SAP NetWeaver
Process Integration 7.1

Configuring Web Service Scenarios

Document Version 2.0 – April 2008
Contents

1 Introduction ...................................................................................5

2 Configuring the Basic Communication Settings .............................6

2.1 Supported Scenarios ........................................................................... 6
2.1.1 Overview................................................................................................................ 6
2.1.2 Scenarios and Relevant Configuration Tasks.......................................................... 7
  2.1.2.1 Scenarios Using Integration Server Communication.............................7
  2.1.2.2 Scenarios Using Direct Communication .......................................................8

2.2 Configuration Tasks in Detail.............................................................. 9
  2.2.1 Overview of Configuration Tasks and Tools............................................................ 9
  2.2.2 Configuring WS Provider...................................................................................... 11
    2.2.2.1 Configuring WS Provider (AS ABAP 7.0 >= SP 14) ....................................11
    2.2.2.2 Configuring WS Provider (AS ABAP < 7.0 SP 14) ..................................13
  2.2.3 Configuring Message Processing on the Integration Server................................. 14
    2.2.3.1 Overview of Configuration Tasks........................................................14
    2.2.3.2 Defining the System Landscape in the System Landscape Directory..15
    2.2.3.3 Defining Communication Components and Configuring Logical Routing 16
    2.2.3.4 Configuring Outbound Processing (Integration Server)......................16
    2.2.3.5 Configuring Inbound Processing (Integration Server) .........................17

2.2.4 Configuring WS Consumer................................................................................... 18
  2.2.4.1 Finding out Destination of the WS Provider ........................................18
  2.2.4.2 Configuring WS Consumer (AS ABAP 7.0 >= SP14)..................................20
  2.2.4.3 Configuring WS Consumer (AS ABAP < 7.0 SP 14).............................22
  2.2.4.4 Configuring WS Consumer (AS Java 7.1) ..........................................23
  2.2.4.5 Configuring WS Consumer (AS Java 7.0) ..........................................24
  2.2.5 Configuring Communication Paths for Back-Ends Based on AS ABAP 7.1.......... 24
    2.2.5.1 Configuring Communication Path from Consumer (AS ABAP 7.1) to IS 25
    2.2.5.2 Configuring Communication Path From IS to Provider (AS ABAP 7.1)26
    2.2.5.3 Configuring Communication Path From Consumer to Provider (Both AS 27
       ABAP 7.1)

3 Security Configuration ...............................................................28

3.1 Supported Security Scenarios ................................................................. 28
3.2 Configuration Tasks and Tools......................................................... 29
  3.2.1 Configuration Tasks for Back-Ends Based on AS ABAP >= 7.0 SP 14................. 29
  3.2.2 Configuration Tasks for Back-Ends Based on AS ABAP < 7.0 SP 14................. 30
  3.2.3 Configuration Tasks for Back-Ends Based on AS Java 7.1............................... 31
  3.2.4 Configuration Tasks for Back-Ends Based on AS Java < 7.1 ......................... 31
## 4 End-to-End Task Descriptions .........................................................33

### 4.1 Integration Server Communication Using X.509 Client Certificate (Back-Ends Based on AS ABAP < 7.0 SP 14) .................. 33

<table>
<thead>
<tr>
<th>Task</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1 Overview of Configuration Tasks</td>
<td>33</td>
</tr>
<tr>
<td>4.1.2 Configure Security Settings for X.509 Authentication (Using HTTPS)</td>
<td>34</td>
</tr>
<tr>
<td>4.1.2.1 Checking HTTPS and SSL Connection</td>
<td>34</td>
</tr>
<tr>
<td>4.1.2.2 Setting up Secure Communication Between Sender and IS</td>
<td>34</td>
</tr>
<tr>
<td>4.1.2.3 Setting up Secure Communication Between IS and Receiver</td>
<td>46</td>
</tr>
<tr>
<td>4.1.2.4 Defining User Mappings</td>
<td>52</td>
</tr>
<tr>
<td>4.1.3 Configuring WS Provider</td>
<td>55</td>
</tr>
<tr>
<td>4.1.4 Configuring Message Processing</td>
<td>55</td>
</tr>
<tr>
<td>4.1.4.1 Defining System Landscape in the System Landscape Directory</td>
<td>56</td>
</tr>
<tr>
<td>4.1.4.2 Defining Communication Components</td>
<td>56</td>
</tr>
<tr>
<td>4.1.4.3 Configuring Logical Routing</td>
<td>56</td>
</tr>
<tr>
<td>4.1.4.4 Configuring Outbound Processing</td>
<td>57</td>
</tr>
<tr>
<td>4.1.4.5 Configuring Inbound Processing</td>
<td>58</td>
</tr>
<tr>
<td>4.1.4.6 Activating Configuration Objects</td>
<td>59</td>
</tr>
<tr>
<td>4.1.5 Configuring WS Consumer</td>
<td>59</td>
</tr>
</tbody>
</table>
1 Introduction

This guide describes the configuration of communication between the Web service consumer and Web service provider using either of the following communication types:

- Direct (or "point-to-point") communication
- Integration Server communication using a message broker

The scope of this document is to provide an overview of the necessary configuration tasks and tools for selected combinations of back-end stack and release.

This guide is intended for use in conjunction with the accompanying presentation. The slides serve as an entry point to the guide by providing high-level guidance at a glance, while the guide provides step-by-step instruction for the individual configuration tasks.

More information: SAP NetWeaver Process Integration 7.1 - Configuring Web Service Scenarios (Presentation)

Structure of This Guide

The guide is structured in the following way:

- Configuring Basic Communication Settings (2)
  Describes the configuration of the basic communication settings for the supported scenarios independent from any security settings.

- Security Configuration (3)
  Provides an overview of additional configuration tasks to be performed when special security settings have to be applied at runtime. The configuration tasks are listed for different security scenarios and links to detailed documentation are provided for different back-end stacks and releases for each configuration task.

- End-to-End Scenarios (4)
  Provides complete end-to-end configuration procedures for selected scenarios.
2 Configuring the Basic Communication Settings

This chapter describes the configuration of the basic communication between the Web service consumer (WS consumer) and Web service provider (WS provider) - for both direct and Integration Server communication.

Chapter 3 describes additional configuration tasks required for when specific security settings are applied.

2.1 Supported Scenarios

2.1.1 Overview

The configuration procedure is dictated by the use case - or scenario - at hand. In turn, the scenario at hand depends on:

- The basic communication type: direct communication or Integration Server communication
- The technology stack of the backends (AS ABAP, AS Java, or external)
- The release of the backends (for example, AS ABAP 7.1, 7.0, or lower)

The actual scenario results from combining these criteria. However, this guide makes the following assumptions:

- An ABAP back-end is used at the provider side.
- SAP NetWeaver PI 7.1 is installed.

Therefore, this guide considers following scenarios are considered within this guide - respectively for both Integration Server communication and direct communication:

- AS ABAP \(\rightarrow\) AS ABAP where either of the following conditions apply for the back-ends of the WS consumer or provider:
  - Back-end based on AS ABAP 7.0 \(\geq\) SP14
  - Back-end based on AS ABAP \(< 7.0\ SP\ 14

- AS Java 7.1 \(\rightarrow\) AS ABAP where either of the following conditions apply for the ABAP back-end of WS provider:
  - Back-end based on AS ABAP 7.0 \(\geq\) SP14
  - Back-end based on AS ABAP \(< 7.0\ SP\ 14

- AS Java 7.0 \(\rightarrow\) AS ABAP where either of the following conditions apply for the ABAP back-end of WS provider:
  - Back-end based on AS ABAP 7.0 \(\geq\) SP14

- Back-end based on AS ABAP \(< 7.0\ SP\ 14

- AS ABAP 7.1 \(\rightarrow\) AS ABAP 7.1

Note that back-end 1 \(\rightarrow\) back-end 2 means: back-end 1 (WS consumer) calls back-end 2 (WS provider). In the case of Integration Server communication, the Integration Server (IS) is interconnected with the two back-
ends. In this case, the IS acts as the WS provider for back-end 1, whereas back-end 2 acts as WS provider for the IS.

In this document, we use the following abbreviated forms when referring different back-end stacks and releases:

- **AS ABAP 7.0 >= SP14** means: back-end based on AS ABAP 7.0 with support package stack higher or equal SP 14.
- **AS ABAP < 7.0 SP 14** means: back-end based on AS ABAP 7.0 support package stack lower than SP 14, or release lower than AS ABAP 7.0

### 2.1.2 Scenarios and Relevant Configuration Tasks

#### 2.1.2.1 Scenarios Using Integration Server Communication

In this type of communication, communication between the back-end systems (BE) is mediated by the Integration Server (IS). This can be a WS consumer in the sender system and a WS provider in the receiver system. The back-end systems can either be ABAP-based or Java-based.

This is illustrated in the following figure:

![Diagram of IS and BE systems](image)

Note that in this document ABAP-based back-ends are illustrated by a blue rectangle while Java-based or non-SAP back-ends are illustrated by an orange rectangle.

Configuring Integration Server communication between a WS consumer and a WS provider comprises the following tasks (as illustrated in the figure):

- (1) Configuring the receiver back-end system as WS provider
- (2) Configuring message processing on the Integration Server
- (3) Configuring the sender back-end system as WS consumer

Therefore, the following configuration tasks are necessary to configure the supported scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Configuration Tasks</th>
</tr>
</thead>
</table>
| **AS ABAP → AS ABAP,** where either of the following conditions apply for the back-ends of WS consumer or provider:  
  - Back-end based on AS ABAP 7.0 >= SP14  
  - Back-end based on AS ABAP < 7.0 SP 14 | Configuring WS Provider (AS ABAP 7.0 >= SP 14) (see section 2.2.2.1)  
Configuring WS Provider (AS ABAP < 7.0 SP 14) (see section: 2.2.2.2)  
Configuring WS Consumer (AS ABAP 7.0 >= SP14) (see section: 2.2.4.2)  
Configuring WS Consumer (AS ABAP < 7.0 SP 14) (see section: 2.2.4.3)  
Configuring Message Processing on IS (see section 2.2.3) |
| **AS Java 7.1 → AS ABAP,** | Configuring WS Provider (AS ABAP 7.0 >= SP 14) (see |
where either of the following conditions apply for the back-ends of WS consumer or provider:

- Back-end based on AS ABAP 7.0 >= SP14
- Back-end based on AS ABAP < 7.0 SP 14

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Java 7.0 → AS ABAP,</td>
<td>Configuring WS Provider (AS ABAP 7.0 &gt;= SP14) (see section 2.2.2.1)</td>
</tr>
<tr>
<td>where either of the following conditions apply for the back-ends of WS consumer or provider:</td>
<td>Configuring WS Provider (AS ABAP &lt; 7.0 SP 14) (see section 2.2.2.2)</td>
</tr>
<tr>
<td></td>
<td>Configuring WS Consumer (AS Java 7.0) (see section 2.2.4.5)</td>
</tr>
<tr>
<td></td>
<td>Configuring Message Processing on IS (see section 2.2.3)</td>
</tr>
<tr>
<td>AS ABAP 7.1 → AS ABAP 7.1</td>
<td>Configuring Communication Path from Consumer (AS ABAP 7.1) to IS (see section 2.2.5.1)</td>
</tr>
<tr>
<td></td>
<td>Configuring Communication Path From IS to Provider (AS ABAP 7.1) (see section 2.2.5.2)</td>
</tr>
</tbody>
</table>

Note that scenarios where the Web service consumer or provider communicates with applications running on a different technical protocol or standard are possible, for example using the IDoc adapter. However, this document focuses on the configuration settings of the Web service consumer/provider and the Integration Server only (provided Integration Server communication is available). Connectivity to other protocols or standards is described elsewhere.

More information: SAP NetWeaver Process Integration

### 2.1.2.2 Scenarios Using Direct Communication

Direct communication means that back-end systems communicate directly using WS runtime. The back-end systems can either be ABAP-based or Java-based.

This type of scenario is illustrated in the following figure:

![Direct Communication Diagram](image)

Configuring such scenarios involves the following tasks (as illustrated in the figure):

- (1) Configuring the receiver back-end system as the WS provider
- (2) Configuring the sender back-end systems as the WS consumer

Therefore, the following configuration tasks are required to configure the supported scenarios.
2.2 Configuration Tasks in Detail

This chapter covers the configuration steps in the back-end systems for direct and Integration Server communication as well as the configuration steps for message processing with Integration Server communication.

2.2.1 Overview of Configuration Tasks and Tools

In the table below you will find an overview of the relevant configuration tasks and tools needed to perform the different tasks:

**Configuration Tasks (Back-Ends Based on AS ABAP <= 7.0 and AS Java <= 7.1)**

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Configuration Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring WS Provider (AS ABAP 7.0 &gt;= SP14)</td>
<td>Transaction SOAMANAGER</td>
</tr>
<tr>
<td>Configuring WS Provider (AS ABAP &lt; 7.0 SP 14)</td>
<td>More information: Section 2.2.2.2</td>
</tr>
<tr>
<td>Configuring WS Consumer (AS ABAP 7.0 &gt;= SP14)</td>
<td>More information: Section 2.2.4.2</td>
</tr>
<tr>
<td>Configuring WS Consumer (AS ABAP &lt; 7.0 SP 14)</td>
<td>More information: Section 2.2.4.3</td>
</tr>
<tr>
<td>Configuring WS Provider (AS ABAP 7.0 &gt;= SP14)</td>
<td>More information: Section 2.2.2.1</td>
</tr>
<tr>
<td>Configuring WS Provider (AS ABAP &lt; 7.0 SP 14)</td>
<td>More information: Section 2.2.2.2</td>
</tr>
<tr>
<td>Configuring WS Consumer (AS Java 7.1)</td>
<td>More information: Section 2.2.4.4</td>
</tr>
<tr>
<td>Configuring WS Provider (AS ABAP 7.0 &gt;= SP14)</td>
<td>More information: Section 2.2.2.1</td>
</tr>
<tr>
<td>Configuring WS Provider (AS ABAP &lt; 7.0 SP 14)</td>
<td>More information: Section 2.2.2.2</td>
</tr>
<tr>
<td>Configuring WS Consumer (AS Java 7.0)</td>
<td>More information: Section 2.2.4.5</td>
</tr>
<tr>
<td>Configuring Communication Path from Consumer to Provider (Both AS ABAP 7.1)</td>
<td>More information: Section 2.2.5.3</td>
</tr>
</tbody>
</table>
| Configuring WS Provider (AS ABAP < 7.0 SP 14) | Transactions: \textit{WSCONFIG}, \textit{WSADMIN}, \textit{WSSPROFILE}  
More information: Section 2.2.2.1 |
|------------------------------------------------|----------------------------------------------------------------------------------|
| Configuring Inbound Processing (Integration Server) | System Landscape Directory  
Integration Directory  
More information: Section 2.2.3 |
| Configuring Outbound Processing (Integration Server) | Direct Communication:  
Transaction \textit{WSADMIN}  
More information: Section 2.2.4.1 |
| Finding out Destination of WS provider/Integration Server | Integration Server Communication:  
Integration Directory  
More information: Section 2.2.4.1 |
| Configuring WS Consumer (AS ABAP 7.0 >= SP14) | Transaction \textit{SOAMANAGER}  
More information: Section 2.2.4.2 |
| Configuring WS Consumer (AS ABAP < 7.0 SP 14) | Transactions: \textit{LPCONFIG}, \textit{SM59}  
More information: Section 2.2.4.2 |
| Configuring WS Consumer (AS Java 7.1) | SAP NetWeaver Administrator  
More information: Section: 2.2.4.4 |
| Configuring WS Consumer (AS Java 7.0) | Visual Administrator  
More information: Section 2.2.4.5 |

**General Sequence of Configuration Tasks When Configuring End-to-End Scenarios**

The following sections describe the individual configuration tasks for the WS consumer and WS provider independently from an end-to-end scenario.

To ensure a smooth configuration procedure SAP recommends the following general sequence of tasks for configuring end-to-end scenarios using direct communication:

1. Configure the WS provider.
2. Configure the WS consumer.

SAP recommends the following general sequence of tasks when configuring end-to-end scenarios using Integration Server communication:

1. Configure the WS provider.
2. Configure outbound processing (from the Integration Server to the WS provider).
3. Configure inbound processing (from the WS consumer to the Integration Server).
4. Configure the WS consumer.

If your scenario includes additional security settings (chapter 3), SAP recommends that you configure the security settings at the very beginning and then configure the basic communication using the sequence described above.

**Configuration Tasks (Back-Ends Based on AS ABAP 7.1)**

For scenarios that involve back-end systems based on AS ABAP 7.1, the complete communication paths (either WS consumer to IS, IS to WS provider, or WS consumer to WS provider) can be configured within the Integration Directory - for both direct and Integration
Server communication. This means that the general sequence of tasks recommended above does not apply for those scenarios.

The relevant configuration settings are propagated into the back-end systems by cache refresh as soon as the configuration objects are activated in the Integration Directory.

The following table lists the relevant configuration tasks:

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Configuration Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring Communication Path AS ABAP 7.1 to IS (Integration Server Communication) (see section 2.2.5.1)</td>
<td>System Landscape Directory</td>
</tr>
<tr>
<td>Configuring Communication Path IS to AS ABAP 7.1 (Integration Server Communication) (see section 2.2.5.2)</td>
<td>Integration Directory</td>
</tr>
<tr>
<td>Configuring Communication Path AS ABAP 7.1 to AS ABAP 7.1 (Direct Communication) (see section 2.2.5.3)</td>
<td></td>
</tr>
</tbody>
</table>

Note that you still have to use additional tools like the Trust Manager for setting up special security scenarios. To find out the necessary additional steps and tools, refer to chapter 3.

### 2.2.2 Configuring the WS Provider

The part of the configuration covered in this section is highlighted in the following figures:

**Direct communication:**

![Direct Communication Diagram]

**Integration Server communication:**

![Integration Server Communication Diagram]

You need to create and configure an endpoint to configure a WS provider. The endpoint contains a runtime configuration that is needed to implement the service.

You can create more than one endpoint for one and the same service. In this way you can provide the same service with several different runtime configurations.

The configuration procedure depends on the stack and version of the provider back-end.

---

Prerequisite: You have created a service definition.

For more information, see Providing Web Services.

### 2.2.2.1 Configuring the WS Provider (AS ABAP 7.0 >= SP 14)

You perform the necessary configuration tasks using the SOA Manager (transaction SOAMANAGER).

Follow the steps below to configure the ABAP back-end as a WS provider.
1. Log on to the back-end system of the WS consumer.
2. To start the SOA Manager, use the transaction code *SOAMANAGER*.
3. Select the *Business Administration* tab.
4. Choose *Web Service Administration*.
5. Specify the service or search for it.
6. Select a row and choose *Apply Selection* to display more details
7. Go to the *Configurations* tab.
8. Choose *Create Endpoint*.
   A dialog box is displayed.
9. Specify a unique name for the endpoint in the *Binding Name* field.
   An endpoint contains a single runtime configuration for a service. To define different runtime behaviors, you can create multiple endpoints for the same service which allow you to provide the same service with a different runtime behavior to different consumers.
10. Choose *Apply Settings*.
    The configuration settings are displayed.
11. Check that the configuration settings meet your requirements.
    To make changes, choose *Edit*.

You have the following configuration options for endpoints:

| Provider Security | Transport Guarantee Type: Here you determine if messages are signed or encrypted and how.
|                  | Using a signature confirms to the recipient that a message was not changed in transit – authenticity.
|                  | Encryption prevents a message from being seen in transit – integrity.
|                  | Authentication Settings: Here, you determine how the consumer and provider will authenticate themselves to each other. |
| Web Service Addressing | Specify the protocol for Web Service addressing and a reference parameter.
|                  | If only one protocol is supported, the protocol is pre-set.
|                  | If no protocols are supported, the field is empty. |
| Messaging | Set the protocol for reliable messaging (WSRM) and the confirmation interval.
|                  | If only 1 protocol is supported, the protocol is pre-set.
|                  | If no protocols are supported, the field is empty.
|                  | The confirmation interval (in seconds) is the time that the provider has to acknowledge receipt of a message to the consumer (Messages are returned asynchronously.)
|                  | The default timeout is 600 seconds. |
| Transport Settings | Configure an alternative path for messages.
|                  | The path that you set here overrides the path defined in the URL.
|                  | If the service is not local, for example behind a firewall, you need to specify
If the target WS can only be accessed through a proxy server, you can also specify some proxy information here.

| Message Attachments | Allow or disallow attachments. If you allow attachments, multiple files can be sent with a message. Only the message, not the attachments, then needs to be parsed by the recipient. |
| Operation specific | You can specify the SOAP action for the operation. The value in this field is a URI that identifies the intent of the call. The SOAP action is set as a HTTP header when the WS call is made using HTTP. |

You have to perform additional configuration tasks as explained in chapter 3 to ensure that the applied security settings work correctly.

12. Save the endpoint.

2.2.2.2 Configuring the WS Provider (AS ABAP < 7.0 SP 14)

You perform the necessary configuration steps using transactions WSCONFIG and WSADMIN (optional: WSSPROFILE).

To create an endpoint, perform the procedure as explained under Releasing for the SOAP Runtime.

The most important steps are summarized as follows:

1. Log on to the back-end system of the WS consumer.
2. Choose transaction WSCONFIG.
3. Search for the service definition (input help in field Service Definition).
4. Choose Create.
5. In the tab WS Settings, choose ICF Details.
6. Under Virtual Hosts/Services double-click the relevant service node.
   The dialog Create/Change a Service is opened.
7. Choose tab Logon Data.
8. In the Procedure input field you can choose an authentication method from the dropdown list box.
   More information: Defining the Logon Procedure
   Enter the authentication method required for your security scenario.
9. Save the endpoint and Web service.
10. To display and test the Web service and endpoint definition, choose transaction WSADMIN.
    Follow the instructions provided under Administration for the SOAP Runtime.

You need to assign a security profile to the service or specific security settings using signature and encryption.

If no security profile exists then you first have to create one. To do this, you need to call transaction WSSPROFILE. To create a security profile, follow the steps explained under Configuring Security.

After creating a security profile, assign it to your service.
To do this, call transaction **WSCONFIG** again and choose the **Operations** tab.
Under **Operation Settings**, assign the security profile either in the **Security Profile (Outbound)** field or **Security Profile (Inbound)**.

To make the applied security settings work correctly, you have to perform additional configuration tasks as explained in chapter 3.

### 2.2.3 Configuring Message Processing on the Integration Server

This section is only relevant if you are configuring scenarios using Integration Server communication.

![Diagram](diagram.png)

#### 2.2.3.1 Overview of Configuration Tasks

To configure message processing on the Integration Server you have to perform some tasks in the System Landscape Directory (SLD) and the Integration Directory.

In the SLD, define the systems that interact with each other (sender system, receiver system, and Integration Server system).

In the Integration Directory, define the following:

1. Communication components for the sender and receiver systems defined in the SDL.
2. Which adapter is used for inbound processing.
3. Receiver of the message (logical routing).
4. Inbound interface that is used at the receiver side.
5. Which adapter is used for outbound processing.

The following figure illustrates the general configuration procedure and points out the relevant configuration objects:
If you have defined a process integration scenario in the Enterprise Services Repository for communication between the WS consumer and WS provider then you can use a process integration scenario as configuration template. The necessary configuration settings are then automatically made by the system (where possible). When using this option, the configuration objects (for example the receiver determination or collaboration agreements) are generated automatically. This significantly decreases configuration time.

More information: Using the Process Model as a Configuration Template

This documentation, however, describes how to create all necessary configuration objects manually in order to show the interrelations with the other configuration tasks.

### 2.2.3.2 Defining the System Landscape in the System Landscape Directory

First, you have to define the technical and the business system landscape in the System Landscape Directory (SLD). The system landscape consists of the Integration Server system and the involved back-end systems.

1. To call the System Landscape Directory, choose Process Integration → Start Integration Builder in the user menu.
2. On the initial screen of SAP NetWeaver Process Integration, choose the System Landscape Directory link.
3. Log on using your user and password.

More information:
Tasks in the System Landscape Directory

Configuring Business Systems in the SLD

In the scenario illustrated in the figure you have to define systems for the following entities:

- One technical and business system for the Integration Server
- One technical and business system for the sender back-end
- One technical and business system for the receiver back-end

Classify the business systems for the back-ends as *application systems*.

For back-ends based on AS ABAP, select the *AS ABAP* system type, for back-ends based on AS Java, select the *Java Standalone* system type.

### 2.2.3.3 Defining Communication Components and Configuring Logical Routing

By performing the following tasks you:

- Define the involved business systems as communication components in the Integration Directory in order to make these systems addressable for messaging.
- Define the logical routing to specify the message flow at runtime.

For more information about the underlying concepts and the detailed procedures, please see [Integration Directory](#).

Perform the following tasks in the Integration Directory:

1. To call the Integration Directory, choose *Process Integration ➔ Start Integration Builder* in the user menu.
2. On the initial screen choose the *Integration Builder* link (under *Integration Directory*).
3. Log on using your user and password.
4. To address the back-end systems as the sender and receiver of messages, define communication components for each back-end system involved in the communication.
5. To determine the message flow from sender to receiver, define the required receiver determinations and interface determinations

More information: [Defining Logical Routing](#)

### 2.2.3.4 Configuring Outbound Processing (Integration Server)

Perform the following tasks in the Integration Directory:

1. Create a receiver communication channel for the communication component of the back-end system.

More information: Defining a [Configuring Communication Channels with Adapter Type WS](#)

2. Create a receiver agreement.

More information: [Defining a Receiver Agreement](#)

When creating the receiver agreement, execute the following:

- For the receiver communication component, select the business system component of the receiver back-end system.
- For the interface select the relevant inbound interface.
When editing the receiver agreement, perform the following:

- Assign the communication channel defined in step 1.
- Enter the attributes of the receiver agreement.

3. Save the receiver agreement.

### 2.2.3.5 Configuring Inbound Processing (Integration Server)

Perform the following tasks in the Integration Directory.

1. Create a sender communication channel for the communication component of the back-end system.
   
   More information: Defining a [Configuring Communication Channels with Adapter Type WS](#)

2. Create a sender agreement.
   
   More information: [Defining Sender Agreements](#)

When creating the sender agreement, perform the following:

- For the sender communication component, select the business system component of the sender back-end system.
- For the Interface, select the relevant outbound interface.

When editing the sender agreement, do the following:

- Assign the sender communication channel defined in step 1.
- Enter the attributes of the sender agreement.

3. Save the sender agreement.

4. Activate all created configuration objects in the Integration Directory.
2.2.4 Configuring WS Consumer

The part of the configuration covered in this section is highlighted in the following figures:

Direct communication:

Integration Server communication:

You configure a WS consumer – either ABAP-based or Java-based – by creating and configuring one or more runtime representations of its proxy definition called logical ports. A prerequisite is that you have defined and activated a consumer proxy.

More information: Generating a Consumer Proxy

Note that when creating a logical port you specify the physical destination of the WS provider located on a specific target system. In general, you can create destinations that point to Web services available on AS ABAP, AS Java, or any external system.

To create the destination, you have to know how the consumer application can find the Web service on the provider back-end. How the provider back-end can be found depends on the scenario.

The detailed procedure depends on whether the consumer and the provider communicates directly with each other or using the an Integration Server.

2.2.4.1 Finding Out the Destination for the WS Provider

You have already executed one of the following to find out the destination of the provider system (see 2.2.1 General Sequence of Configuration Tasks When Configuring End-to-End Scenarios):

- You have configured the endpoint of the WS provider (when using direct communication).
- You have already configured the inbound processing of the message sent from the WS consumer to the Integration Server (when using Integration Server communication).

In general, you can specify the destination in the logical port in the following way:

- As WSDL URL
- As endpoint URL

Note the following difference between direct and Integration Server communication.
Finding Out The Destination for Direct Communication

In this case you have to find out the destination of the WS provider.

If the WS provider is based on AS ABAP, you can find out the destination by performing the following steps:

1. In the back-end of the WS provider, choose transaction **WSADMIN**.
2. Select the Web service.
3. Choose **WSDL** or **Ctrl F1**.
4. Select **Document Style**.

   The WSDL for the endpoint is displayed.
   The WSDL URL is displayed in the address field of the browser.
   The endpoint URL is displayed in the following tag (example):
   ```xml
   
   In the example, the host is: PWDF2709.wdf.sap.corp
   
   In the example, the port is: 50009
   ```

Another option for finding the destination is to discover the endpoint of the WS provider using the Services Registry.

More information: [Services Registry](#)

Finding Out the Destination for Integration Server Communication

In this case, the Integration Server acts logically as WS provider. Therefore, you have to know the destination of the corresponding sender agreement when configuring the logical port for the WS consumer to point to the Integration Server (note that a sender agreement defines the inbound processing on the Integration Server).

To find out the destination of the sender agreement, execute the following:

1. Configure the inbound message processing (see 2.2.3.5).
   When performing this task, you define a sender agreement.
2. Display the WSDL of the sender agreement.
   To do this, select the sender agreement in the navigation area of the Integration Directory and select **Display WSDL** in the context menu.
   The WSDL URL is displayed in the field **WSDL URL**.
The endpoint URL can be extracted from the WSDL document (as highlighted in the example below).

```xml
<soap:address
location="http://www.sap.com:50091/sap/bc/srt/xip/sap/http177x
i2com7xiveri7wsdl1p20g/outboundstatelesspropertiessendandrecei
veg20404900/000/edc44fa7ce6531f0b4eb978362550aef/agr_A1C780A5B
66B3288827327E22E5418FC"/>
```

Note that in business-to-business scenarios, WS consumer and WS provider (respectively, Integration Server) are typically separated by a firewall. In these cases, the URL determined by the steps described above has to be changed manually since it contains an internal host name.

### 2.2.4.2 Configuring the WS Consumer (AS ABAP 7.0 >= SP14)

You perform the necessary configuration steps using the SOA Manager (transaction `SOAMANAGER`).

More information on the prerequisites: 2.2.4.1

The detailed procedure depends on the Support Package of AS ABAP that the back-end is based on.

In this case, you perform the necessary configuration tasks in the SOA Manager:

1. Log on to the back-end system of the WS consumer.
2. To start the SOA Manager, use the transaction code `SOAMANAGER`.
3. Select `Web Service Administration`.
4. Specify the consumer proxy or search for it.
5. Select the consumer proxy from the overview and choose `Apply Selection` to display an overview of design time information.
6. Go to the `Configurations` tab.
7. To create a new logical port, choose `Create Logical Port`.
   A dialog box is displayed.
8. Specify a name for the service.
9. Specify a name for the new logical port.
   To make this logical port the default logical port, select `Logical Port is Default`. If a consumer application is not configured to call a specific logical port, it calls the default logical port.
10. Select the configuration type.
    For WSDL-based configuration, the system extracts the endpoint URL from the WSDL document for the service.
    For manual configuration, the system uses a user-specified endpoint URL to create a logical port. You need to specify a unique name for the logical port.
    Select `Manual Configuration` if a WSDL of the service provider is not available.
11. For WSDL-based configuration, specify the WSDL access settings.
    You have the following options:
    - By HTTP access
Specify the path to the WSDL document, a user, and a password.

For more information on finding out the path to the WSDL document, see 2.2.4.1.

- By File
  Specify a file for WSDL Access

12. Choose Apply Settings.

The default settings for the logical port are displayed.

13. Check the configuration.

The following configuration options are available:

<table>
<thead>
<tr>
<th>Consumer Security</th>
<th>User and Password for Authentication with the Service Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Service Addressing</td>
<td>Specify the protocol for Web service addressing. A reference parameter is optional. &lt;br&gt; If only 1 protocol is supported, the protocol is pre-set. &lt;br&gt; If no protocols are supported, the field is empty.</td>
</tr>
<tr>
<td>Messaging</td>
<td>Set the protocol for reliable messaging (WSRM) and the confirmation interval. &lt;br&gt; <strong>RM Protocol</strong>: Specify the protocol for reliable messaging. &lt;br&gt; If only 1 protocol is supported, the protocol is pre-set. &lt;br&gt; If no protocols are supported, the field is empty. &lt;br&gt; <strong>Interval in Seconds</strong>: The time in which an acknowledgement can be sent. After this time has elapsed, the consumer tries to resend a message that has not been acknowledged by the provider. &lt;br&gt; <strong>Inactivity Timeout</strong>: The time (in seconds) to hold the sequence open. If no acknowledgement is received after this time has elapsed, the sequence is considered to be terminated, and a rollback is performed (0 seconds is infinite). &lt;br&gt; <strong>Sequence Lifetime</strong>: The time for which a sequence remains active (0 seconds is infinite). &lt;br&gt; <strong>Message Identifier</strong>: Each message needs an ID for two protocols, WS-A Message ID and SAP Message ID. Suppress ID transfer – this option is for older releases.</td>
</tr>
<tr>
<td>Transport Settings</td>
<td>The settings for transport between the provider and the consumer. The settings here are stored together in the complete URL. &lt;br&gt; <strong>Make Local Call</strong>: A local call is only used if processing is done within the same provider system. If local only calls are made, unnecessary steps, such as authentication, can be omitted. &lt;br&gt; <strong>Maximum Wait for WS Consumer</strong>: The HTTP timeout determines how long the consumer waits for a response from the provider. If you expect a high load on the provider system that would delay the responses to consumers, consider increasing the default value (0 seconds is infinite). &lt;br&gt; <strong>Optimized XML Transfer</strong>: Determines the XML transfer between consumer and provider. If both the consumer and the provider are ABAP systems, you can select binary XML to reduce the conversion overhead between these systems.</td>
</tr>
</tbody>
</table>
**Compress HTTP Message**: Allows the consumer to send zip-compressed requests. This option is only of interest if you are expecting very large message sizes. For normal message sizes, the compression would take too long to be any benefit.

**Accept Compressed HTTP Message**: Allows the provider to return zip-compressed responses to the consumer. This option is useful if you expect the provider to return data volumes of several MB in size.

<table>
<thead>
<tr>
<th>Message Attachments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow or disallow attachments. If you allow attachments, multiple files can be sent with a message. Only the message, not the attachments, then needs to be parsed by the recipient.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation-Specific</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If a message is assigned to an operation, the operation parameter can be transferred in the HTTP header. This means that the message body does not need to be parsed for the operation parameter.</td>
<td></td>
</tr>
</tbody>
</table>

14. Save the logical port.

### 2.2.4.3 Configuring WS Consumer (AS ABAP < 7.0 SP 14)

You perform the necessary configuration steps using transaction **LPCONFIG**.

When creating a logical port, you have different options to specify the destination of the endpoint in the provider system (which is either the Integration Server or the receiver back-end). However, we recommend that you first create an RFC destination from the consumer to the provider system (using transaction **SM59**) and then assign this destination to the logical port (in transaction **LPCONFIG**).

When you first define an RFC destination, you can already specify security settings within the destination.

Perform the following tasks when configuring the WS consumer.

**Create RFC Destination to Target System**

Note the following difference between direct and Integration Server communication:

- Integration Server communication: Target system is the Integration Server.
- Direct communication: Target system is the back-end where the WS provider is implemented.

More information: Section 2.2.4.1

To create an RFC destination for the target system, perform the procedure as explained under **Establishing a Connection Using a Destination (SM59)**.

The most important steps are summarized as follows:

1. In the sender back-end, call transaction **SM59**.
2. Choose **Create**.
3. Enter a name for the RFC destination and specify the **Connection Type**.
   - In the case of Integration Server communication, select type **H** *(HTTP Connection to ABAP System)*.
   - In the case of direct connection with an ABAP-based WS provider, select type **H** *(HTTP Connection to ABAP System)*.
   - In the case of direct connection with a Java-based WS provider, select type **G** *(HTTP Connection to External System).*
4. Save the destination.

5. In the tab **Technical Settings** make the following entries:
   - **Target Host**: Enter the host of the target system, for example **pwdf1234**.
   - **Service Number**: Enter the service number of the HTTP or HTTPS port, for example **8080**.
   - **Path Prefix**: SAP recommends leaving the path prefix open. When doing this, the destination can be used for all Web services located on the specified host. Later on, when defining a logical port, you enter the complete path to specify the Web service.
     For more information on how to find out the destination of the WS provider, see 2.2.4.1.

6. On the **Logon & Security** tab you can specify security settings.
   These settings depend on the overall security scenario (section 3).
   For example, if you set up communication between the WS consumer and WS provider using X.509 client certificates (as described in detail in section 4.1) under Logon Procedure you select **SAP Trusted System** and under Status of Secure Protocol/SSL you select **Activate**. Then select the client certificate that you have specified when defining the trusted relationship in transaction STRUST.

7. Save the destination.

**Define Logical Port**

To define a logical port for the consumer proxy, perform the procedure as explained under [Creating a Logical Port](#).

The most important steps are summarized as follows:

1. In the back-end of the sender, call transaction **LPCONFIG**.
2. Select the consumer proxy.
   Use the input help in field **Proxy Class**.
3. Specify a name for the logical port (field **Logical Port**).
4. Choose **Create**.
5. Enter a description for the logical port.
6. Under **Call Parameters**, specify the following attributes.
   - Select the RFC destination defined in the preceding task. Use the input help in the field **HTTP Destination**.
   - In field **Path Prefix**, enter the part of the endpoint URL that follows the host and port.
     More information: Section 2.2.4.1

7. Save the logical port.

**2.2.4.4 Configuring the WS Consumer (AS Java 7.1)**

Perform the configuration steps that are necessary using the SAP NetWeaver Administrator.

More information on the prerequisites: 2.2.4.1

You can find more information on the overall concept and on SAP NetWeaver Administrator under [Configuring WSs and WS Clients in the SAP NetWeaver Administrator](#).
For a detailed description of the configuration steps, see: [Configuring WS Clients](#).
For more information on how to find out the destination of the WS provider, see 2.2.4.1.
For more information on the underlying concepts when using Java-based WS consumers together with the Integration Server, see [Creating and Configuring WS Providers and Consumers for Brokered Communication](#).

### 2.2.4.5 Configuring WS Consumer (AS Java 7.0)

You configure an AS Java 7.0 as WS consumer using the Visual Administrator.

More information: [Visual Administrator](#)

More information on the prerequisites: 2.2.4.1

Perform the following steps:

1. Call the Visual Administrator.
   
   If you have installed SAP NetWeaver 7.0, you can start the Visual Administrator by starting the application `go.bat`, located in the following directory: `C:\usr\sap\<SID>\JC<NN>\j2ee\admin\go.bat`.

   - `<SID>`: ID of the system, which you have installed
   - `JC<NN>`: number of the Java instance of the system

   Example: `C:\usr\sap\LKG\JC70\j2ee\admin\go.bat`

   More information: [How to Start the Visual Administrator](#)

2. Configure the logical port.

   More information: [Configuring Logical Ports](#)

   When configuring the logical port, in the *Runtime* tab specify the WSDL URL of either the WS provider or the Integration Server (depending on whether you configure a scenario using direct communication or Integration Server communication).

   More information: 2.2.4.1

   Depending on the security scenario (see chapter 3), select the relevant logon data and authentication method in the *Authentication* dropdown box.

### 2.2.5 Configuring Communication Paths for Back-Ends Based on AS ABAP 7.1

For scenarios using WS consumer and WS provider based on AS ABAP 7.1, the communication can be configured centrally in the Integration Directory. That means that all configuration settings - for both the back-end systems and message processing - are done centrally in the Integration Directory. This central configuration is supported for Integration Server communication as well as for direct communication.

When you activate the configuration objects in the Integration Directory, the configuration settings relevant for the WS consumer or WS provider are propagated to the back-end systems by cache notifications.

That means that when configuring scenarios (including back-end systems based on AS ABAP 7.1) it makes no sense to differentiate between back-end configuration and message processing configuration. The configuration of a complete communication path (either back-end to IS, IS to back-end or back-end to back-end) is done within one configuration task.

The configuration tasks for the relevant communication paths are described in the following sections.
As a prerequisite, the tasks in the System Landscape Directory have to be performed. These are mentioned below.

### 2.2.5.1 Configuring Communication Path from the Consumer (AS ABAP 7.1) to the IS

This configuration task applies for sender systems based on AS ABAP 7.1.

Within this communication path, the WS consumer based on AS ABAP 7.1 sends a message to the Integration Server (IS). The communication path is illustrated in the following figure:

![Communication Path Diagram]

To configure this communication path, the following configuration objects play a major role within the Integration Directory:

- Sender agreement
- Sender communication channel (adapter type WS)

With the tasks described below you configure the ABAP back-end as the WS consumer as well as the inbound processing of the message on the IS.

### Defining the System Landscape in the System Landscape Directory

First, you have to define the technical and business system landscape in the System Landscape Directory (SLD). The system landscape consists of the Integration Server system and the sender back-end system.

Classify the business systems for the back-end systems as application systems and select the AS ABAP system type.

More information:
- [Tasks in the System Landscape Directory](#)
- [Configuring Business Systems in the SLD](#)

### Defining a Communication Component

Define a communication component for the sender back-end system.

More information: [Configuring Business Systems (WS-RM-enabled applications)](#)

### Configuring a Communication Path

1. Create a sender communication channel for the communication component of the back-end system.

   More information: [Configuring Communication Channels with Adapter Type WS](#)

   In the communication channel you specify different security settings for the WS consumer calling the Integration Server. Note that you have to maintain different prerequisites for each security.

   You can find more information on the special prerequisites for the security settings used in section 3.

2. Create and edit a sender agreement.

   More information: [Defining Sender Agreements](#)

   Note the following when creating/editing the sender agreement:
When creating the sender agreement, select the business system component of the sender back-end system in the object key (as sender communication component) and the relevant outbound interface (as interface).

Assign the communication channel defined in step 1.

Since you have assigned the sender agreement a communication channel with adapter WS, you can specify further adapter-specific attributes on the Parameters tab page.

Note that the adapter-specific attributes are only displayed in case the business system assigned in the key of the sender agreement (communication component field) represents a back-end system based on AS ABAP 7.1.

More information: Adapter-Specific Attributes

3. Activate the sender agreement and the communication channel.

Those configuration settings relevant for the WS consumer are propagated to the back-end system of the WS consumer.

More information: Change Lists

4. Check the cache status.

Only once the relevant caches contain the necessary information can you be sure that the configuration settings are propagated to the back-end systems involved.

More information: Analyzing Cache Notifications

2.2.5.2 Configuring Communication Path from the IS to Provider (AS ABAP 7.1)

This configuration task applies to receiver systems based on AS ABAP 7.1.

Within this communication path, the Integration Server (IS) acts as the WS consumer and sends a message to the WS provider implemented on the receiver back-end. The communication path is illustrated in the following figure.

To configure this communication path, the following configuration objects play a major role within the Integration Directory:

- Receiver agreement
- Receiver communication channel (adapter type WS)

Use the tasks described below to configure the ABAP back-end as the WS provider as well as the outbound processing of the message on the IS.

Proceed as described under 2.2.5.1.

Note the following differences:

Instead of a sender channel, create a receiver channel of type WS for the receiver business system.

More information: Defining a Configuring Communication Channels with Adapter Type WS
In the communication channel, you specify different security settings for the Integration Server calling a WS provider.

Note that for each security setting you have to ensure different prerequisites as described in section 3.

Instead of a sender agreement, create a receiver agreement for the receiver business system. Specify the inbound interface and receiver channel for the WS call.

More information: Defining a Receiver Agreement

2.2.5.3 Configuring Communication Path from the Consumer to Provider (Both AS ABAP 7.1)

This configuration task applies for sender and receiver systems based on AS ABAP 7.1.

Within this communication path, the sender back-end acts as WS consumer and sends a message to the receiver back-end. The receiver back-end acts as WS provider. **No Integration server is interconnected between sender and receiver.** The communication path is illustrated in the following figure.

You can make the necessary configuration settings for the back-ends **centrally** in the Integration Directory rather than having to make them locally in the back-end systems involved.

To configure this communication path, the following configuration objects play a major role within the Integration Directory:

- Direct connection
- Receiver communication channel (adapter type WS)

With the tasks described below you configure the ABAP back-end as WS provider as well as the outbound processing of the message on the IS.

More information: Configuring Direct Communication
3 Security Configuration

This chapter sums up the supported security scenarios for communication between the WS consumer and WS provider and refers to detailed documentation of the relevant configuration tasks.

This section summarizes the configuration tasks that are - in addition to the tasks for configuring the basic communication settings in chapter 2 - necessary in order to configure the individual security scenarios.

Due to the fact that the configuration tasks depend on the stack and version of the back-end systems involved - for each back-end stack/version the necessary configuration tools are listed (analog to the chapter about the basic communication settings).

3.1 Supported Security Scenarios

The table below lists the necessary configuration tasks for each security scenario supported--for both direct and Integration Server communication.

Note that each scenario is a combination of an authentication mechanism and a security setting for the connection (for example, SSL).

Since the configuration procedure for each task depends heavily on the stack and release of the corresponding back-end, the configuration tasks are listed for each back-end stack and release separately in section 3.2.

Supported Security Scenarios and Corresponding Configuration Tasks (see section 3.2)

<table>
<thead>
<tr>
<th>Security Scenario</th>
<th>Configuration Tasks</th>
</tr>
</thead>
</table>
| SAP Authentication Assertion Ticket and SSL | Configuring the AS ABAP for Supporting SSL  
Configuring the Use of SSL on the AS Java  
Configuring the AS ABAP for Issuing Logon Tickets  
Configuring the AS ABAP to accept Logon Tickets  
Additional Configuration Tasks in Integration Directory (Integration Server communication only) |
| WS UsernameToken and WS SecureConversation | Configuring the AS ABAP for Supporting SSL  
Configuring the Use of SSL on the AS Java |
| User ID and Password in the HTTP Header and SSL | Configuring the AS ABAP for Supporting SSL  
Configuring the Use of SSL on the AS Java |
| X.509 SSL Client Certificate (Authentication at Transport Level Using HTTPS) (Explained in Detail for Back-Ends Based on AS ABAP in Section 4.1 (End-to-End)) | Configuring the AS ABAP for Supporting SSL  
Configuring the AS ABAP to Use X.509 Client Certificates  
Configuring the Use of SSL on the AS Java  
Additional Configuration Tasks in Integration Directory (Integration Server communication only) |
### 3.2 Configuration Tasks and Tools

In the tables below you will find an overview of the relevant configuration tasks and the tools that are necessary to perform those different tasks. The tables are grouped according to back-end stack/release.

The tables list the individual configuration tasks and provide links to detailed documentation of the configuration tasks.

#### 3.2.1 Configuration Tasks for Back-Ends Based on AS ABAP

**>= 7.0 SP 14**

In this case, the most important tools are the SOA Manager and several back-end transactions (for example, transaction \STRUST\).

For an overview of the concepts for Web Services security within SAP NetWeaver PI 7.1, as well as the underlying infrastructure, see the Security Guide WSs.

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the AS ABAP for</td>
<td>More information: Configuring the AS ABAP for Supporting</td>
</tr>
</tbody>
</table>

**WS Security:**
Signature Authentication and Asymmetrical Encryption

- Preparing the WS Provider AS ABAP for Accepting the Signature
- Preparing the WS Provider AS ABAP for Signature Authentication
- Importing an Encryption Certificate into WS Consumer AS ABAP
- Configuring WS-Security XML Signature & Encryption for J2EE Engine
- Additional Configuration Tasks in Integration Directory (Integration Server communication only)

**SAML and WS Secure Conversation**

- Configuring the AS ABAP for Supporting SSL
- Configuring the Use of SSL on the AS Java
- Configuring the SAP System to Accept SAML Assertions
- Additional Configuration Tasks in Integration Directory (Integration Server communication only)

**Principal Propagation (SAP Assertion Ticket) (Integration Server communication only)**

- Configuring the AS ABAP for Supporting SSL
- Configuring the Use of SSL on the AS Java
- Configuring the AS ABAP for Issuing Logon Tickets
- Configuring the AS ABAP to Accept Logon Tickets
- Additional Configuration Tasks in Integration Directory (Integration Server communication only)

**Principal Propagation (SAML) (Integration Server communication only)**

- Configuring SSO with SAML Token Profiles
- Additional Configuration Tasks in Integration Directory (Integration Server communication only)
Supporting SSL

<table>
<thead>
<tr>
<th>Task</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the AS ABAP to Use X.509 Client Certificates</td>
<td>More information: Configuring the AS ABAP to Use X.509 Client Certificates</td>
</tr>
<tr>
<td>Configuring the AS ABAP for Issuing Logon Tickets</td>
<td>More information: Configuring the AS ABAP for Issuing Logon Tickets</td>
</tr>
<tr>
<td>Configuring the AS ABAP to Accept Logon Tickets</td>
<td>More information: Configuring the AS ABAP to Accept Logon Tickets</td>
</tr>
<tr>
<td>Preparing the WS Provider AS ABAP for Accepting the Signature</td>
<td>WS-Security: Signature authentication &amp; asymmetrical encryption</td>
</tr>
<tr>
<td>Preparing the WS Provider AS ABAP for Signature Authentication</td>
<td>Signatures: Preparing the WS Provider AS ABAP for Accepting the Signature</td>
</tr>
<tr>
<td>Importing an Encryption Certificate into WS Consumer AS ABAP</td>
<td>Encryption: Preparing the WS Consumer AS ABAP for Encryption</td>
</tr>
<tr>
<td>Configuring the SAP system to Accept SAML Assertions</td>
<td>More information: Configuring SSO with SAML Token Profiles</td>
</tr>
<tr>
<td>Configuring SSO with SAML Token Profiles</td>
<td>More information: Configuring SSO with SAML Token Profiles</td>
</tr>
<tr>
<td>Additional Configuration Tasks in Integration Directory</td>
<td>More information: Configuring Communication Channels with Adapter Type WS</td>
</tr>
</tbody>
</table>

3.2.2 Configuration Tasks for Back-Ends Based on AS ABAP < 7.0 SP 14

In this case, the most important tools are several back-end transactions (for example, transaction `STRUST`).

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the AS ABAP for Supporting SSL</td>
<td>More information: Configuring the AS ABAP for Supporting SSL</td>
</tr>
<tr>
<td>Configuring the AS ABAP to Use X.509 Client Certificates</td>
<td>More information: Configuring the System for Using X.509 Client Certificates</td>
</tr>
<tr>
<td>Configuring the AS ABAP for Issuing Logon Tickets</td>
<td>More information: Configuring AS-ABAP to Accept and Verify Logon Tickets</td>
</tr>
<tr>
<td>Configuring the AS ABAP to Accept Logon Tickets</td>
<td>More information: Configuring AS-ABAP to Accept and Verify Logon Tickets</td>
</tr>
<tr>
<td>Preparing the WS Provider AS ABAP for Accepting the Signature</td>
<td>More information: WS-Security XML Signature/Encryption</td>
</tr>
<tr>
<td>Preparing the WS Provider AS ABAP for Signature</td>
<td>More information: WS-Security XML Signature/Encryption</td>
</tr>
</tbody>
</table>
### Authentication

<table>
<thead>
<tr>
<th>Importing an Encryption Certificate into WS Consumer AS ABAP</th>
<th>More information: <a href="#">WS-Security XML Signature/Encryption</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Configuration Tasks in Integration Directory</td>
<td>More information: <a href="#">Configuring Communication Channels with Adapter Type WS</a></td>
</tr>
</tbody>
</table>

#### 3.2.3 Configuration Tasks for Back-Ends Based on AS Java 7.1

In this case, the most important tool is the SAP NetWeaver Administrator.

For an overview of the concepts of Web Services security within SAP NetWeaver PI 7.1, as well as the underlying infrastructure, see the Security Guide WSs.

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the Use of SSL on the AS Java</td>
<td>More information: <a href="#">Configuring the Use of SSL on the AS Java</a></td>
</tr>
<tr>
<td>Configuring WS-Security XML Signature &amp; Encryption for AS Java</td>
<td>Signatures: <a href="#">Preparing the WS Consumer AS Java for Issuing the Signature</a>, <a href="#">Preparing the WS Provider AS Java for Accepting the Signature</a>, <a href="#">Preparing the WS Provider AS Java for Signature Authentication</a> Encryption: <a href="#">Preparing the WS Consumer AS Java for Encryption</a>, <a href="#">Exporting an Encryption Certificate for the WS Provider AS Java</a></td>
</tr>
<tr>
<td>Configuring the SAP system to Accept SAML Assertions</td>
<td>More information: <a href="#">Configuring SSO with SAML Token Profiles</a></td>
</tr>
<tr>
<td>Configuring SSO with SAML Token Profiles</td>
<td>More information: <a href="#">Configuring SSO with SAML Token Profiles</a></td>
</tr>
<tr>
<td>Additional Configuration Tasks in Integration Directory</td>
<td>More information: <a href="#">Configuring Communication Channels with Adapter Type WS</a></td>
</tr>
</tbody>
</table>

#### 3.2.4 Configuration Tasks for Back-Ends Based on AS Java < 7.1

In this case, the most important tool is the Visual Administrator.

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the Use of SSL on the AS Java</td>
<td>More information: <a href="#">Configuring the Use of SSL on the AS Java</a></td>
</tr>
<tr>
<td>Configuring WS-Security XML Signature &amp; Encryption for AS Java</td>
<td>More information: <a href="#">WS Security</a></td>
</tr>
</tbody>
</table>
### SAP NetWeaver Process Integration - Configuring Web Service Scenarios

#### Java

<table>
<thead>
<tr>
<th>Additional Configuration Tasks in Integration Directory</th>
<th>More information: Configuring Communication Channels with Adapter Type WS</th>
</tr>
</thead>
</table>

---

32 April 2008

April 2008
4 End-to-End Task Descriptions

4.1 Integration Server Communication Using X.509 Client Certificate (Back-Ends Based on AS ABAP < 7.0 SP 14)

In this section, we consider the scenario where sender back-end (WS consumer) and receiver back-end are both based on AS ABAP < 7.0 SP14 and an Integration Server (SAP NetWeaver PI 7.1) is interconnected in the communication.

The following communication paths have to be considered:
- WS consumer (on ABAP back-end) calls Integration Server.
- Integration Server calls WS provider (on ABAP back-end).

The communication flow is illustrated in the following figure:

The following security features will be configured:
Authentication using X.509 SSL client certificate (authentication using HTTPS)

4.1.1 Overview of Configuration Tasks

The configuration comprises the following parts:

<table>
<thead>
<tr>
<th>Configuration Task</th>
<th>Configuration Setting</th>
<th>Configuration Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Setup</td>
<td>Setup SSL connection</td>
<td>Transaction SMICM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICM Tracing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing the SSL Configuration</td>
</tr>
<tr>
<td></td>
<td>Maintain certificates for trust relationships</td>
<td>Transaction STRUST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trust Manager</td>
</tr>
<tr>
<td></td>
<td>User mapping</td>
<td>Transaction SM30 (table VUSREXTID)</td>
</tr>
<tr>
<td>Configuring ABAP 7.0 Back-End as WS Consumer</td>
<td>Defining a logical port</td>
<td>Transaction LPCONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating a Logical Port</td>
</tr>
<tr>
<td>Configuring Message Processing on Integration Server</td>
<td>Defining business systems in SLD</td>
<td>System Landscape Directory</td>
</tr>
<tr>
<td></td>
<td>Defining business systems in Integration Directory</td>
<td>Integration Directory</td>
</tr>
<tr>
<td></td>
<td>Defining logical routing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defining inbound and</td>
<td></td>
</tr>
</tbody>
</table>
| Configuring ABAP 7.0 Back-End as WS Provider | Defining an endpoint | Transaction WSCONFIG  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>outbound processing on Integration Server</td>
<td></td>
<td>Transaction WSADMIN</td>
</tr>
</tbody>
</table>

### 4.1.2 Configure Security Settings for X.509 Authentication (Using HTTPS)

Follow the steps below to configure the security settings in the back-end systems and in the system of the Integration Server:

**Overview of the steps:**
- Checking HTTPS and SSL connection between the back-end systems and the system of the Integration Server.
- Setting up secure communication between the sender back-end and the Integration Server
- Setting up secure communication between the Integration Server and the receiver system

#### 4.1.2.1 Checking HTTPS and SSL Connection

Perform the following steps in each back-end system as well as in the Integration Server system.
1. Choose transaction ICM Monitor (`SMICM`).
2. Choose Services (§).
3. Check if a check mark (column Actv) is set for the protocol HTTPS.

#### 4.1.2.2 Setting up Secure Communication Between Sender and IS

Communication between the sender and the Integration Server is based on certificates as illustrated in the following figure. In our example, the Integration Server has a certificate that is signed by a certification authority, whereas the sender has a self-signed certificate.
Building up secure communication between the sender and Integration Server (IS) is composed of the following parts:

1. Sender back-end trusts IS
   If the SSL connection between the sender and IS is active ("SSL handshake"), the sender has to trust the server certificate of the IS. This certificate is signed by a certification authority. To set up this part of the secure communication, this certification authority has also to be made known to the sender.

2. IS trusts client certificate of the sender
   If the sender calls the IS, the authentication of the sender is checked on the basis of a self-signed client certificate. Since the certificate is self-signed, it has to be made known to the IS outside the SSL connection, for example by secure email exchange between the administrators of the sender system and the Integration Server system.

3. User mapping
   The sender self-authenticates with the client certificate when calling the IS. However, when calling a system, a service user is needed. Therefore, the client certificate has to be mapped to a (service) user name.

The configuration steps necessary to set up the three parts of the communication are explained in detail below.

The following figure illustrates on which system the individual configuration steps are performed.
Making available certificate signed by a certification authority

1. Import server certificate signed by SAP CA

Making sure that sender trusts IS

➔ install IS certificate on sender system

2. Export server certificate from IS system

3. Import server certificate on sender system

Making sure that IS trusts client certificate in message of sender

➔ install sender certificate on IS

4. Export client certificate from sender system

5. Import client certificate on IS

Making Certificate Signed by Certification Authority Available

In our example, we use a certificate signed by the certification authority SAP Trust Center Services.

To access this certificate, do the following:


2. Under Root CA Certificates of SAP TCS select SAP Server CA Certificate.

3. Save the certificate on your hard disc.
4. In the system of the Integration Server, call the Trust Manager (transaction STRUST).

5. Select the SSL Server Standard node.


7. Select the certificate signed by SAP Trust Center Services (saved on your hard disc).

After importing the certificate, it has to be exported to the certificate database. To do this, perform the following steps:
2. Specify the name of the Trust Center (in this example, SAPNET).

Perform the following steps to create a certificate request.

1. Select the node SSL Server Standard.
3. Copy the certificate request to the clipboard or save to a file.
5. Select SSL Test Server Certificates.
6. Select Test it Now!
7. Paste the certificate request from the clipboard into the input field.

8. Select Continue.
9. Copy the certificate to your clipboard.
10. In the Trust Manager (transaction STRUST), select the node **SSL Server Standard**.
12. Paste the signed server certificate from SAP Trust Center Services (from the clipboard).
13. Select **Input** to confirm.

### Making Sure that the Sender Trusts IS

The sender back-end has to trust the server certificate of the Integration Server. The certificate in this example is signed by a certification authority (**SAP Trust Center Services**). Therefore, the certificate of **SAP Trust Center Services** has to be added to the certificate list within the Personal Security Environment (PSE) of the sender back-end (in this case, the...
standard PSE of the sender). This step is necessary to make this certification authority also known to the sender.

In the system of the Integration Server, first export the server certificate.

To do this, perform the following steps:

1. In the system of the Integration Server, call the Trust Manager (transaction STRUST).
2. Double-click the node SSL server Standard.
3. Under Own Certificate double-click the name of the certificate.
5. Specify the file path to save the certificate on your local hard disc.

In the sender system, import the certificate previously exported to your local hard disc.

To do this, perform the following steps:

1. In the sender system, call the Trust Manager (transaction STRUST).
2. Double-click the node SSL Client (Standard).
4. Select the certificate from your local hard disc.
5. Select Add to Certificate List.
Making Sure that IS Trusts the Client Certificate in the Sender Message

The Integration Server has to trust the client certificate during SSL handshake (SSL connection establishment). To display the certificate, call the Trust Manager (transaction STRUST) in the sender back-end system and double-click the node SSL Client (Standard).

Note that the certificate in question is called a client certificate since it is the certificate that enables the sender to act as WS client and call the IS.

Since the client certificate is self-signed, there is no third authority that guarantees the trustworthiness of the certificate. Due to this, the certificate has to be made known in the Integration Server system by communication outside the actual SSL connection (for example, by using secure email exchange). In order to exchange the client certificate, you must first export the certificate from the sender back-end system (for example to your hard disc). After this, the certificate can be transferred to the owner of the Integration Server system (for example, by email). At the Integration Server side, the certificate then has to be imported into the SSL Server Standard PSE. Note that at the Integration Server side, the certificate is stored under the SSL Server Standard PSE as it enables the IS to act as the server and to accept the call from the sender.

To establish this part of the trusted relationship, you have to do the following.

You have to export the certificate from the sender system. To do this, perform the following steps:

1. In the sender system, call the Trust Manager (transaction STRUST).
2. Double-click the node SSL Client (Standard).
3. Under Own Certificate double-click the certificate.
5. Save the certificate on your local hard disc.

You have to import the certificate to Integration Server system. To do this, execute the following steps:

1. In the system of the Integration Server, call the Trust Manager (transaction STRUST).
2. Double-click the node SSL server Standard.
4. Select the certificate from your local hard disc.
5. Select Add to Certificate List.
4.1.2.3 Setting up Secure Communication Between the IS and Receiver

Communication between the Integration Server and receiver is based on certificates as illustrated in the following figure. In our example, the IS has a certificate that is signed by a certification authority, whereas the receiver has a self-signed certificate.
Building up a secure communication between IS and receiver is composed of the following parts (analogous to communication between sender and IS):

1. IS trusts receiver back-end

   If the SSL connection between IS and receiver is active, the IS has to trust the server certificate of the receiver back-end. In our example, this certificate is self-signed. Because of this, it has to be made known to the IS outside the SSL connection, for example by secure email exchange between the administrators of the receiver system and the Integration Server system.

2. Receiver back-end trusts certificate of the IS

   When the IS calls the receiver, authentication of the IS is done on the basis of a certificate signed by a certification authority. To set up this part of the secure communication, the certificate of the certification authority has to be imported into the SSL Server PSE of the receiver.

3. User mapping

   The IS self-authenticates with the client certificate when calling the receiver. However, a service user is needed when calling a system. Therefore, the client certificate has to be mapped to a (service) user name.

The individual configuration steps of all three tasks are explained in detail below.

The following figure illustrates which system the individual configuration steps are performed on.
SAP NetWeaver Process Integration - Configuring Web Service Scenarios

Making sure that IS trusts receiver
→ install receiver certificate on system of IS

1. Export server certificate from receiver system
2. Import server certificate on system of IS

Making sure that receiver system trusts client certificate in message from IS
→ install IS client certificate on receiver system

3. Export client certificate from system of the IS
4. Import client certificate on receiver system

Making Sure That IS Trusts the Receiver

The IS trusts the SSL server certificate of the receiver. Since this certificate is self-signed, it has to be exported from the receiver system, transferred to the owner of the IS by a secure communication outside the SSL connection, and then imported into the appropriate client PSE of the IS (since here the IS acts as client).

Perform the following steps:

1. Call the Trust Manager (transaction STRUST) in the receiver back-end and double-click the SSL Server node.
2. Double-click the self-signed certificate (under Own Certificate, field Owner).
4. Choose the file format Base 64.
5. In the File Path field select a location on your hard disc where the exported certificate should be stored.
6. Enter a name of the locally-stored certificate and add the suffix .cert (as file type).
To import the certificate into the SSL Client PSE of the Integration Server system, perform the following steps:

1. Call the Trust Manager (transaction STRUST) in the Integration Server system.
2. Double-click the node SSL Client (Standard).
3. In the Certificate frame, choose Import Certificate  
4. Choose the file format Base 64.
5. In the File Path field, select the certificate that was previously exported to your hard disc.
6. Choose Input .
7. Choose Add to Certificate List.
As a result, the server certificate of the receiver system is imported into the client PSE of the Integration Server.

**Making Sure That the Receiver Trusts the IS Client Certificate**

When the IS calls the receiver, authentication of the IS is done on the basis of a certificate signed by a certification authority. To define the relevant part of the trust relationship, the certificate of the certification authority has to be imported into the SSL Server PSE of the receiver.

Therefore, the certificate of the certification authority has to be added to the certificate list within the Personal Security Environment (PSE) of the receiver back-end (in this case, into the SSL Server PSE of the receiver). This step is necessary to make this certification authority also known to the receiver.

Perform the following steps:

1. In the system of the IS, call the Trust Manager (transaction STRUST).
2. Double-click the node SSL Client (Standard).
3. Under Own Certificate double click the certificate (field Owner).
4. Under Certificate choose Export Certificate and save the certificate on your hard disc.
To import the certificate into the SSL Client PSE of the Integration Server system, perform the following steps:

1. Call the Trust Manager (transaction STRUST) in the receiver system.
2. Double-click the SSL Client (Standard) node.
3. In the frame Certificate, choose Import Certificate ( ).
4. Choose the file format Base 64.
5. In the field File Path select the certificate that was previously exported to your hard disc.
6. Choose Input ( ).
7. Choose Add to Certificate List.
4.1.2.4 Defining User Mappings

Defining User Mapping – IS

No service user is assigned yet to the certificate sent with the sender's message. However, a service user is necessary to call the Integration Server as WS provider. Therefore, the certificate sent with the message has to be mapped to a service user.

To maintain the user mapping, perform the following steps:

2. In the field Table/View enter VUSREXTID.
3. Choose Maintain.
4. In the External ID type field select the type DN (Work Area field).

5. In transaction Change View “Assignment of External ID to Users”: Overview under Assignment of External ID to Users, activate the exported certificate from the WS Consumer system (checkbox).

Alternatively, you can do the following:

a. In transaction Change View “Assignment of External ID to Users”: Overview, select New Entries.

b. In the field External ID, enter the name of the sender's certificate:

For example: CN=X7AClient, OU=10120021077, O=SAP-AG, C=DE.

c. Specify a user for the WS call.
Select user PIAPPLUSER, XIAPPLUSER, or a user which has the permissions to execute the Web service.

6. Perform the steps described under Configuring the System for Using X.509 Client Certificates.
You have the option to test the connection from the WS consumer (sender) to the Integration Server by creating a HTTP connection in transaction SM59.

**Defining User Mapping - Receiver**

A service user is not yet assigned to the certificate within the message sent from the IS to the receiver. A service user is necessary to call the receiver system as WS provider. The user assignment is done in table VUSREXTID.

Perform the steps described under **Configuring the System for Using X.509 Client Certificates**.

In the *External ID* field, enter the name of the Integration Server certificate.

### 4.1.3 Configuring WS Provider

Follow the steps below to configure the ABAP back-end as a WS provider. The relevant communication path is illustrated in the following figure:

![Diagram](image)

Proceed as described under 2.2.2.1.

Since the back-end is based on AS ABAP 7.0 < SP14, use the transaction *WSCONFIG* to define the endpoint.

To configure the necessary security settings, perform the following additional steps:

In transaction *WSCONFIG*, go to the *Logon Data* tab,

- Select the following entry with the dropdown list box in the *Procedure* field: *Required with Client Certificate (SSL)*
- Enter the client of the WS provider

### 4.1.4 Configuring Message Processing

To configure message processing on the Integration Server, you have to execute tasks in the System Landscape Directory and the Integration Directory. This configuration task is mainly standard process integration configuration. Because of this, the chapter is kept short.

More information:

Section 2.2.3

**Tasks in the System Landscape Directory**

**Integration Directory**

The part of the communication that is the subject of this section is illustrated in the figure below.
4.1.4.1 Defining System Landscape in the System Landscape Directory

Firstly you have to define the system landscape in the System Landscape Directory (SLD).

The system landscape consists of the Integration Server system and the back-end systems involved.

1. To call the System Landscape Directory, choose Process Integration → Start Integration Builder in the user menu.
2. On the initial screen of SAP NetWeaver Process Integration, choose the System Landscape Directory link.
3. Log on using your user and password.

You have to define technical systems for the sender and receiver for this scenario. The Integration Server was already registered as a technical system during the installation of SAP NetWeaver Process Integration. AS ABAP technical systems are also registered in the SLD server automatically. If automatic registration is not possible, then you have to create the new AS ABAP technical systems manually.

Additionally, you have to define business systems for the following entities:

- One business system for the Integration Server
- One business system for the sender and one business system for the receiver back-ends

Classify the back-end systems as Application Systems.

Since the back-ends are based on AS ABAP, select the system type AS ABAP.

More information:
Configuring Business Systems in the SLD

4.1.4.2 Defining Communication Components

Use the following tasks to define the business systems involved as communication components in the Integration Directory in order to make these systems addressable for messaging.

You perform the following tasks in the Integration Directory.

1. To call the Integration Directory, choose Process Integration → Start Integration Builder in the user menu.
2. On the initial screen, choose the Integration Builder link (under Integration Directory).
3. Log on using your user and password.
4. To address the back-end systems as the sender and receiver of messages, you have to define communication components for each back-end system involved in the communication.

Define a communication component of type business system for the sender and another communication component for the receiver:

4.1.4.3 Configuring Logical Routing

With the following tasks you define the logical routing to specify the message flow at runtime.

You have to define a receiver determination to route the message sent by the sender back-end to the receiver back-end. Additionally, you have to define an interface determination to specify the inbound interface at the receiver side.

More information: Defining Logical Routing
Create a receiver determination and enter the following objects in the key of the receiver determination:

- Interface Name and Namespace: Select the outbound interface used at the sender side to perform the Web service call
- Communication Component (sender): Select the previously defined communication component for the sender business system.

When editing the receiver determination, add the communication component for the receiver system under *Configured Receivers*.

Create an interface determination and enter the following objects in the key of the interface determination:

- Interface Name and Namespace: Select the outbound interface used at the sender side to perform the Web service call
- Communication Component (sender): Select the previously defined communication component for the sender business system.
- Communication Component (receiver): Select the previously defined communication component for the receiver business system.

When editing the interface determination, under *Receiver Interfaces* (column *Name*) select the inbound interface that is used at the receiver back-end side to perform the WS call.

### 4.1.4.4 Configuring Outbound Processing

To configure outbound processing of the message, you have to define:

- A receiver communication channel
- A receiver agreement and assign the receiver channel to the sender agreement

More information:

- [Configuring Communication Channels with Adapter Type WS](#)
- [Defining a Receiver Agreement](#)

When creating the communication channel, do the following:

- Create a communication channel for the communication component of the receiver back-end.

When editing the communication channel, do the following:

- Select *Receiver* as the direction of the communication.
- Select the Adapter Type *WS*.
- Under *Security Settings* select the following Authentication Method:
  - X.509 SSL Client Certificate (Authentication Using HTTPS)
- Select the Transport Protocol *HTTPS*.
- Specify the following attributes:
  - **Target Host**: Enter the host name of the receiver back-end system.
  - **Service Name/Port**: Enter port of the receiver back-end system.
o HTTPS Service Name/Port: Enter HTTPS port of the receiver back-end system.

o Select checkbox Access Path Points to a WSDL.

When creating the receiver agreement, do the following:

- In the object key, in the receiver fields select the communication component for the receiver back-end and the inbound interface of the WS provider.

When editing the receiver agreement, note the following:

- Assign the receiver communication channel defined before.
- Enter the Adapter-Specific Attributes of the receiver agreement.

When entering the attributes, note the following:

In the field SSL Client PSE, enter name of the client PSE of the receiver back-end.

To find out the name of the SSL Client PSE, call transaction STRUST in the Integration Server system and double-click the node SSL Client (Standard).


### 4.1.4.5 Configuring Inbound Processing

To configure inbound processing of the message, you have to

- Define a sender communication channel
- Define a sender agreement and assign the sender channel to the sender agreement

More information:

- Configuring Communication Channels with Adapter Type WS
- Defining Sender Agreements

When creating the communication channel, note that you have to:

- Create a communication channel for the communication component of the sender back-end.

When editing the communication channel, note the following:

- Select Sender as the direction of the communication.
- Select the Adapter Type WS.
- Under Security Settings select the following Authentication Method:
  X.509 SSL Client Certificate (Authentication Using HTTPS)
- Select the Transport Protocol HTTPS.
- Check the Secure Conversation checkbox.
- Specify the following attributes:
  - Target Host: Enter target host of the Integration Server (for example pwdf1234).
  - Service Name/Port: Enter port of the Integration Server.
  - HTTPS Service Name/Port: Enter HTTPS port of the Integration Server.
For more information on how to find out these values, see 2.2.4.1.

When creating the sender agreement, note the following: In the object key, for the sender select the communication component for the sender back-end and the outbound interface of the WS consumer.

When editing the sender agreement, note the following: Assign the sender communication channel previously created.

### 4.1.4.6 Activating Configuration Objects

Activate all created configuration objects in the Integration Directory.

### 4.1.5 Configuring WS Consumer

To configure the sender back-end as WS consumer, the following steps have to be performed.

The relevant part of the communication flow is illustrated in the figure below.

Since the sender back-end is based on AS ABAP 7.0 (<SP14), perform the necessary configuration tasks with transaction `LPCONFIG`.

Proceed as described under 2.2.4.3.

Since the back-end is based on AS ABAP 7.0 < SP14, use the transaction `SM59` to create an RFC destination to the Integration Server and then transaction `LPCONFIG` to create a logical port.

To configure the necessary security settings, perform the following additional steps:

In transaction `SM59`,

- Choose tab `Logon & Security` and under `Status of Secure Protocol`, select the `Active` checkbox for SSL.
- In tab `Logon & Security` in the `Client Certificate` field, select the certificate `DFAUTH SSL Client (Standard)` with the dropdown list box (see section 4.1.2.2 → 2. Make Sure That Integration Server Trusts Client Certificate in the Message of the Sender).