

webMethods EntireX

IDL Extractor for Integration Server

Version 10.3

October 2018

This document applies to webMethods EntireX Version 10.3 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1997-2018 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: EXX-EEXXAIETRACTOR-103-20191129

Table of Contents

Software AG IDL Extractor for webMethods Integration Server	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
I Introduction to the IDL Extractor for Integration Server	5
2 Using the IDL Extractor for Integration Server	9
Step 1: Start the IDL Extractor for Integration Server	10
Step 2a: Connect Designer with Integration Server	11
Step 2b: Connect Designer with Integration Server using an Existing Connection	12
Step 3: Select the Integration Server Package to Extract	14
Step 4: Create or Update an Adapter Connection	17
Step 5: Define a Listener	19
3 Mapping Integration Server Data Types to IDL	21
4 Using the Service <code>pub.wmentirex.listener:generateIDLfromService</code>	25
II Integration Server Preferences	27
5 Integration Server Preferences	29
Integration Server Connections	30
Setting Integration Server Preferences	32

Software AG IDL Extractor for webMethods Integration Server

The Software AG IDL Extractor for webMethods Integration Server is a wizard that reads a package from the Integration Server and generates a Software AG IDL file from all the existing services and nodes. Each service results in a program in the IDL file. All parameters of the services are mapped to an IDL alphanumeric data type, available as variable (AV) or fixed (An) length.

Introduction	Introduction to the IDL Extractor for Integration Server.
Using	Using the IDL Extractor for Integration Server.
IS to IDL Mapping	Mapping Integration Server data types and syntax to Software AG IDL.
Automating IDL Generation	Creating an IDL file can be automated using service <code>pub.wmentirex.listener.generateIDLfromService</code> .
Preferences	Describes the Integration Server preferences.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.asp and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

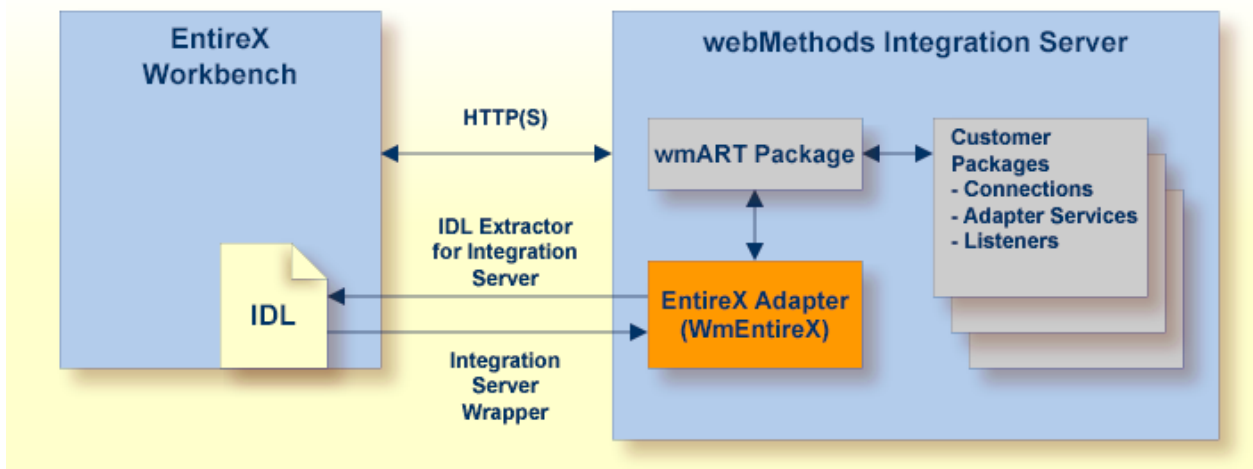
Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

I Introduction to the IDL Extractor for Integration Server

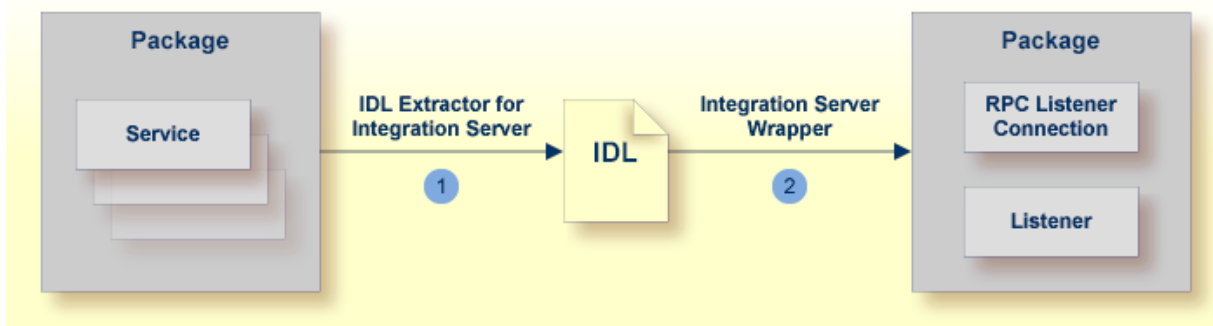
Scope

The Software AG IDL Extractor for webMethods Integration Server is a wizard that reads a package from the Integration Server and generates a Software AG IDL file from all the existing services and nodes. Each service results in a program in the IDL file. All parameters of the services are mapped to an IDL alphanumeric data type, available as variable (AV) or fixed (An) length.



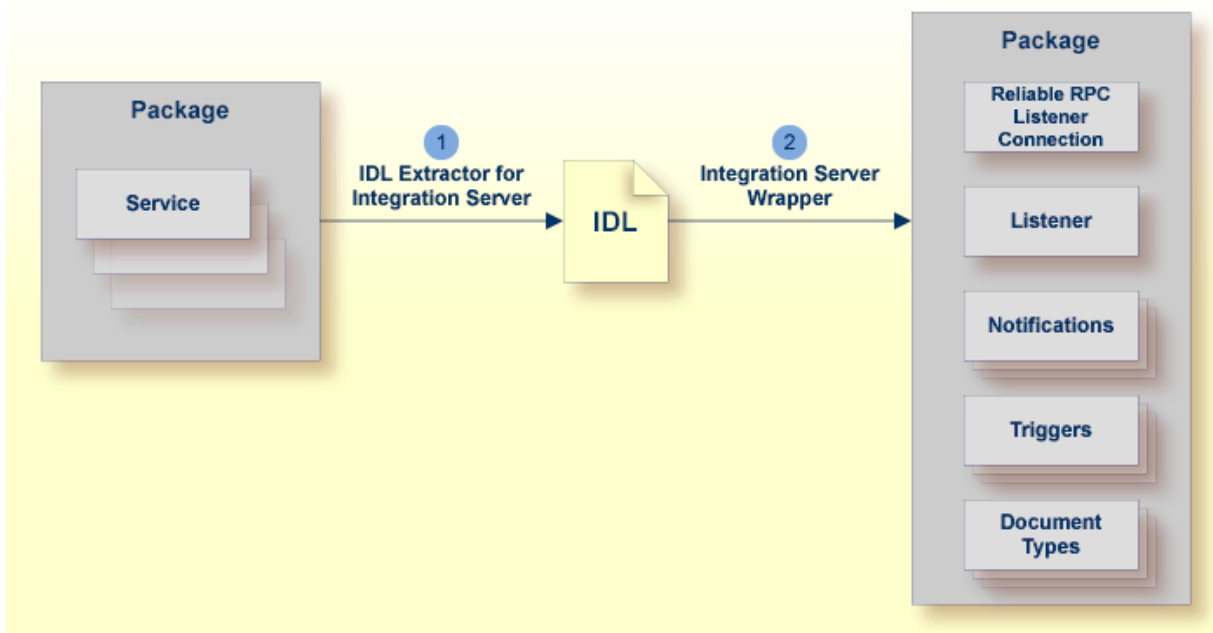
With the Integration Server Wrapper you can generate a server of type "RPC Listener" or "Reliable RPC Listener". See *Using the Integration Server Wrapper*.

■ RPC Listener



1. See [Using the IDL Extractor for Integration Server](#).
2. See *Step 4c: Create a Connection and Related Adapter Listener* in the Integration Server Wrapper documentation.

■ **Reliable RPC Listener**



1. See [Using the IDL Extractor for Integration Server](#).
2. See *Step 4c: Create a Connection and Related Adapter Listener* in the Integration Server Wrapper documentation.

Calling an Integration Server Flow from a Mainframe Program

➤ **To call an Integration Server flow from a mainframe program**

- Use the plug-in to generate the connection and see *Using the Integration Server Wrapper* with the steps 1 to 3, 4c and 5.


2 Using the IDL Extractor for Integration Server

▪ Step 1: Start the IDL Extractor for Integration Server	10
▪ Step 2a: Connect Designer with Integration Server	11
▪ Step 2b: Connect Designer with Integration Server using an Existing Connection	12
▪ Step 3: Select the Integration Server Package to Extract	14
▪ Step 4: Create or Update an Adapter Connection	17
▪ Step 5: Define a Listener	19

Step 1: Start the IDL Extractor for Integration Server

➤ To start the IDL Extractor for Integration Server

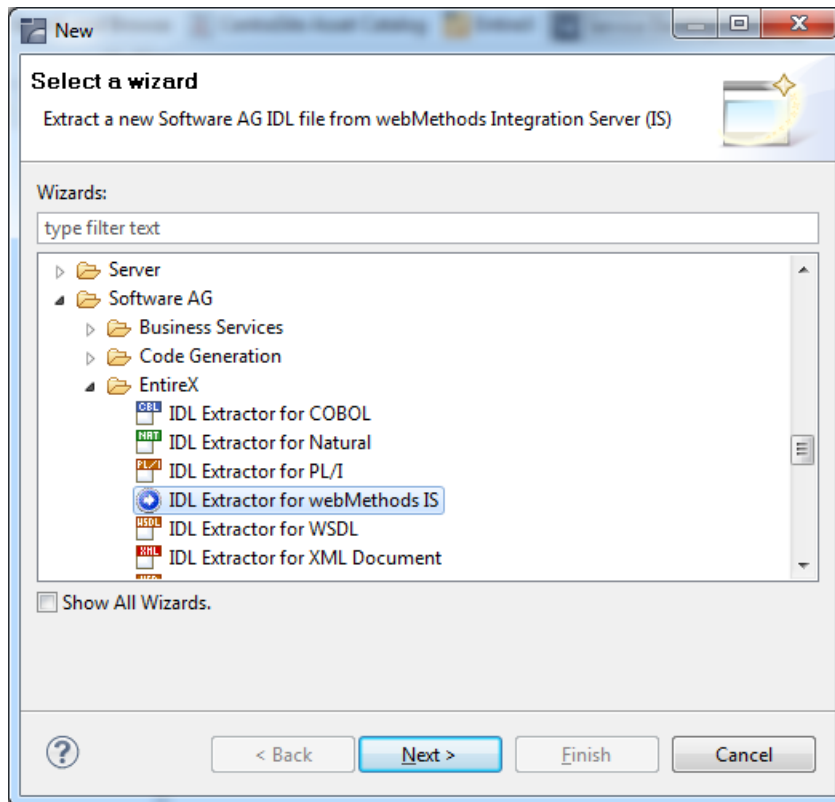
- 1 The IDL Extractor is a *New Wizard* in Eclipse. Choose **New** from the file menu, select **IDL Extractor for webMethods IS** in the following page and choose **Next**.

 **Note:** You can also choose **New** from the toolbar or context menu of a view showing resources. Also **Ctrl-N** starts the selection of the *New Wizards*.

- 2 If you are using the wizard for the first time without any predefined Integration Server connections, continue with [Step 2a: Connect Designer with Integration Server](#).

Or:

If Integration Server connections are already defined (see *Integration Server Connections*), or you want to communicate with an additional Integration Server, continue with [Step 2b: Connect Designer with Integration Server using an Existing Connection](#).



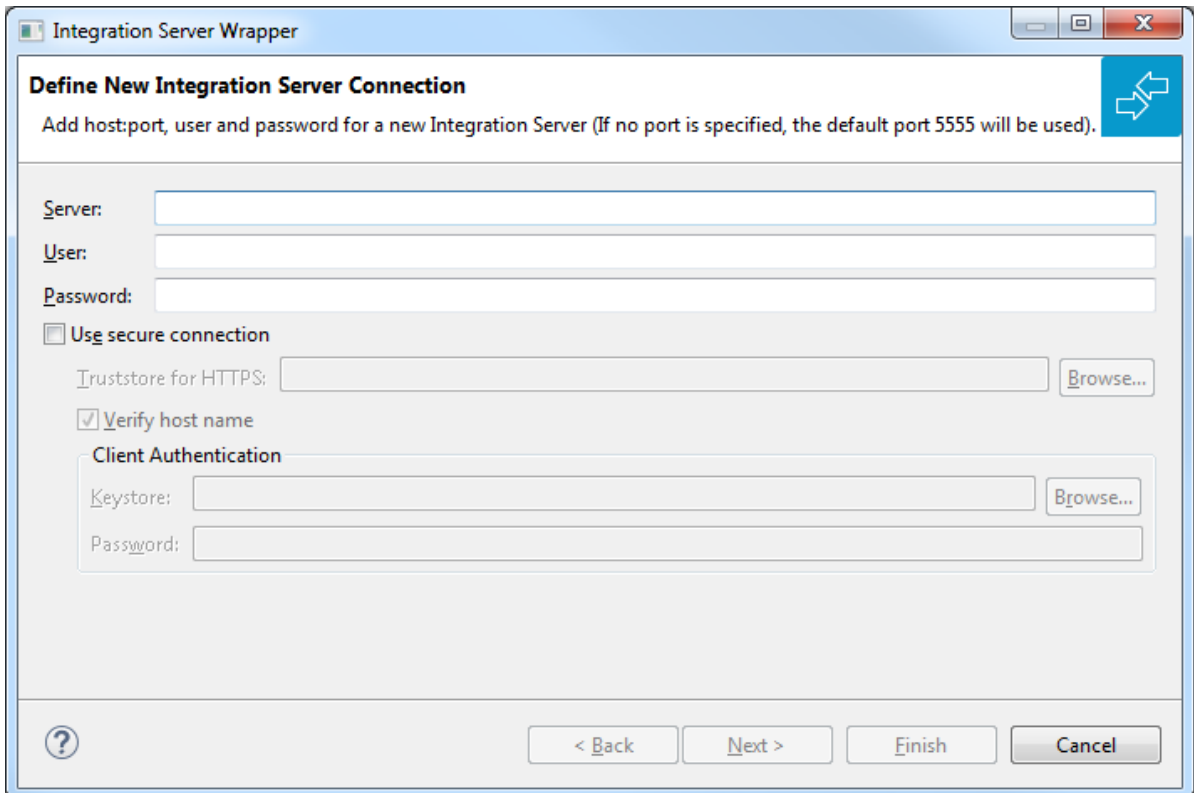
Step 2a: Connect Designer with Integration Server

> To create a new Integration Server connection

- 1 Define the new Integration Server connection on the following wizard page.

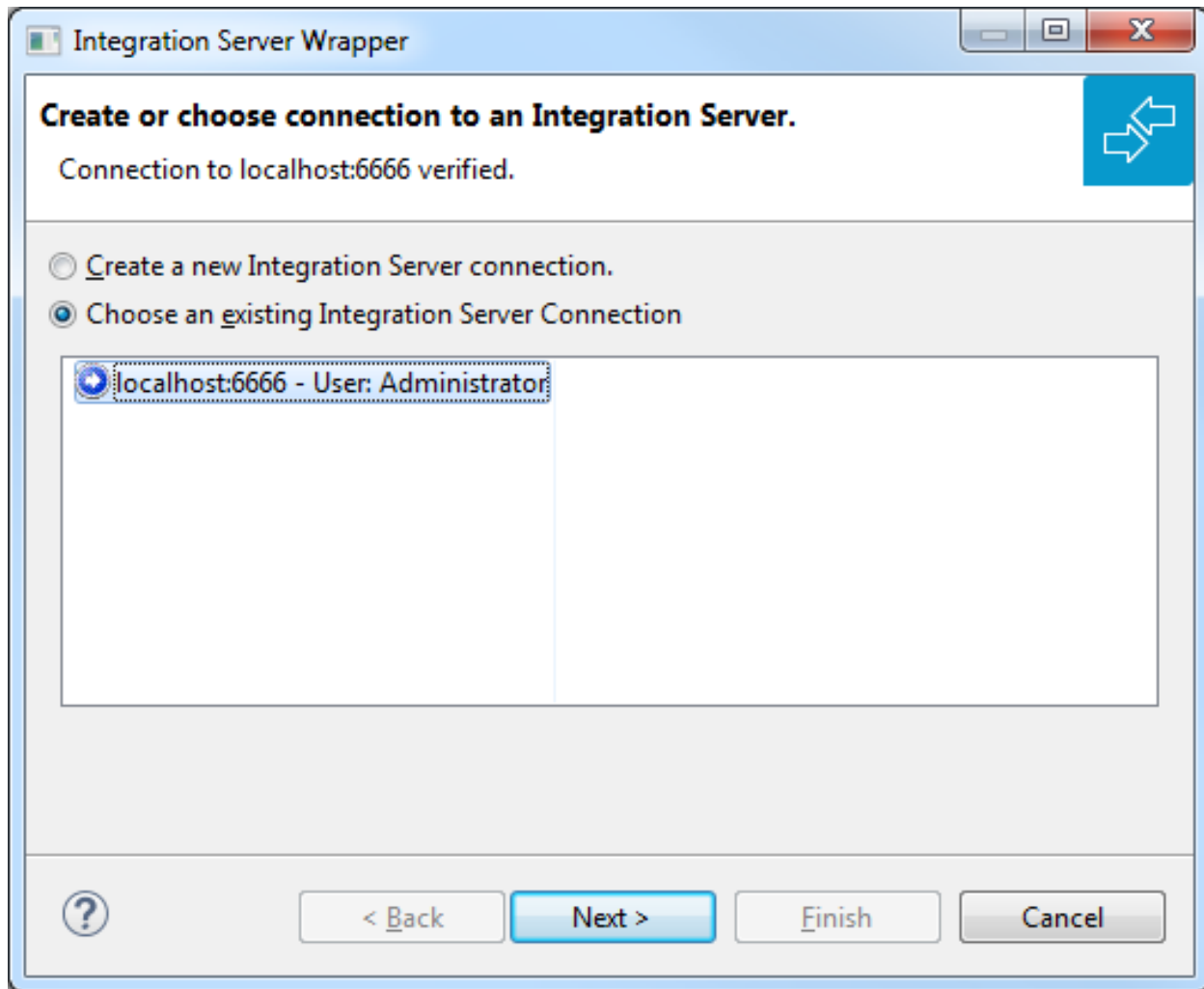
 **Notes:**

1. The only required field is **Server**. Enter the hostname of the Integration Server including an optional port number. If no port number is specified, port number defaults to "5555". The **Integration Server Authentication** can be passed with the **User** and **Password** fields.
2. Optional settings are for secure connections. The **Truststore for HTTPS** contains all signed certificates and must be a valid truststore.
3. The check box **Verify host name** checks that the hostname is entered in the stored certificate.
4. When the Integration Server has **Client Authentication** enabled, you can specify your **Keystore** file and keystore **Password**.
5. For managing Integration Server connections, see [Preferences](#).



- 2 Choose **Next** and continue with [Step 3: Select the Integration Server Package to Extract](#).

Step 2b: Connect Designer with Integration Server using an Existing Connection



> To use an existing Integration Server connection

- 1 Select **Choose an existing Integration Server Connection** and an Integration Server connection from the list.

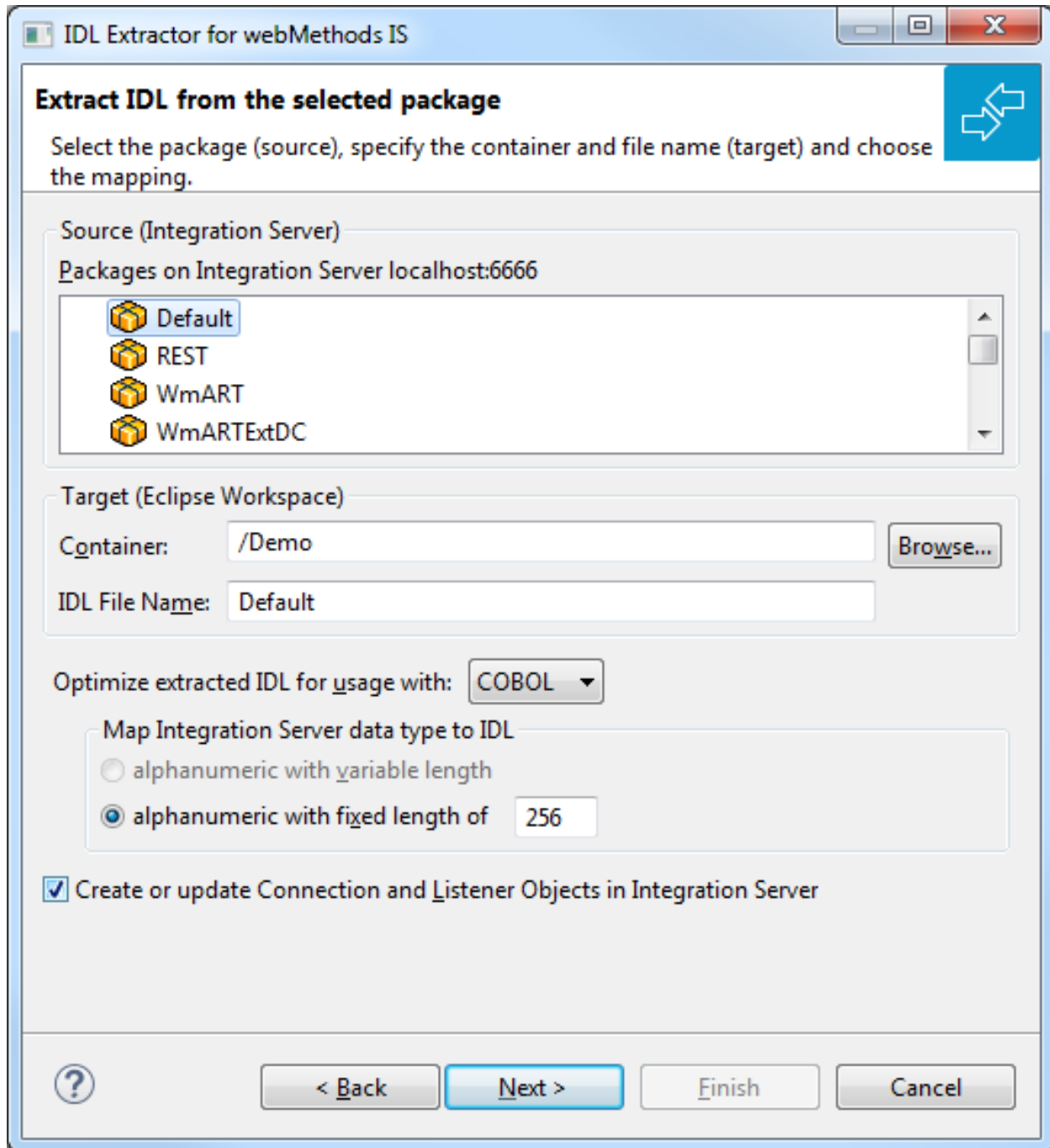
The selected connection is verified by a ping command. If the response is valid, the **Next** button is enabled. If invalid, an error message is displayed.

- 2 Continue with *Step 3: Select the Integration Server Package to Extract.*

➤ **To create an additional Integration Server connection**

- Select **Create a new Integration Server Connection** and continue with *Step 2a: Connect Designer with Integration Server*.

Step 3: Select the Integration Server Package to Extract



➤ **To extract the IDL from the selected package**

- 1 Select the package to extract (from the list indicated by **Source**), for example the package "Default" in the screen above.
- 2 Specify the target. By default, the wizard tries to find a valid container based on your position in the Navigator or Package Explorer View.

Or:

Choose **Browse...** to select a container from your workspace.



Notes:

1. The IDL file name is based on the selected Integration Server package from the list below, but this is only a proposal and can be changed.
 2. If the file name already exists in your container, a warning will be displayed in the title area of the wizard page.
 3. If the extension "idl" is not specified, it will be added automatically.
- 3 Optimize the extracted IDL. Use the combo-box and select the target programming language:

■ **COBOL**

For usage with the COBOL Wrapper. Enter the *default length* for IDL type *A(default-length)* fields which map then to COBOL alphanumeric data items PIC *X(default-length)*.

■ **Natural**

For usage with the Natural Wrapper. Choose one option indicated by **Map Integration Server data type to IDL:**

- alphanumeric with variable length: Extract to IDL Type *AV* fields which map then to Natural *DYNAMIC* data types. The interface does not have any length restrictions.
- alphanumeric with fixed length: Extract to IDL Type *A(default-length)* fields which map then to Natural fixed-length alphanumeric data types. The interface from a Natural point of view is more legacy-like and easier to program, but there are length restrictions.

■ **PL/I**

For usage with the PL/I Wrapper. Enter the *default length* for IDL Type *A(default-length)* fields which map then to PL/I fixed-length alphanumeric fields *CHAR(default-length)*.

■ **Other**

For usage with other *EntireX Wrappers*. If you later use wrappers other than the COBOL Wrapper, Natural Wrapper or PL/I Wrapper, no options need to be specified.

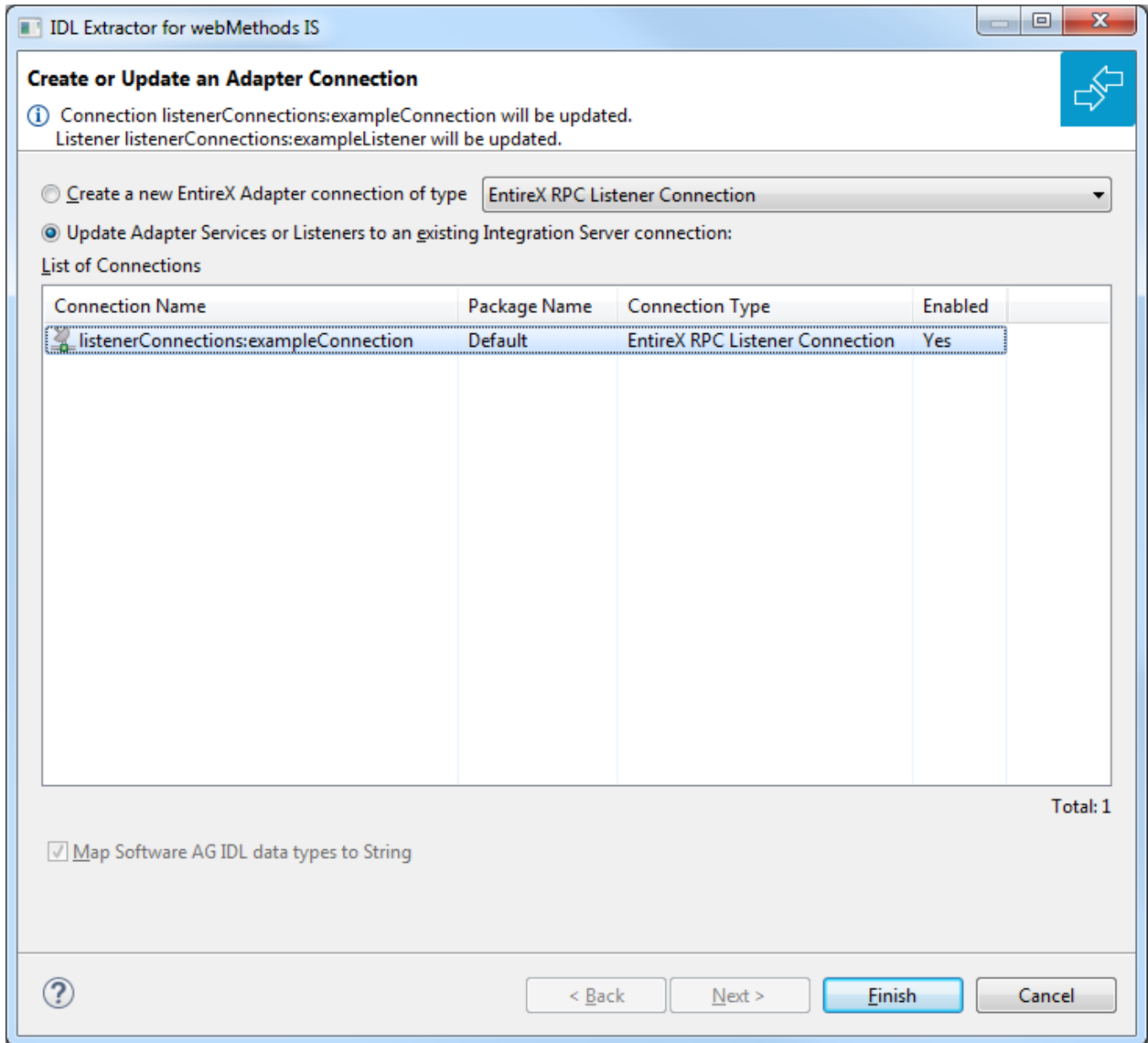
- 4 Clear **Create Listener Objects in Integration Server** to only extract the IDL file from the package. The wizard finishes after extracting the IDL.



Tip: This is useful if you want to modify the IDL (manually) before generating clients. In this case, the connections and listeners have to be created in a separate step. See *Step 4c: Create a Connection and Related Adapter Listener* in the Integration Server Wrapper documentation.

- 5 Choose **Next** to store the IDL file in the selected container in the Eclipse workspace. When you go to the next page, the **Back** button is disabled, because the IDL file has already been created and this step cannot be reverted.

Step 4: Create or Update an Adapter Connection



As a prerequisite, the IDL file has been stored in the workspace and the **Back** button is now disabled.

> To create a new connection

- Select a **Connection type** and click **Next**.

Notes:

1. An EntireX RPC Listener connection is always available.

2. An EntireX Reliable RPC Listener connection is available if all IDL programs contain only `IN` parameters.
3. An EntireX Direct RPC Listener connection is available if it is enabled by the license for the webMethods EntireX Adapter for Integration Server.

➤ **To update an existing connection**

- Select a connection from the list of available listener connections and click **Next**.




Notes:

1. An EntireX RPC Listener connection is always available.
2. An EntireX Reliable RPC Listener connection is available if all IDL programs contain only `IN` parameters.
3. An EntireX Direct RPC Listener connection is available if it is enabled by the license for the webMethods EntireX Adapter for Integration Server.

Step 5: Define a Listener

> **To define a listener**

- 1 Select an Integration Server package where the listener will be stored.
- 2 Specify the names for **Folder Name** (default: *library name*), **Connection Name** (default: *library nameConnection*) and **Listener Name** (default: *library nameListener*).
- 3 If necessary, edit the Broker settings for the Listener (**Broker ID**, **Server Address**, **User ID**, **Password**, and **Encoding**).
- 4  **Note:** When creating a connection, a package dependency is added such that the selected package depends on webMethods EntireX (the package `WmEntireX`) with the version currently used.
- 5 Choose **Finish**.

As a result, the following objects will be created:

- one connection of type EntireX RPC Listener, EntireX Reliable RPC Listener or EntireX Direct RPC Listener
- one listener object

For a connection of type **EntireX Reliable RPC Listener** the following objects will be created in addition for each IDL program:

- one notification object
- one trigger object
- one document type object

3 Mapping Integration Server Data Types to IDL

The signature of an Integration Server service specifies the data types for the parameters of the service. There are three data types: `String`, `Record` and `Object`. Parameters of type `String` are mapped to an IDL alphanumeric data type, parameters of type `Record` to IDL groups. Parameters of type `Object` cannot be mapped to an IDL data type and thus result in an error. If such an error occurs, the IDL program which contains the invalid data type mapping is written to the IDL file, however, all lines are comments. Starting with version 9.12 Fix 1 of the EntireX Adapter a subset of the Java Wrapper types are mapped to IDL types. This subset has been chosen so it is compatible to the JSON data type mapping in Integration Server.

Integration Server Object Type Java Wrapper Types	Software AG IDL Data Type		
	Adapter Version 10.3 or higher	Adapter Version 9.12 Fix 1 or higher	Adapter Version 9.12 or lower
<code>java.lang.String</code>	See table below	See table below	See table below
<code>java.lang.Long</code>	N19	N19	error
<code>java.lang.Integer</code>	I4	I4	error
<code>java.lang.Float</code>	F4	F4	error
<code>java.lang.Double</code>	F8	F8	error
<code>java.lang.Boolean</code>	L	L	error
<code>java.util.Date</code>	Same as <code>String</code> , see table below	error	error
<code>byte[]</code>	BV, BV256 (depending on target language)	BV, BV256 (depending on target language)	error
All other wrapper types or record without subcomponents	Input parameter: ignored; output parameter: same as <code>String</code> , see table below	error	error
No wrapper type	BV, BV256 if extracting from EntireX Adapter Service ⁽¹⁾ ; otherwise see above	error; or BV, BV256 if extracting from EntireX Adapter Service ⁽¹⁾	error; or BV, BV256 if extracting from EntireX Adapter Service ⁽¹⁾

⁽¹⁾ A parameter of an EntireX Adapter service of type `Object` is mapped to an IDL binary data type. Depending on the target language, it is either mapped to `BV` (for `Natural` and `Other`) or to `BV256` (for `COBOL` and `PL/I`).

All parameters of type `String` are mapped to an IDL alphanumeric data type, available as variable (`AV`, `AVn`) or fixed (`An`) length. Which alphanumeric type is used is specified in the wizard page of the IDL Extractor for Integration Server or as a parameter of the generation service.

Parameters of type `String` may have an associated context type which is specified in the constraints of the parameter's properties. This content type is only used if all parameters are of type `String`. This content type may influence the data type mapping. The following table shows the mapping.

Integration Server Content Type	Software AG IDL Data Type	
<code>boolean</code>	<code>L</code>	Logical
<code>decimal</code>	<code>Nx</code> , <code>Nx.y</code> or <code>An</code> , <code>AV</code> , <code>AVn</code>	Unpacked decimal or Alphanumeric ⁽¹⁾
<code>float</code>	<code>F4</code>	Floating point (small)
<code>double</code>	<code>F8</code>	Floating point (large)
<code>Date</code>	<code>An</code> , <code>AV</code> , <code>AVn</code>	Alphanumeric ⁽²⁾
<code>dateTime</code>	<code>An</code> , <code>AV</code> , <code>AVn</code>	Alphanumeric ⁽²⁾
<code>base64Binary</code>	<code>An</code> , <code>AV</code> , <code>AVn</code>	Alphanumeric ⁽²⁾
<code>byte</code>	<code>I1</code>	Integer (small)
<code>unsignedByte</code>	<code>NU3</code>	Unpacked decimal unsigned
<code>short</code>	<code>I2</code>	Integer (medium)
<code>unsignedShort</code>	<code>NU5</code>	Unpacked decimal unsigned
<code>int</code>	<code>I4</code>	Integer (large)
<code>unsignedInt</code>	<code>NU10</code>	Unpacked decimal unsigned
<code>long</code>	<code>N19</code>	Unpacked decimal
<code>unsignedLong</code>	<code>NU20</code>	Unpacked decimal unsigned
<code>integer</code>	<code>N29</code>	Unpacked decimal
<code>positiveInteger</code>	<code>NU29</code>	Unpacked decimal unsigned
<code>nonPositiveInteger</code>	<code>N29</code>	Unpacked decimal
<code>negativeInteger</code>	<code>N29</code>	Unpacked decimal
<code>nonNegativeInteger</code>	<code>NU29</code>	Unpacked decimal unsigned
<code>string</code>	<code>An</code> , <code>AV</code> , <code>AVn</code>	Alphanumeric ⁽³⁾
<code>all others</code>	<code>An</code> , <code>AV</code> , <code>AVn</code>	Alphanumeric ⁽²⁾



Notes:

1. If the content type specifies a total number of digits and a number of fraction digits, then `Nx.y` is used. If only a total number of digits is specified, then `Nx` is used. If no total number of digits

is specified, or the total number exceeds the maximum allowed value, or the number of fraction digits exceeds the maximum allowed value, then a string according to ⁽²⁾ is used. The maximum allowed value for the total number of digits is 29, for COBOL it is 31. The maximum allowed value for the number of fraction digits is 7, for COBOL it is 31 and for Natural it is 29.

2. Depending on what is specified in the wizard page of the IDL Extractor for Integration Server or as a parameter of the generation service, the IDL data type `AV`, `AVn` or `An` is used.
3. If the content type specifies a length `len`, then `Alen` is used. If the content type specifies a maximum length `max`, then `AVmax` or `Amax` is used, see ⁽²⁾ for the choice between `A` versus `Av`. If neither length nor maximum length is specified, the choice is the same as in ⁽²⁾.

4 Using the Service

pub.wmentirex.listener:generateIDLfromService

Service `pub.wmentirex.listener:generateIDLfromService` generates an IDL file for a given list of Integration Server services. For each service, a program in the IDL file is created. If the IDL file already exists, it is extended with the newly created program definitions. Duplicate definitions for the same program are possible. In this case, the IDL file has to be corrected manually. Note that this service creates/updates only the IDL file. Creating or updating a listener with the definitions of the IDL file has to be done with *webMethods Integration Server Wrapper* of the Designer. The following parameters are relevant. Parameter direction can be In or Out:

Parameter	I/O	Description
<code>serviceName String</code>	I	The full name of the Integration Server service (e.g. <code>folder1.folder2:service</code>). Multiple services can be specified using a semicolon (;) as delimiter.
<code>fileName String</code>	I	The name of the IDL file (with extension ".idl"). You cannot specify a folder name. The file is created in the folder <code><IntegrationServer_instance>/packages/WmEntireX/resources</code> .
<code>libraryName String</code>	I	The library name to be used in the IDL file. If the IDL file already exists, this parameter is ignored.
<code>stringType String</code>	I	The IDL data type which is used for string data types. Possible values are AV, AVn, or An. This parameter is optional, default is AV.
<code>language String</code>	I	The target language for the RPC clients. Possible values are COBOL, Natural, PL/I, or Other.
<code>result String</code>	O	Either a success message if the generation was successful (which includes the full pathname of the IDL file) or an error message.

II

Integration Server Preferences

5 Integration Server Preferences

- Integration Server Connections 30
- Setting Integration Server Preferences 32

The Integration Server preferences are used to manage Integration Server connections. This chapter applies both to the Integration Server Wrapper and the IDL Extractor for Integration Server.

Integration Server Connections

The Integration Server connections are responsible for the HTTP/HTTPS communication to the Integration Server. They are used in the wizards described in *Using the Integration Server Wrapper* and *Using the IDL Extractor for Integration Server* and are managed in the Integration Server preferences.

An Integration Server connection contains the following information:

- Server name (required, consists of hostname and optional port number, where the default port number is 5555)
- User name
- Password
- optional parameters for SSL (HTTPS):
 - Truststore (name of the file)
 - Verify hostname
 - Optional parameters for client verification:
 - Keystore (name of the file)
 - Password for the Keystore

This information can be specified in the following dialog:

Integration Server Connection
Edit an Integration Server Connection

Server: localhost:6666

User: Administrator

Password: ●●●●●●●●

Use secure connection

Truststore for HTTPS: C:\Program Files\Software AG\EntireX\Etc\ExxJavaAppCert.jks

Verify host name

Client Authentication

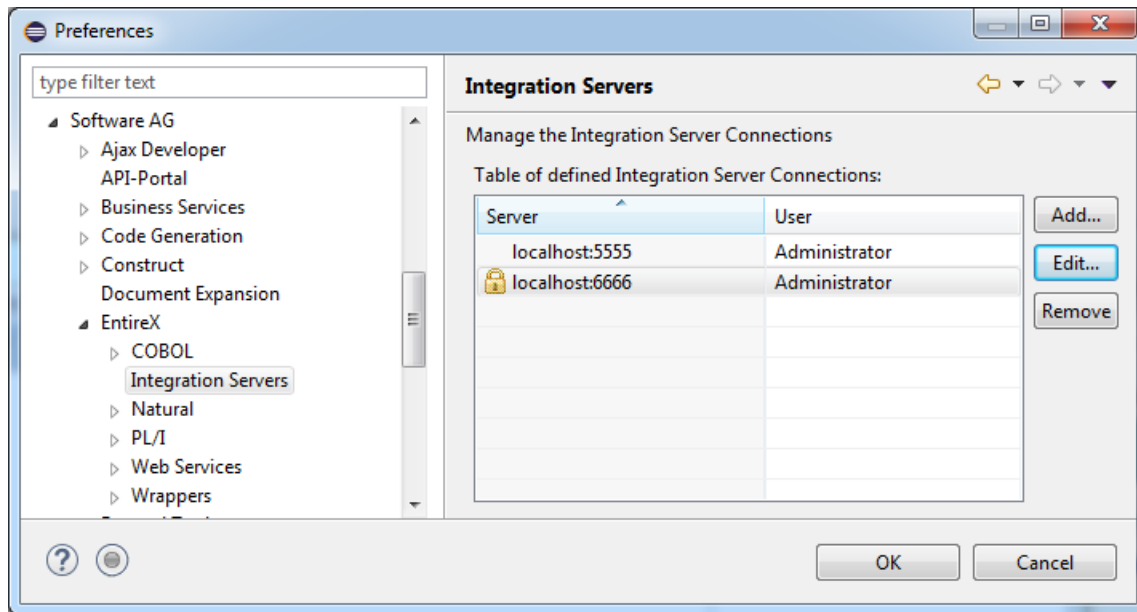
Keystore:

Password:

 **Notes:**

1. The only required field is **Server**. Enter the hostname of the Integration Server including an optional port number. If no port number is specified, port number defaults to "5555". The **Integration Server Authentication** can be passed with the **User** and **Password** fields.
2. Optional settings are for secure connections. The **Truststore for HTTPS** contains all signed certificates and must be a valid truststore.
3. The check box **Verify host name** checks that the hostname is entered in the stored certificate.
4. When the Integration Server has **Client Authentication** enabled, you can specify your **Keystore** file and keystore **Password**.
5. For managing Integration Server connections, see [Preferences](#).


Setting Integration Server Preferences



> To add, edit or remove Integration Server connections

- Open the Preferences page and choose **Add...**, **Edit...** or **Remove**.

All changes in the table will be stored permanently after leaving the preferences with **OK**.

- 
Caution: The creation of duplicates is forbidden. A duplicate will be detected if server (including port number, for instance the default "5555"), user and truststore file have the same name.