaging" | "all"
<component_list> ::= <sld_instance_name> ("," <sld_instance_name> )*
<sld_instance_name> ::= <application_id> "." <SID> "." <DBHOST>
<application_id> ::= "repository" | "directory" | "af" | "runtime" ...
```

The parameter `secure_connections` specifies where the internal XI communication should use secure communication (HTTPS) instead of plain HTTP.

Basically, it is a comma-separated list of use cases. Currently there are only two use cases:

- **Messaging**
  
  Internal transport of payload messages between XI components (for example, between the Integration Server and adapters).
• All

Includes all communications.

If only the use case is given, the setting applies to all components in the XI landscape. It is also possible to restrict the selection to specific XI components by adding a list of SLD instance names in parentheses after the use case. In this case, the setting only applies for communications which involve one of the listed instances.

Example 1:
```plaintext
com.sap.aii.connect.secure_connections = messaging
```
All components use secure communication for messaging.

Example 2:
```plaintext
com.sap.aii.connect.secure_connections =
```
No secure communication is used (only standard HTTP); this is the default setting.

Example 3:
```plaintext
com.sap.aii.connect.secure_connections = messaging, all(af.BXI.pwdf0591, af.Y6D.ld0050)
```
Secure communication is used for messaging (on all systems) and for all connections from or to the adapters on system BXI/pwdf0591 and Y6D/ld0050.

The SLD instance name of the Integration Server uses the system number instead of the system ID as <SID>.

In addition to setting the parameter `secure_connections`, you also have to configure the HTTPS ports of the involved components in the `Connections` section of the exchange profile.
Appendix

The appendix of this document lists the J2EE service properties that belong to the Adapter Framework, as well as some service properties for which changes may be required.

Service Properties that may Require Changes

Service SAP XI AF Adapter: XI
You may need to change the following properties of the service SAP XI AF Adapter: XI:

- `xiadapter.inbound.persistDuration.default`
- `xiadapter.outbound.persistDuration.default`

These properties determine how long asynchronous messages are to be kept in the database before they can be deleted or archived. The default value is 30 days.

If you expect a message load that exceeds the maximum database size within 30 days, you must reduce this value before the first messages are sent, because changes to this value will only affect new messages.

Service SAP XI AF Messaging
You may need to change the following properties of the service SAP XI AF Messaging:

- `messaging.syncMessageRemover.checkInterval`
  
  The headers of synchronous messages are kept in main storage for monitoring purposes. Therefore, the main storage available to the SAP J2EE Engine may quickly fill up in scenarios with very large synchronous messages or with large numbers of small synchronous messages. This in turn may lead to a Java OutOfMemoryError.

  This property determines how often the headers of synchronous (error-free) messages are to be removed from main storage.

- `messaging.connections`
  
  The Adapter Engine messaging system delivers messages to enterprise Java beans (EJBs) and tries to read these EJBs from the JNDI when starting the SAP XI AF Messaging service.

  This property determines how often and at what time intervals the messaging service tries to find the EJBs.

  Since EJBs are part of J2EE applications and therefore only started after the J2EE services, it may take some time before they are available.

  By default, the messaging service tries to find the EJBs 10 times, with an interval of one minute each time. Therefore, if it takes longer than 10 minutes to start up the applications in the J2EE Engine, the EJBs cannot be found during startup. This in turn leads to error code HTTP 503 (The requested application, AFW, is currently unavailable) when messages are sent to the Adapter Engine.

  In this case, you must either increase the number of retries or the time interval for all connections of this property.
Service SAP XI AF CPA Cache

You may need to change the following properties of the service SAP XI AF Messaging:

- SLD.selfregistration.*

The Adapter Engine registers itself automatically in the System Landscape Directory (SLD) at startup. By default, the name of the host without the domain name is used for this purpose. If you want to register the host of a (non-)central Adapter Engine with full host and domain names in the SLD, you can use this property to overwrite the name that is used for SLD registration.

J2EE Service Properties

The following sections list the available J2EE services of the Adapter Framework together with their properties. You must not change most of these properties. Some of them, however, might require changes in certain situations. They are indicated accordingly.

SAP XI Adapter: BC

The following properties belong to the service SAP XI Adapter: BC

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| XRfc.receiverChannel.normalizeService.From.toTechnicalSystem | Value type: Boolean  
Default: true  
If the property is set to true, the sender service of the XI message is changed to the adapter-specific identifier for RFC. The result is set for the From value for the RFC-XML document with envelope. If the parameter is set to false, the sender service value is used as the From value.  
The internal default is set to true. |
| XRfc.receiverChannel.normalizeService.To.toTechnicalSystem | Value type: Boolean  
Default: true  
If the property is set to true, the receiver service of the XI message is changed to the adapter-specific identifier for RFC. The result is set for the To value for the RFC-XML document with envelope. If the parameter is set to false the value of the receiver service value is used as the To value.  
The internal default is set to true. |
| XRfc.receiverChannel.normalizeService.failOnError | Value type: Boolean  
Default: false  
If one or both of the properties above are set to true, this property controls the system behavior in the case of a normalization step failure. If this property is set to false and the normalization step fails, the values from the XI messages are used as values for the RFC- |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML with envelope document. If it set to true and the normalization step fails, the processing of this message is aborted with an error. The internal default is set to false.</td>
<td></td>
</tr>
<tr>
<td>syncResponseAdaptEncoding</td>
<td>Value type: Boolean: Default: true Encoding problems may occur during synchronous communication from the SAP Business Connector to the SAP Business Connector Adapter. The SAP Business Connector sends a synchronous request with a specific encoding to the adapter. The adapter converts this request to an XI message and sends it to the Integration Server. The encoding is retained. The response from the Integration Server may arrive at the SAP Business Connector Adapter using a different encoding. If this property is set to false, the synchronous response is sent from the adapter to the SAP Business Connector using the encoding set in the response message from the Integration Server. If this property is set to true, the synchronous response from the Integration Server is converted to the original encoding of the request from the SAP Business Connector. The internal default is set to true.</td>
</tr>
<tr>
<td>syncMessageDeliveryTimeoutMsec</td>
<td>Value type: Integer Default: 300000 (time in msec.) This property controls the timeout while sending synchronous messages from the SAP Business Connector Adapter to the Adapter Framework and the other way round. It is also used to set the valid until field of a synchronous response message which is sent from the adapter to the SAP Business Connector. The internal default is the same as shown above.</td>
</tr>
<tr>
<td>Property</td>
<td>Value type</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>idMappPersistDurationDefault</td>
<td>Integer</td>
</tr>
<tr>
<td>senderChannel.TID.duplicate.Check</td>
<td>Boolean</td>
</tr>
<tr>
<td>senderChannel.TID.duplicate.ErrorSendback</td>
<td>Boolean</td>
</tr>
</tbody>
</table>
SAP XI Adapter: JMS

The following properties belong to the service SAP XI Adapter: JMS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapterName</td>
<td>Adapter name identical to the adapter name defined in the adapter metadata in the Integration Repository</td>
</tr>
<tr>
<td>adapterNamespace</td>
<td>Namespace identical to the namespace defined in the adapter metadata in the Integration Repository</td>
</tr>
<tr>
<td>destroyWaitTime</td>
<td>Internal timeout; waiting period before worker deletion</td>
</tr>
<tr>
<td>destroyWaitMax</td>
<td>Number of waiting periods before worker deletion</td>
</tr>
<tr>
<td>General</td>
<td>Not used</td>
</tr>
<tr>
<td>loggerOn</td>
<td>Set to true to switch off JMS logger</td>
</tr>
<tr>
<td>modulePoolAdminLookupName</td>
<td>Name of the JMS Adapter Service in JNDI; is read by the Adapter Framework</td>
</tr>
<tr>
<td>transferAttributeNames</td>
<td>Names of configuration parameters, internally kept in a cache</td>
</tr>
</tbody>
</table>

None of these properties should be changed at any time.

SAP XI Adapter: Marketplace

The following properties belong to the service SAP XI Adapter: Marketplace

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| messaging.system.eventHandler.MML.adapterTypeNamespace | The adapter type namespace typically has the form of a URL.  
Default: http://sap.com/xi/XI/System  
This value is used to retrieve the Marketplace Adapter metadata from the Adapter Framework CPACache service. |
| deliverHopAckToApp            | Value type: Boolean  
Default: true  
This property enables the end-to-end acknowledgment handling of the Marketplace Adapter. |

None of these properties should be changed at any time.

SAP XI Adapter: RFC

Each RFC server needs a metadata repository, which is provided by an SAP system. This SAP system is accessed by an RFC client. The connections are only used for reading. The properties affecting these connections are described in the following sections.
### Properties Related to RFC Clients (Receiver Channels)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **initialRfcClientConnectCheck**          | **Value type:** Boolean  
**Default:** true  
The connection parameters provided by the adapter configuration are checked if  
A channel is created/updated in the Integration Directory  
The RFC adapter is restarted  
A dependent service like the SAP XI AF CPA Cache is restarted  
The J2EE Engine is restarted  
The parameter check is done by connecting/disconnecting to the remote system. If the check fails, the channel is marked as failed and the channel cannot be used for message exchange. To enable the channel or force a recheck, the channel has to be activated from outside once again. This can be done by updating this channel in the Integration Directory or by restarting the RFC adapter or the J2EE Engine.  
If this property is set to false, the check is disabled. Messages sent through a channel with a configuration error and a disabled check produce a runtime error. |
| **clientMode**                            | **Value type:** Fixed text: reset/none  
**Default:** reset  
Each receiver channel has a pool of RFC clients associated with it. If a message is sent through a channel, an RFC client is retrieved from this channel and then the message is processed by using this client. The client is released to the client pool afterwards and it can be used for another message which is sent through the same channel. After the client has been used, a state is assigned to the client in the receiver system.  
The default value of this property is reset, which will issue a reset of the client context in the receiver system after each usage of a client. This leads to SAP systems with versions lower than 4.0A being disconnected/connected.  
If this value is set to none, nothing will be done after the client has been used. |
**RfcClient.confirmTID**

Value type: Boolean  
Default: true  
If set to true, the current TID is directly confirmed after each tRFC call to the remote target system. If set to false, no confirmation of the TID is sent.

**RfcClient.supportEOIO**

Value type: Boolean  
Default: true  
If set to true, receiver channels of the type RFC send a message with quality of service (QoS) Exactly Once In Order (EOIO) as a tRFC to the remote target system. This system must guarantee the direct execution of this tRFC call in the same way that SAP systems do. If set to false, a runtime error occurs if a message with QoS EOIO is sent to a receiver channel of type RFC.

**ignoreJrfcJdsrError**

Value type: Boolean  
Default: true  
Internal flag to prevent errors in certain releases in the JDSR layer. This property will be removed in a future release. This property must not be changed.

---

**Properties Related to RFC Servers (Sender Channels)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **RfcServer.maxRequestSizeOfLUW**          | Value type: Integer  
Default: 1  
This property affects the behavior of tRFC transactions (also called LUW) of an RFC server. The RFC server limits the number of RFC calls in one transaction to the set value. If the value is set to 0, there is no limitation within the RFC adapter. If a value less than 0 is chosen, the internal default of 1 will be set. Use this property with care.  

![Important reminder](image)

If the property is set to values greater than 1, duplicate detection cannot be guaranteed. It is not recommended to change this property. Refer to SAP Note 774705 before changing this property.  

If this property is changed to a value other than 1, the property RfcServer.TIDconversionMode is internally changed, too. For more information, refer to the description of the property. |
### RfcServer.TIDconversionMode

**Value type:** Fixed text: calculate/create  
**Default:** calculate  

This property affects the behavior of the tRFC of an RFC server. To ensure QoS EO and duplicate detection the tRFC transaction identifier (TID) is used to calculate the XI message ID. It guarantees that one TID is always translated to the same XI message ID.

If this property is set to **create**, a new XI message ID is created for each tRFC call. Guaranteed EO handling or duplicate detection is not carried out.

You are not recommended to change this property if the property RfcServer.maxRequestSizeOfLUW has also been changed.

If the property RfcServer.maxRequestSizeOfLUW is set to a value other than 1, this property is internally set to the value **create**. This is necessary since one tRFC LUW only has one TID and each tRFC call is transformed into one XI message. Each XI message needs a unique message ID and they cannot be calculated from one TID.

### RfcServer.duplicateCheck

**Value type:** Fixed text swallow/sendback  
**Default:** swallow  

When an RFC server sends a tRFC call as an XI message to the Adapter Framework, it uses the module processor. The last module in the module chain must be the RFC adapter module (localejbs/RfcAFBean). Within this module, the XI message is sent to the Adapter Framework messaging which carries out a duplicate detection, among other things.

If this property is set to **swallow**, a detected duplicate will be ignored and the success is indicated to the sender system. In this way the EO contract is fulfilled. If the sender system wants to handle these duplicates on its own, the property can be set to the value **sendback**. The Java exception describing the detected duplicate is sent back to the sender system of the tRFC call.

The internal default is the same as shown above.

### RfcServer.TIDmanagementMode

**Value type:** Fixed text weak/strict  
**Default:** weak  

When a tRFC transaction (LUW) is finished, the sender of this transaction can inform the RFC server that this transaction will never be sent again. This process is called confirmTID. The confirmTID signal must not be sent during transaction execution; it can be sent later.

This property controls the behavior of an RFC server when the confirmTID signal is received. If it is set to **weak**, the internal tRFC buffers are cleared; if it is set to **strict**, the buffers are not cleared.

---

*Note: The information provided is a summary of the document content regarding the configuration and usage of properties related to tRFC transactions in SAP Exchange Infrastructure 3.0.*
to strict, a check is performed:
If the confirmTID signal is received and the transaction (TID) is processed immediately beforehand by exactly the same RFC server, the internal tRFC buffers are cleared (unless an exception is thrown). Since the internal tRFC buffers are only stateful pertaining to one transaction, this function is not used to administer the internal TID management for duplicate detection. The duplicate detection for the RFC adapter takes place in the Adapter Framework messaging and its persistency layer.

The internal default is the same as shown above.

This property was introduced with XI 3.0 SP10. The behavior of RFC servers before SP10 is the same as for the strict mode.

<table>
<thead>
<tr>
<th>RfcServer.bufferMode</th>
<th>Value type: Fixed text: normal/compatible</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default: normal</td>
</tr>
<tr>
<td></td>
<td>When a tRFC transaction (LUW) is committed or rolled back, the internal tRFC buffers can be cleared, because they are no longer used.</td>
</tr>
<tr>
<td></td>
<td>If this property is set to normal, the buffers are cleared as described above.</td>
</tr>
<tr>
<td></td>
<td>If it is set to compatible, the buffers are only cleared when a confirmTID signal is sent to the RFC server or a new transaction (LUW) is processed by the same RFC server. This enhancement saves memory especially in environments where large messages are rarely exchanged.</td>
</tr>
<tr>
<td></td>
<td>The internal default is the same as shown above.</td>
</tr>
</tbody>
</table>

This property was introduced with XI 3.0 SP10. The behavior of RFC servers before SP10 is the same as for the compatible mode.
### poolCheckRate

**Value type:** Integer  
**Default:** 3600 (time in secs.)  

The RFC adapter creates a pool of RFC servers for each sender channel. The value *initial connections* from the sender channel affects the number of RFC servers started initially. The value of *maximum connections* affects the number of RFC servers which can be started dynamically.  

This property defines how often the RFC adapter checks the configured RFC server pools. The pool does not increase its size itself, but it is increased, if a pool check is performed. During this check, another RFC server is started for a pool. The pool check produces load, therefore it should not be started too frequently. A check rate of 0 means that no check is performed.

### poolLoadThreshold

**Value type:** Integer  
**Default:** 0  

A new RFC server is only started if a threshold is reached.  

This property is used to calculate it. If the current size of the pool (available servers) minus the value of this property is less or equal than the number of currently working servers, a new RFC server is started. A value of 0 means that a new server is only started if all available servers (current size of the pool) are working. If it is larger, a new server is started before all available servers are working.

### verifySender

**Value type:** Boolean  
**Default:** true  

When an RFC server receives an RFC call it only has some identification information about the sender system involved (SYS-ID and Client).  

It has to find the XI values for the sender party and the sender service to send a message through XI.  

If set to true, the Integration Directory is queried for the adapter specific identifiers for RFC to find these values. If set to false, the values specified in the sender channel are used.  

This property should only be changed when the sender system cannot provide identification information. If set to false, every client with access to the SAP Gateway, at which the RFC server is registered, can send RFC calls to this server without further checks.  

The internal default is the same as shown above.
onlyPartylessServiceAsSender | Value type: Boolean  
Default: true  
If set to true, an RFC server only permits XI messages to be sent when configured as a sender channel related to a service without a party in the Integration Directory. This is done because RFC has no concept of parties. If set to false, XI configurations with parties are also allowed.  
Changing this property will affect the party-less semantic of RFC  
The internal default is the same as shown above.

syncMessageDeliveryTimeoutMsec | Value type: Integer  
Default: 300000 (time in msec.)  
When an RFC server sends an sRFC call as a synchronous XI message to the Adapter Framework, it uses the module processor. The last module in the chain must be the RFC adapter module (localejbs/RfcAFBean). Within this module, the XI message is sent to the Adapter Framework messaging.  
This property sets the timeout, which is used to send the message and wait for a response. If the given time has elapsed, an exception is thrown by the Adapter Framework and sent back to the sender system.  
The internal default is the same as shown above.

Properties Related to Debugging

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| debugPayloadChunkSize | Value type: Value  
Default: MAX/an integer value  
If the tracing for the RFC adapter is enabled, the payload of a message (the RFC-XML document) is also traced.  
This property controls the number of characters of the payload that are traced. If set to MAX, the whole payload is traced. It also can be set to a positive integer number, which is interpreted as the number of characters to trace from the beginning of the payload.  
The internal default is the same as shown above. |
### SAP Exchange Infrastructure 3.0 Configuration Guide

#### JCO.traceLevel

<table>
<thead>
<tr>
<th>Value type: Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default: 0</td>
</tr>
</tbody>
</table>

The RFC adapter relies on the JCo library of the SAP J2EE Engine.

This property is used to activate the trace from this layer. A description of this property can be found in the JCo documentation.

A JCO.traceLevel of 0 means no trace.

Since JCo is a library within the J2EE Engine, it is shared among all components which run on the engine. If one component sets the tracing parameter, this can be reset by another component later.

The internal defaults are the same as shown above.

This property has been removed with XI 3.0 SP11 in order to have only one location in the SAP J2EE Engine where JCo tracing is controlled. Refer to SAP Note 628962 for information about how to enable tracing for the JCo library.

---

#### JCO.tracePath

| See description above. |

This property has been removed with XI 3.0 SP11 in order to have only one location in the SAP J2EE Engine where JCo tracing is controlled. Refer to SAP Note 628962 for information about how to enable tracing for the JCo library.

---

#### traceExceptionListener

<table>
<thead>
<tr>
<th>Value type: Boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default: false</td>
</tr>
</tbody>
</table>

If the tracing for the RFC adapter is enabled, some information offered by RFC servers can be written to the trace.

- traceExceptionListener: Trace exceptions that occur in an RFC server
- traceServerStateChangeListener: Trace each state change of each RFC server
- traceServerErrorListener: Trace errors that occur in an RFC server

This property should only be used to analyze errors in detail.

The internal default is the same as shown above.
### Properties Related to the Module Processor

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| modulprocessorBean        | Value: localejbs/ModuleProcessorBean  
The JNDI name of the module processor Enterprise JavaBeans. If not set, the internal default is the same as shown above. This value is used to send messages from sender channels to the Adapter Framework. |

| loopbackMode              | Value type: Boolean  
Default: false  
Internal flag for internal tests of the RFC adapter. This property will be removed in a future release. This property must not be changed. |

### SAP XI Adapter: XI

#### Properties Related to Connection

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
Default: XIAFUSER  
Default: 100  
Default: de  
The XI adapter as part of the J2EE Adapter Engine gets the required connection information for sending XI messages to the Integration Server (IS) from the Landscape System Directory (SLD) and the Exchange Profile (EP). If the SLDAccess property of the J2EE Service SAP XI AF CPACache is set to false, the connection information of the Integration Server can be explicitly set with the properties listed above. In the PCK environment (SAP XI CPA Cache... |
Service property `cacheType` is set to `PCK`, the property values are ignored at runtime. The connection information to communicate with the Integration Server has to be provided by the channel definition and the collaboration agreement configuration in the Integration Directory.

These properties must not be changed.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>xiadapter.adapterTypeNamespace</code></td>
<td>The adapter type namespace typically has the form of a URL. Default: <code>http://sap.com/xi/XI/System</code>. It is used to retrieve the XI adapter metadata from the Adapter Framework CPACache Service. This property must not be changed.</td>
</tr>
<tr>
<td><code>xiadapter.inbound.numberRetries.default</code></td>
<td>Default: 3</td>
</tr>
<tr>
<td><code>xiadapter.inbound.retryInterval.default</code></td>
<td>Default: 300000</td>
</tr>
<tr>
<td><code>xiadapter.inbound.persistDuration.default</code></td>
<td>Default: 2592000000</td>
</tr>
<tr>
<td><code>xiadapter.inbound.timeout.default</code></td>
<td>Default: 180000</td>
</tr>
</tbody>
</table>

All time period values are defined in milliseconds. The default retry interval is 5 minutes, the default persist duration is 30 days and the default synchronous message timeout is 3 minutes.

These properties are relevant for inbound message processing, but only in case of a central or non-central Adapter Engine. In case of the PCK, these properties are not taken into account at runtime. Instead, values have to be provided by the channel and collaboration agreement configuration in the Integration Directory.

These properties control the guaranteed delivery of XI messages between the Integration Server and the J2EE Adapter Engine.

The first three properties are only relevant for asynchronous messages.

`numberRetries` defines how often an asynchronous message with QoS EO(IO) should be resent in case of a delivery error.

`retryInterval` defines how long the messaging system should wait before the next retry.

`persistDuration` specifies how long the message should be kept in the message database of the messaging system. The value must correspond to the product of `numberRetries` and `retryInterval`. In the case of duplicate detection, the messages should be persisted for the same period of time.
as the message is persisted on the Integration Server. The default value is 30 days.

Only messages with an expired persist duration can be deleted from the database or archived and subsequently deleted.

It can happen that message throughput in the Adapter Engine is so high that the database volume is not able to store all asynchronous messages for the predefined period of time (30 days). In this case, the persist duration value needs to be decreased to a value that allows the messages to be kept in the database for the configured period of time, before the first messages are sent.

Timeout is only relevant for synchronous messages. It defines how long the XI adapter waits for the synchronous response message before returning a MessageExpired exception.

**Properties Related to Outbound Processing (not Relevant for the PCK)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xiadapter.outbound.numberRetries.default</td>
<td>Default: 3</td>
</tr>
<tr>
<td></td>
<td>This, and the next two properties described below control the guaranteed delivery of XI messages between the Integration Server and the J2EE Adapter Engine.</td>
</tr>
<tr>
<td></td>
<td>It is only relevant for asynchronous messages.</td>
</tr>
<tr>
<td></td>
<td>This property defines how often an asynchronous message with quality of service Exactly Once (In Order) should be resent in the case of a delivery error.</td>
</tr>
<tr>
<td>xiadapter.outbound.retryInterval.default</td>
<td>Default: 300000 (5 minutes)</td>
</tr>
<tr>
<td></td>
<td>The property defines how long the messaging system waits between each retry.</td>
</tr>
</tbody>
</table>
xiadapter.outbound.persistDuration.default

Default: 2592000000 (30 days)
The property specifies how long a message should be persisted in the message database of the messaging system.
The value must be greater than the retry interval multiplied by the maximum number of configured retries.
In the case of duplicate detection, the messages should be persisted for the same period of time as the message is persisted in the Integration Server. The default value is 30 days.
Only messages with expired persist durations can be deleted from the database or archived and subsequently deleted.
It can happen that message throughput in the Adapter Engine is so high that the database volume is not able to store all asynchronous messages for the predefined period of time (30 days). In this case, the persist duration value needs to be decreased to a value that allows the messages to be kept in the database for the configured period of time, before the first messages are sent.

For the PCK, these properties are not taken into account at runtime. Instead, these values must be provided by the channel definition and the collaboration agreement.

**SAP XI AF Core**

The following properties belong to the service SAP XI AF Core

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>af.datasource</td>
<td>Default: jdbc/SAP/BC_XI_AF</td>
</tr>
<tr>
<td></td>
<td>This property must not be changed.</td>
</tr>
<tr>
<td>transactionManager</td>
<td>Default for the SAP J2EE Engine: ts</td>
</tr>
<tr>
<td></td>
<td>JNDI name of the J2EE Engine transaction manager.</td>
</tr>
<tr>
<td></td>
<td>This property must not be changed.</td>
</tr>
<tr>
<td>idmapRemover.checkInterval</td>
<td>Value is in minutes.</td>
</tr>
<tr>
<td></td>
<td>Default: 1440 (i.e. once a day)</td>
</tr>
<tr>
<td></td>
<td>The Adapter Framework core service tries to remove persisted IDMapper entries with expired persist durations from the database at system startup and then repeatedly after the configured time interval.</td>
</tr>
</tbody>
</table>
### SAP XI AF CPA Cache

The following properties belong to the service SAP XI AF CPA Cache

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbPoolName</td>
<td>Default: jdbc/notx/SAP/BC_XI_AF_CPA</td>
</tr>
<tr>
<td></td>
<td>This property must not be changed</td>
</tr>
<tr>
<td>cacheType</td>
<td>Possible values are: PCK or DIRECTORY</td>
</tr>
<tr>
<td></td>
<td>The default is set during installation.</td>
</tr>
<tr>
<td></td>
<td>It specifies, whether the CPA Cache service runs in the PCK (Partner</td>
</tr>
<tr>
<td></td>
<td>Connectivity Kit) mode or as the basis of a central or non-central Adapter</td>
</tr>
<tr>
<td></td>
<td>Framework.</td>
</tr>
</tbody>
</table>

### SAP XI AF CPA Cache Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auditLogEnabled</td>
<td>Boolean</td>
<td>true</td>
<td>This property can be used to disable the audit log functionality of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Adapter Framework core service. This property must not be changed.</td>
</tr>
<tr>
<td>adapterMonitor.monitoringMode</td>
<td></td>
<td>cpa</td>
<td>Possible values: cpa or registered</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This property specifies how the adapter monitor tries to identify the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>available adapters. If the value is set to cpa, it checks which adapter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>type metadata schemas are available in the CPA cache service. The list of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>available adapters can also be generated. This list is based on the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>registered adapters of the Adapter Framework core service adapter monitor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This property must not be changed.</td>
</tr>
<tr>
<td>clusterCommunicationTimeoutMsec</td>
<td>Value in</td>
<td>15000 (15 seconds)</td>
<td>It specifies the timeout value for the cluster communication in the SAP</td>
</tr>
<tr>
<td></td>
<td>milliseconds</td>
<td></td>
<td>J2EE Engine. Even though the default value of 15 seconds should be</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sufficient for the adapter monitoring functionality, this value may be</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>increased in cases where you are experiencing cluster timeout exceptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>of the cluster communication mechanism of the J2EE Engine.</td>
</tr>
<tr>
<td>Jarm.[x].enabled</td>
<td>Boolean</td>
<td>true</td>
<td>These properties specify for which components and adapters within the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Adapter Framework the performance measurement (JARM) should be performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This property must not be changed.</td>
</tr>
</tbody>
</table>
### Properties Related to SLD Access

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLDAccess</td>
<td>Value type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Default: true for SAP XI, false for SAP PCK</td>
</tr>
<tr>
<td></td>
<td>This property is set during installation. The value must not be changed.</td>
</tr>
</tbody>
</table>
SLDReader.pollInterval

Value type: Integer (minutes)
Default: 15
Specifies how often the CPA cache tries to read the connection information that is needed to retrieve the cache update data from the Integration Directory, from the SLD.

SLD.selfregistration.hostName
SLD.selfregistration.httpPort
SLD.selfregistration.httpsPort

String values
No default values provided
At system startup, the CPA cache service registers itself with the SLD. Therefore, the Integration Directory knows where to send the cache invalidation event.
The SLD self-registration information is internally read from the J2EE engine. In certain cases where, for example, the host name resolution is not supported in a landscape, it may however be necessary to redefine the values for SLD self-registration (that is, usage IP addresses instead of host names).
These properties usually do not need to be changed and may only be changed after consulting SAP support.

All SLD-related properties are only relevant for central or non-central Adapter Engine installations. The PCK has no SLD access.

Properties Related to Connection

Name | Description
--- | ---
ssl.serverNameCheck | Value type: Boolean
Default: false
This property can be used to enable an additional server name check to check if the HTTP connections between the Integration Directory and the Adapter Framework are configured using SSL (in the Exchange Profile).
If the value is set to true, the client verifies as part of the SSL handshake that the CN name in the server certificate matches the hostname in the URL that is called.
directoryURL | Value type: String
In the case of a central or non-central Adapter Framework, the HTTP connection information for connecting to the Integration Directory is usually read from the SLD.
If the SLDAccess property is manually set to false, the connection information for the Integration Directory can be explicitly maintained.
These properties are provided for SAP internal testing purposes only and must not be changed in a productive environment.
**SAP XI AF Messaging**

The following properties belong to the service SAP XI AF Messaging

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>messaging.connections</td>
<td>List of connection definitions with the following form: (name=ApplicationName, messageListener=localejbs/MessagListenerBeanName, exceptionListener=localejbs/ExceptionListenerBeanName, pollInterval=60000, pollAttempts=10, Send.maxConsumers=5, Recv.maxConsumers=5, Call.maxConsumers=10, Rqst.maxConsumers=10)</td>
</tr>
<tr>
<td></td>
<td>In the SAP XI installation environment the application name and listener bean names must not be changed at any point.</td>
</tr>
<tr>
<td></td>
<td>The messaging system allows multiple applications to send and receive messages in parallel through a “virtual” connection per application. The default connections delivered with the XI installation are for the Adapter Framework, the Java Proxy Runtime, and several adapters.</td>
</tr>
<tr>
<td></td>
<td>The pollInterval and pollAttempts properties define, how often and with which retry interval the messaging system tries to retrieve the listener beans from JNDI at system startup. This is required because the messaging system is implemented as a J2EE service and the message and exception listener beans are implemented as part of J2EE applications, which are started in a second phase during server startup. In the case of a very slow server startup, it may be necessary to increase the pollInterval or pollAttempts values in order to successfully start the four queues for sending and receiving the synchronous and asynchronous messages.</td>
</tr>
<tr>
<td></td>
<td>The messaging system allows to you define the number of threads to be used to process messages in parallel per queue type and per connection definition by the maxConsumer properties.</td>
</tr>
<tr>
<td>messaging.system.defaultMaxConsumersPer Queue</td>
<td>Value type: Positive integer</td>
</tr>
<tr>
<td></td>
<td>Default: 10</td>
</tr>
<tr>
<td></td>
<td>This value is used when the maxConsumer values are omitted in the messaging.connections property.</td>
</tr>
</tbody>
</table>
Properties Related to HTTPS/SSL

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>messaging.ssl.httpsHandler</td>
<td>Java class name</td>
</tr>
<tr>
<td></td>
<td>Default: iaik.protocol.https.Handler</td>
</tr>
<tr>
<td></td>
<td>This property must not be changed. Only IAIK is supported.</td>
</tr>
<tr>
<td>messaging.ssl.securityProvider</td>
<td>Java class name</td>
</tr>
<tr>
<td></td>
<td>Default: iaik.security.provider.IAIK</td>
</tr>
<tr>
<td></td>
<td>This property must not be changed. Only IAIK is supported.</td>
</tr>
<tr>
<td>messaging.ssl.trustedCACerts.viewName</td>
<td>J2EE keystore name</td>
</tr>
<tr>
<td></td>
<td>Default: TrustedCAs</td>
</tr>
<tr>
<td></td>
<td>This property defines the J2EE keystore that is used in the SSL handshake for the trusted CA certificates.</td>
</tr>
<tr>
<td></td>
<td>This property must not be changed.</td>
</tr>
<tr>
<td>messaging.ssl.serverNameCheck</td>
<td>Value type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Default: false</td>
</tr>
<tr>
<td></td>
<td>This property specifies whether the server host name in outbound HTTPS requests should be checked against the host name in the server certificate.</td>
</tr>
</tbody>
</table>

All these properties are delivered with default values, which allow you to send and receive messages by using HTTPS. In order to activate HTTPS/SSL, the IAIK library has to be deployed on the J2EE engine.

Properties Related to JMS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>messaging.jms.0.provider</td>
<td>The only value allowed for the JMS provider name is SonicMQ.</td>
</tr>
<tr>
<td>messaging.jms.0.username</td>
<td>Specify the user name and password that have to be used to connect to the SonicMQ broker.</td>
</tr>
<tr>
<td>messaging.jms.0.password</td>
<td></td>
</tr>
<tr>
<td>messaging.jms.0.url</td>
<td></td>
</tr>
</tbody>
</table>
The URL format to connect to the JMS broker is:
tcp://<host>:<port>, where the default port for SonicMQ is 2506.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>messaging.jms.0.messageLengthKey</td>
<td>JMS provider-specific name for the JMS header field Message Length. Default: message_length. This property must not be changed, because only SonicMQ is supported.</td>
</tr>
<tr>
<td>messaging.jms.0.dmq</td>
<td>JMS provider specific Dead Message Queue name. Default: SonicMQ.deadMessage. This property must not be changed, because only SonicMQ is supported.</td>
</tr>
<tr>
<td>system.SSL_PROVIDER_CLASS</td>
<td>Default: progress.message.net.ssl.iaik.iaikSSLImpl. These properties are only relevant if SSL is used to connect to the SonicMQ broker.</td>
</tr>
<tr>
<td>system.SSL_PRIVATE_KEY_PASSWORD</td>
<td>The property SSL_PROVIDER_CLASS must not be changed, because only SonicMQ is supported.</td>
</tr>
<tr>
<td>system.SSL_CA_CERTIFICATES_DIR</td>
<td>All other default values are only examples. They need to be changed according to the SonicMQ documentation if SSL is to be used to connect to the JMS broker.</td>
</tr>
<tr>
<td>system.SSL_CERTIFICATE_CHAIN</td>
<td>The SSL_CIPHER_SUITES property must not be changed, since it is preconfigured with the default values that are valid for SonicMQ with SSL.</td>
</tr>
<tr>
<td>system.SSL_CERTIFICATE_CHAIN_FORM</td>
<td></td>
</tr>
<tr>
<td>system.SSL_CIPHER_SUITES</td>
<td></td>
</tr>
</tbody>
</table>

JMS as a transport protocol is exclusively supported for the Marketplace Adapter. The only JMS provider currently supported is SonicMQ.

Properties Related to the Messaging System

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>messaging.initial.delay</td>
<td>Values are time periods in milliseconds Default for both properties: 0 These properties must not be changed for the SAP J2EE Engine. In the case of other J2EE engines, it may be necessary to delay the initialization and startup up the messaging system in order to wait for the startup of required resources, such as the data source of the transaction manager and so on.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>messaging.system.transactionManager</td>
<td>Default value for the SAP J2EE Engine: ts JNDI name of the J2EE Engine transaction manager. This property must not be changed for the SAP J2EE Engine.</td>
</tr>
</tbody>
</table>
| messaging.system.failNonDelivered           | Value type: Boolean  
Default: false  
This property specifies whether asynchronous messages should automatically be set to failed, once the maximum amount of retries is exceeded.  
The default value is false, so that messages in state *Not Delivered* have to be manually set to *failed* as part of an administrator task in message monitoring. |
| messaging.system.lazyInit                   | Value type: Boolean  
Default: false  
This property is obsolete and will be removed in a future release.                                                                                                                                 |
| messaging.syncMessageRemover.checkInterval  | Value: Minutes  
Default: 60  
The messaging system keeps the message header for synchronous messages in the memory for monitoring and duplicate detection purposes. The payload of the message is immediately removed to save resources (this behavior can be configured by the removeBody property), as soon as the message reaches a final state.  
The property specifies, how often the synchronous-message-remover job in the messaging system checks for synchronous messages that have reached a final state and can therefore be completely removed from memory. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Value type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>messaging.syncMessageRemover.messageCount</code></td>
<td>Integer</td>
<td>1000</td>
<td>This property defines the maximum number of synchronous message headers that are kept in memory for monitoring and duplicate detection purposes. The synchronous-message-remover job checks how many messages are in final state and could be completely removed from memory. Then it removes the oldest synchronous message headers until a maximum number of message headers are left in memory, as configured by the <code>messageCount</code> property.</td>
</tr>
<tr>
<td><code>messaging.syncMessageRemover.removeBody</code></td>
<td>Boolean</td>
<td>true</td>
<td>The messaging system removes the payload of synchronous messages by default as soon as they reach a final state. This feature can be disabled to allow the message body to be analyzed in the case of errors (also for synchronous messages) by using this property. This property must only be changed for extended error analysis. Keeping all message bytes of synchronous messages in the memory can lead to OutOfMemoryErrors in the SAP J2EE Engine.</td>
</tr>
<tr>
<td><code>messaging.persistMessageRemover.checkInterval</code></td>
<td>Integer</td>
<td>1440 (1 day)</td>
<td>The messaging system tries to remove persisted (asynchronous) messages with an expired persist duration from the database at system startup and then periodically during the configured time interval. The default value is once per day. The persist duration of XI messages can be configured by properties in the J2EE service <code>SAP XI AF Adapter: XI</code>, where the default value is 30 days. The persist-duration properties for the XI messages may need to be decreased at times of heavy message throughput to avoid that the database volume limitation being reached before the predefined default value of 30 days.</td>
</tr>
</tbody>
</table>
### messaging.cluster.timeout

Value is in milliseconds.
Default: 60000 (1 minute)

This property specifies the timeout value for the cluster communication in the SAP J2EE Engine. Even though the default value of one minute should be sufficient, this value may be increased in cases where you are experiencing cluster timeout exceptions of the cluster communication mechanism of the J2EE Engine.

### messaging.system.node

Value type: Integer
No default.
Defines the cluster node ID.
In the SAP J2EE Engine environment, this property is automatically set at system startup and must not be changed.

### messaging.debug

Currently supported values are: HTTP and SSL
Default: HTTP
This property enables extended debugging statements in the trace file for the message body (extended transport level debugging).

### Properties Related to Monitoring

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>messaging.monitoring.displayDefault</td>
<td>Comma-separated list of names for the message field columns to be displayed</td>
</tr>
<tr>
<td></td>
<td>in the native message monitoring tool of the messaging service. Possible</td>
</tr>
<tr>
<td></td>
<td>values are described below.</td>
</tr>
<tr>
<td></td>
<td>Default: fromParty, toParty, status, timesFailedRetries, sentRecvTime</td>
</tr>
<tr>
<td></td>
<td>Possible values are:</td>
</tr>
<tr>
<td></td>
<td>messageId, refToMsgId, conversationId, fromParty, fromService, toParty,</td>
</tr>
<tr>
<td></td>
<td>toService, action, messageType, connectionName, status, profile, transport,</td>
</tr>
<tr>
<td></td>
<td>deliverySemantics, timesFailedRetries, sentRecvTime, transDelvTime,</td>
</tr>
<tr>
<td></td>
<td>scheduleTime, persisUntil, validUntil, nodeId</td>
</tr>
</tbody>
</table>

This property is related to the native monitoring tool of the messaging system, which is not officially supported. The Message Display Tool (MDT) of the Runtime Workbench can be used to monitor messages. For further information, see SAP Library for SAP NetWeaver at [Process Integration → SAP Exchange Infrastructure → SAP XI: Runtime → Central Monitoring → Message Monitoring](#).
Properties Related to the HTTP Proxy

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system.proxySet</td>
<td>Default for system.proxy.Set: false</td>
</tr>
<tr>
<td>system.proxyHost</td>
<td>The HTTP proxy configuration is disabled by default. There are no default values provided for the other properties.</td>
</tr>
<tr>
<td>system.proxyPort</td>
<td>The proxy-related system properties can be set as documented for the JAVA VM proxy properties.</td>
</tr>
<tr>
<td>messaging.proxy.user</td>
<td></td>
</tr>
<tr>
<td>messaging.proxy.password</td>
<td></td>
</tr>
</tbody>
</table>

SAP XI Industry Speak

The following properties belong to the service SAP XI Industry Speak

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION_MODE</td>
<td>Default: Production</td>
</tr>
<tr>
<td></td>
<td>Refers to the mode of execution of the adapter and can have the value TEST or PRODUCTION. The value set here is used by the RNIF Adapter to fill in the value //ServiceHeader/ProcessControl/GlobalUsageCode and indicates whether the message sent out is a test message or a production message.</td>
</tr>
<tr>
<td>GC_INTERVAL</td>
<td>Indicates the interval in milliseconds at which the internal garbage collector service of the adapter is invoked. The service scans through the executed Business Transaction Dialog instances and deletes all the expired Business Transaction Dialog instances from the database.</td>
</tr>
<tr>
<td>ISP_DATA_SOURCE_JNDI_NAME</td>
<td>The JNDI name for the data source that is used by the Industry Speak adapters for persistence operations.</td>
</tr>
<tr>
<td>ISP_FACTORY_JNDI_NAME</td>
<td>The name of JMS factory that is used by the Industry Speak adapters.</td>
</tr>
<tr>
<td>ISP_QUEUE_JNDI_NAME</td>
<td>The name of JMS queue that is used by the Industry Speak Adapter.</td>
</tr>
<tr>
<td>USER_ACCESS_MODE</td>
<td>Indicates the mode in which the adapter is running in the XI landscape. The value DIRECTORY indicates that the adapter is a part of a central or non-central Adapter Engine.</td>
</tr>
<tr>
<td>KEYSTORE_USER_NAME</td>
<td>Credential used for accessing the keystore views. The views contain the private keys, public keys, and certificates used for security-related activities like signing or encryption. The credential specified here is not used if the USER_ACCESS_MODE is set to DIRECTORY.</td>
</tr>
</tbody>
</table>
None of the properties that belong to the SAP XI Industry Speak service should be changed at any time.

**SAP XI Message Security**

The following properties belong to the service SAP XI Messaging Security

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| JARM_ENABLED             | Default: false  
                           Set this value to true to activate JARM (Java Application Response Time Measurement).                                                        |
| JARM_PREFIX              | Default: XI:Security:  
                           Sets the component prefix in the JARM monitor component overview.                                                                          |
| JARM_USER                | Default: XISecurityUser  
                           Sets the JARM user in the JARM monitor user overview.                                                                                     |
| CLEAN_INTERVAL           | Value type: Integer (hours)  
                           Default: 12  
                           Sets the trigger interval for an asynchronous thread that runs in the background of the message security service. This thread deletes messages from the database that are no longer needed. This improves the runtime performance. |
| QUEUE_CONNECTION_FACTORY  | Default: jmsfactory/default/QueueConnectionFactory  
                           Sets the JNDI lookup name for the Connection Factory.                                                                                       |
| ERROR_QUEUE              | Default: jmsqueues/default/MessageSecurityErrorQueue  
                           Sets the JNDI lookup name for the JMS queue that contains defective archiving sessions that have to be restarted.                           |
| QUEUE                    | Default: jmsqueues/default/MessageSecurityQueue  
                           Sets the JNDI lookup name for the JMS queue that contains the archiving sessions for processing.                                             |