webMethods EntireX

webMethods Integration Server Wrapper

Version 10.5

October 2019

WEBMETHODS

This document applies to webMethods EntireX Version 10.5 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 © 1997-2019 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-EEXXXAIWRAPPER-105-20220422

Table of Contents

1 About this Documentation 1
- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

2 Introduction to the Integration Server Wrapper 5
- Scope 6
- Prerequisites 11

3 Using the Integration Server Wrapper 13
- Step 1: Start the Integration Server Wrapper Wizard 14
- Step 2a: Create a New Integration Server Connection 16
- Step 2b: Use an Existing Integration Server Connection 17
- Step 3: Create or Update an Adapter Connection 19
- Step 4a: Create a Connection and Related Adapter Services 21
- Step 4b: Create or Update a REST Resource 22
- Step 4c: Create a Connection and Related Adapter Listener 24
- Step 5: Finish the Wizard 26

4 Software AG IDL to Integration Server Mapping 27
- Mapping IDL Data Types to IS Data Types 28
- Mapping Groups 30
- Mapping Arrays 30

5 Integration Server Preferences 31
- Integration Server Connections 32
- Setting Integration Server Preferences 34

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 *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	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

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

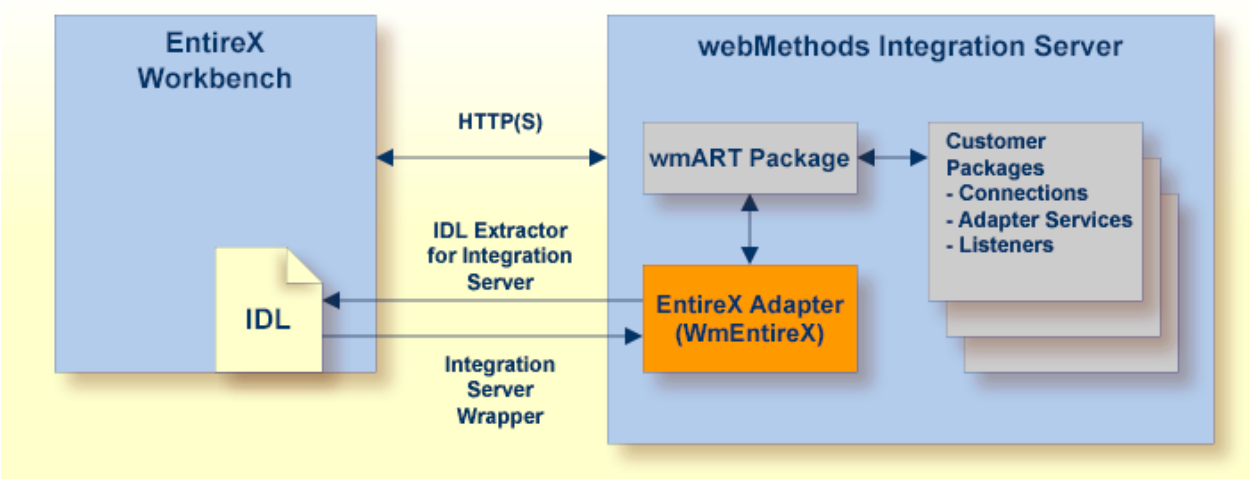
Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction to the Integration Server Wrapper

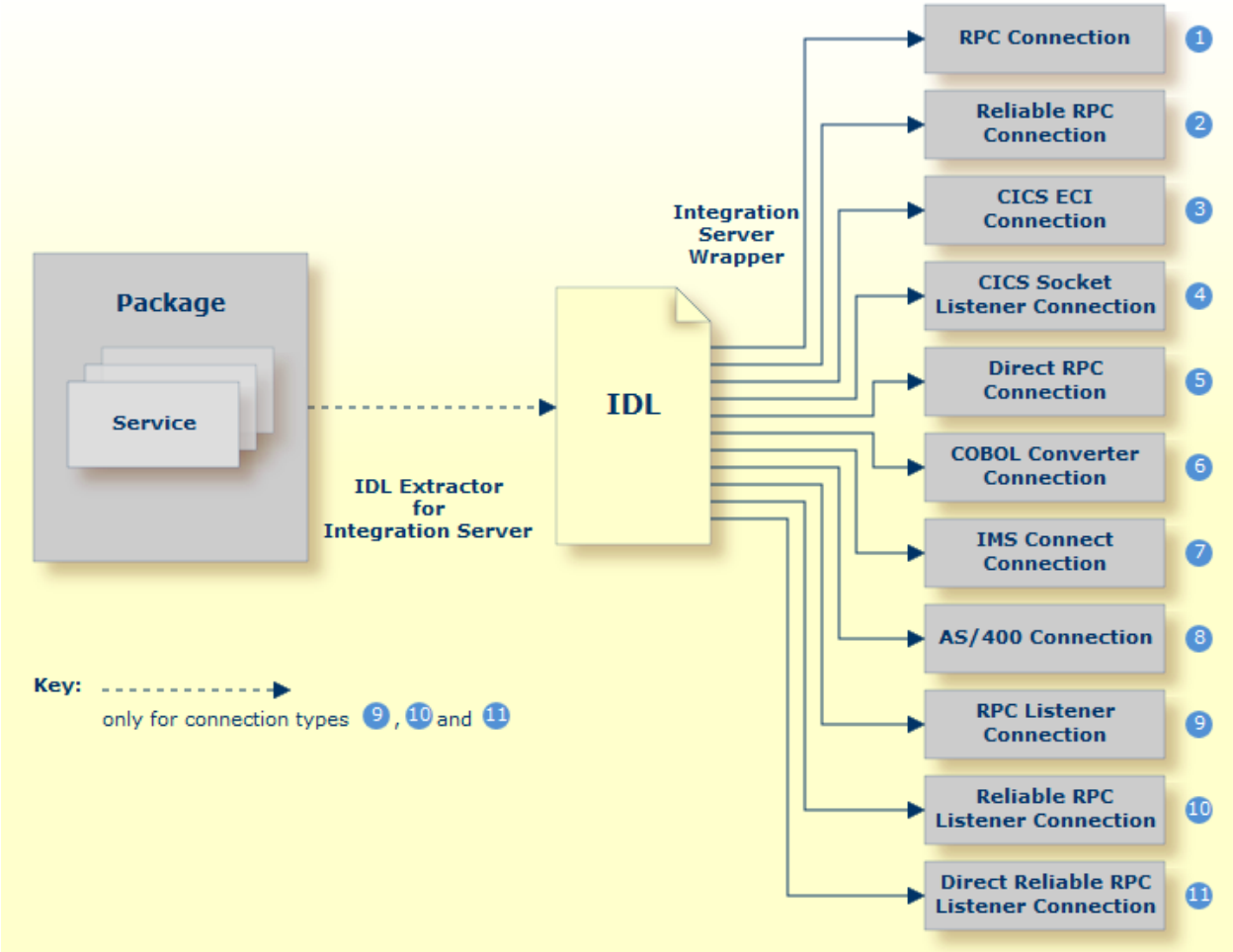
- Scope 6
- Prerequisites 11

Scope

The Integration Server Wrapper provides access to EntireX RPC-based components, IMS Connect, or CICS ECI from Integration Server services. A wizard generates Integration Server objects from a Software AG IDL file.

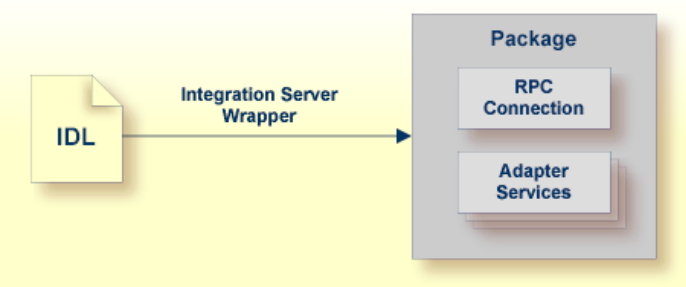


With the Integration Server Wrapper you can generate both client and server objects in the Integration Server. A client consists of a connection of type "RPC", "Reliable RPC", "CICS ECI", "Direct RPC", "IMS Connect" or "COBOL Connector". A server consists of a connection of type "RPC Listener" or "Reliable RPC Listener".



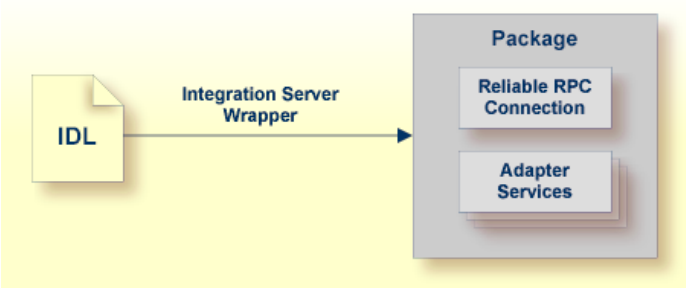
You can create the following objects:

- **RPC Connection**



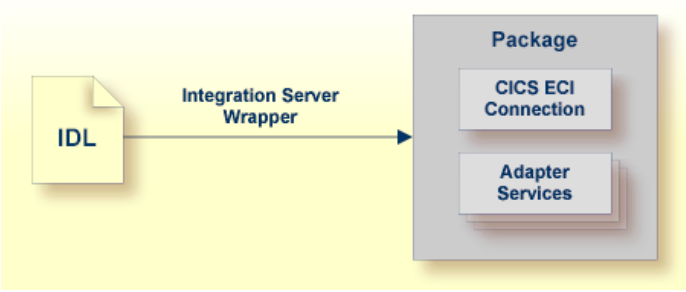
See *Step 3: Create or Update an Adapter Connection* and *Step 4a: Create a Connection and Related Adapter Services*.

- **Reliable RPC Connection**



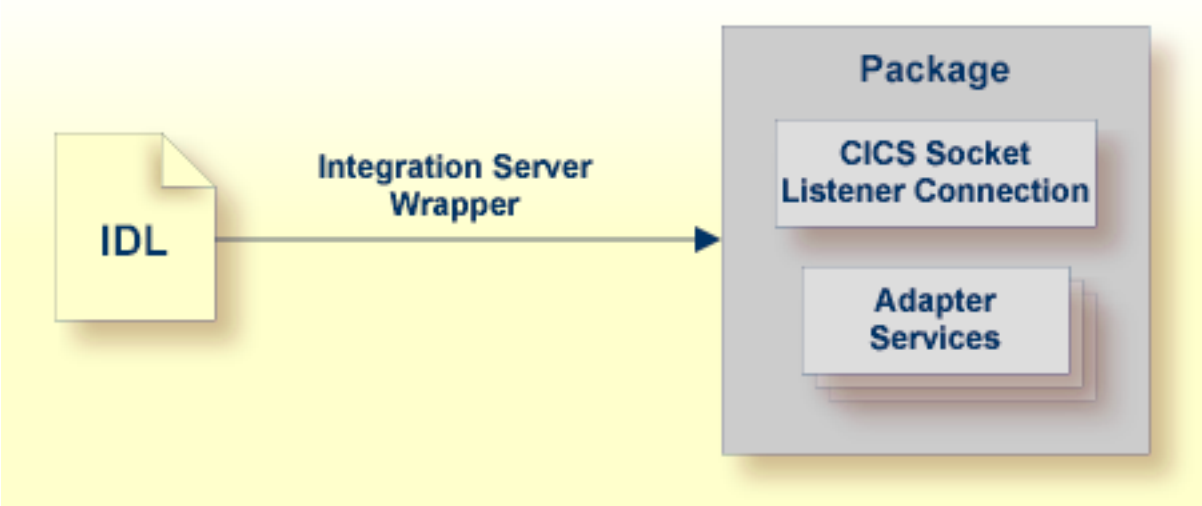
See *Step 3: Create or Update an Adapter Connection* and *Step 4a: Create a Connection and Related Adapter Services*.

- **CICS ECI Connection**



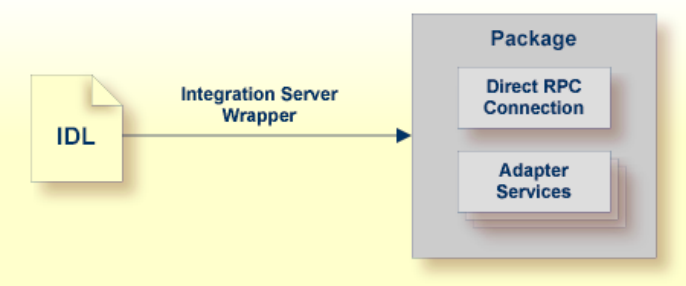
See *Step 3: Create or Update an Adapter Connection*.

- **CICS Socket Listener Connection**



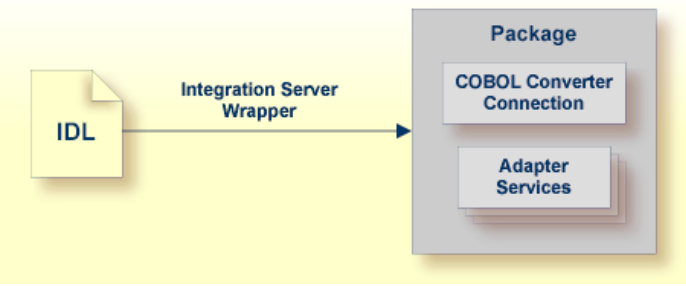
See *Step 3: Create or Update an Adapter Connection*.

- **Direct RPC Connection**



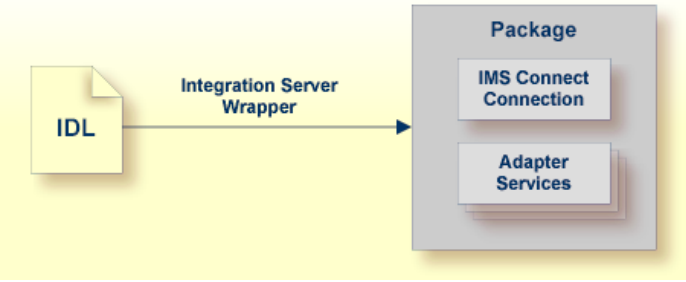
See *Step 3: Create or Update an Adapter Connection*.

- **COBOL Converter Connection**



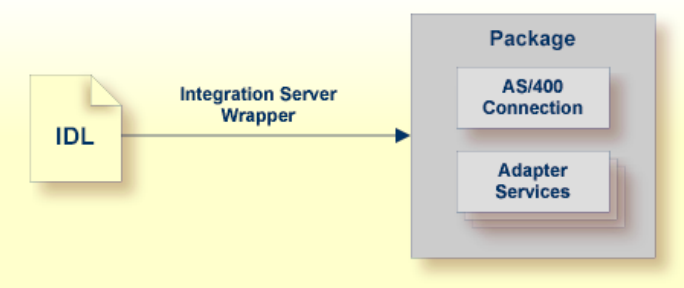
See *Step 3: Create or Update an Adapter Connection*.

- **IMS Connect Connection**



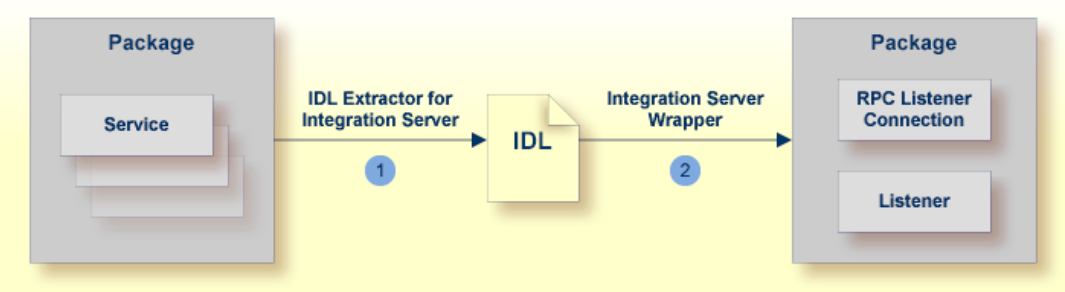
See *Step 3: Create or Update an Adapter Connection*.

- **AS/400 Connection**



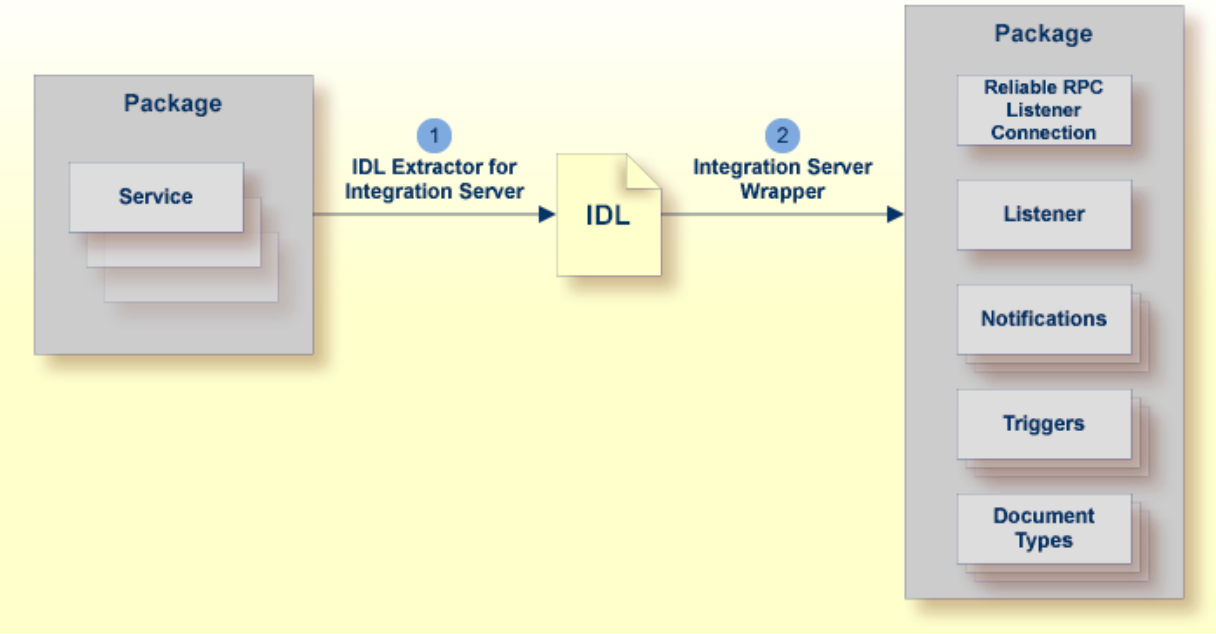
See *Step 3: Create or Update an Adapter Connection*.

- **RPC Listener**



See *Step 3: Create or Update an Adapter Connection* and *Step 4c: Create a Connection and Related Adapter Listener*.

- **Reliable RPC Listener**



See *Step 3: Create or Update an Adapter Connection* and *Step 4c: Create a Connection and Related Adapter Listener*.

Prerequisites

Four components are required for integration:

- the EntireX Adapter installed inside the Integration Server
- the EntireX Broker (not for Direct RPC connections, IMS Connect connections or CICS ECI connections, or COBOL Converter)
- the EntireX RPC Server (not for IMS Connect connections, CICS ECI connections, or ApplinX)
- the Software AG Designer for design time

Note: The Designer is used at design time. All other components are used also at runtime.

3 Using the Integration Server Wrapper

- Step 1: Start the Integration Server Wrapper Wizard 14
- Step 2a: Create a New Integration Server Connection 16
- Step 2b: Use an Existing Integration Server Connection 17
- Step 3: Create or Update an Adapter Connection 19
- Step 4a: Create a Connection and Related Adapter Services 21
- Step 4b: Create or Update a REST Resource 22
- Step 4c: Create a Connection and Related Adapter Listener 24
- Step 5: Finish the Wizard 26

Step 1: Start the Integration Server Wrapper Wizard

To start the Integration Server Wrapper wizard

1 In the context menu of a Software AG IDL file, choose **Integration Server > Generate webMethods IS Connection**.

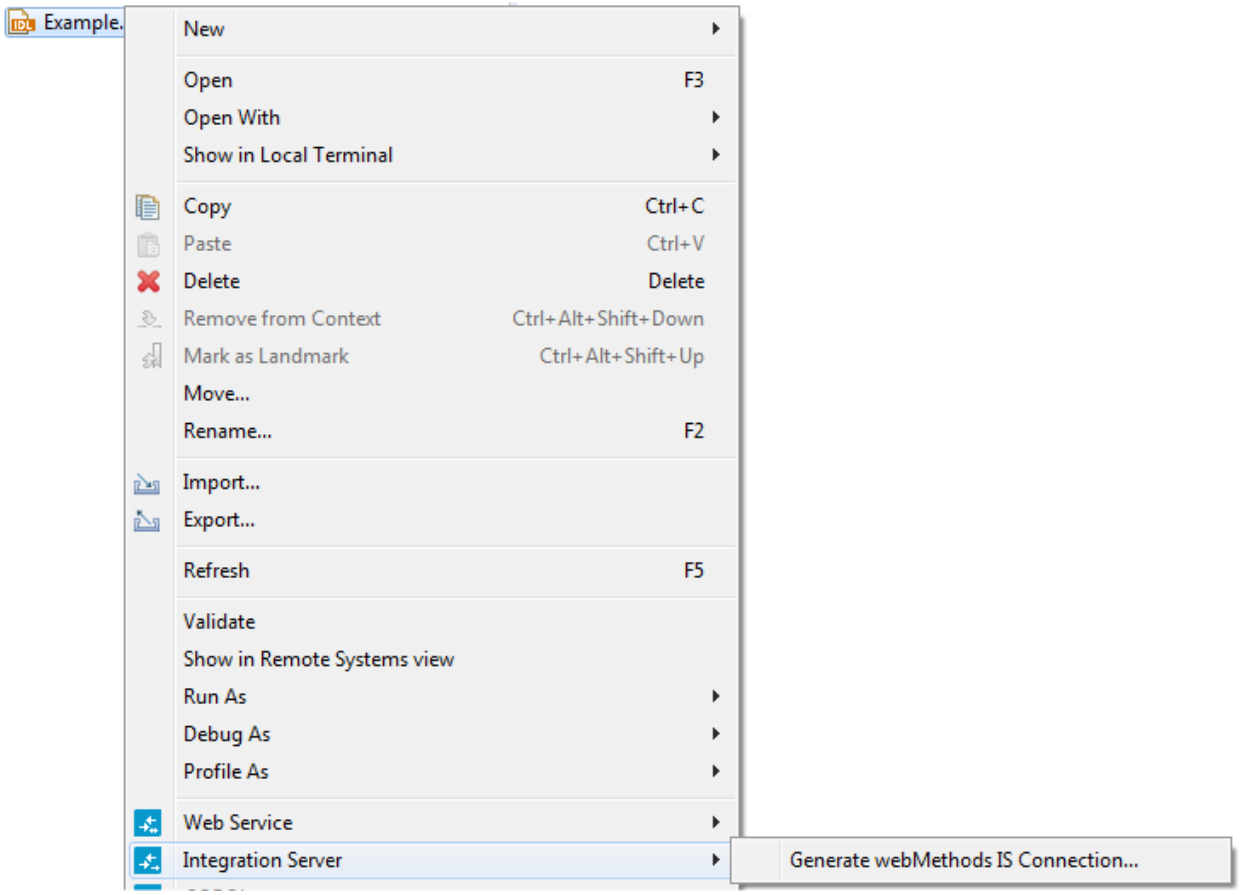
This starts the wizard with a list of existing Integration Server Wrapper connections.

Note: If the selected IDL file is not valid because of a syntax error, an error dialog comes up and the wizard does not start.

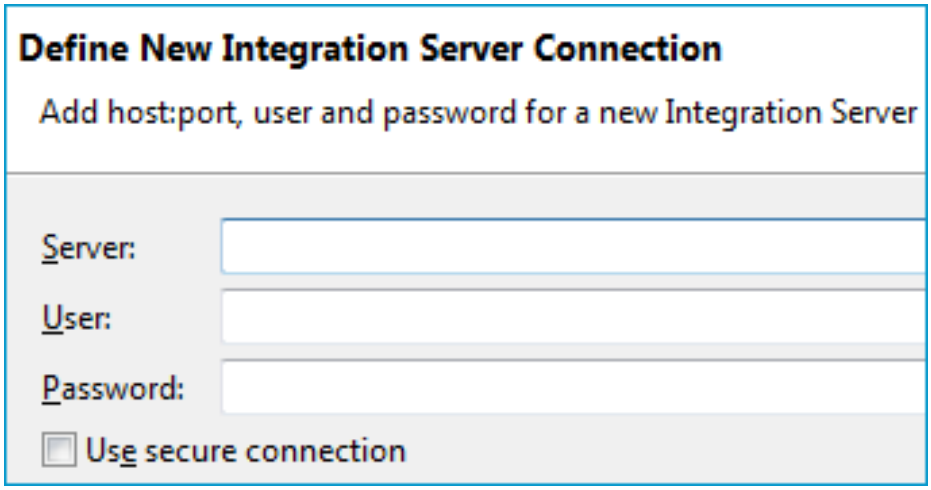
2 If you are using the wizard for the first time without any predefined Integration Server connections, continue with *Step 2a: Create a New Integration Server Connection*.

Or:

If *Integration Server Connections* are already defined, or if you want to communicate with an additional Integration Server, continue with *Step 2b: Use an Existing Integration Server Connection*.


Example.
New
Open F3
Open With
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Mark as Landmark Ctrl+Alt+Shift+Up
Move...
Rename... F2
Import...
Export...
Refresh F5
Validate
Show in Remote Systems view
Run As
Debug As
Profile As
Web Service
Integration Server
Generate webMethods IS Connection...

Step 2a: Create a New Integration Server Connection



To create a new Integration Server connection

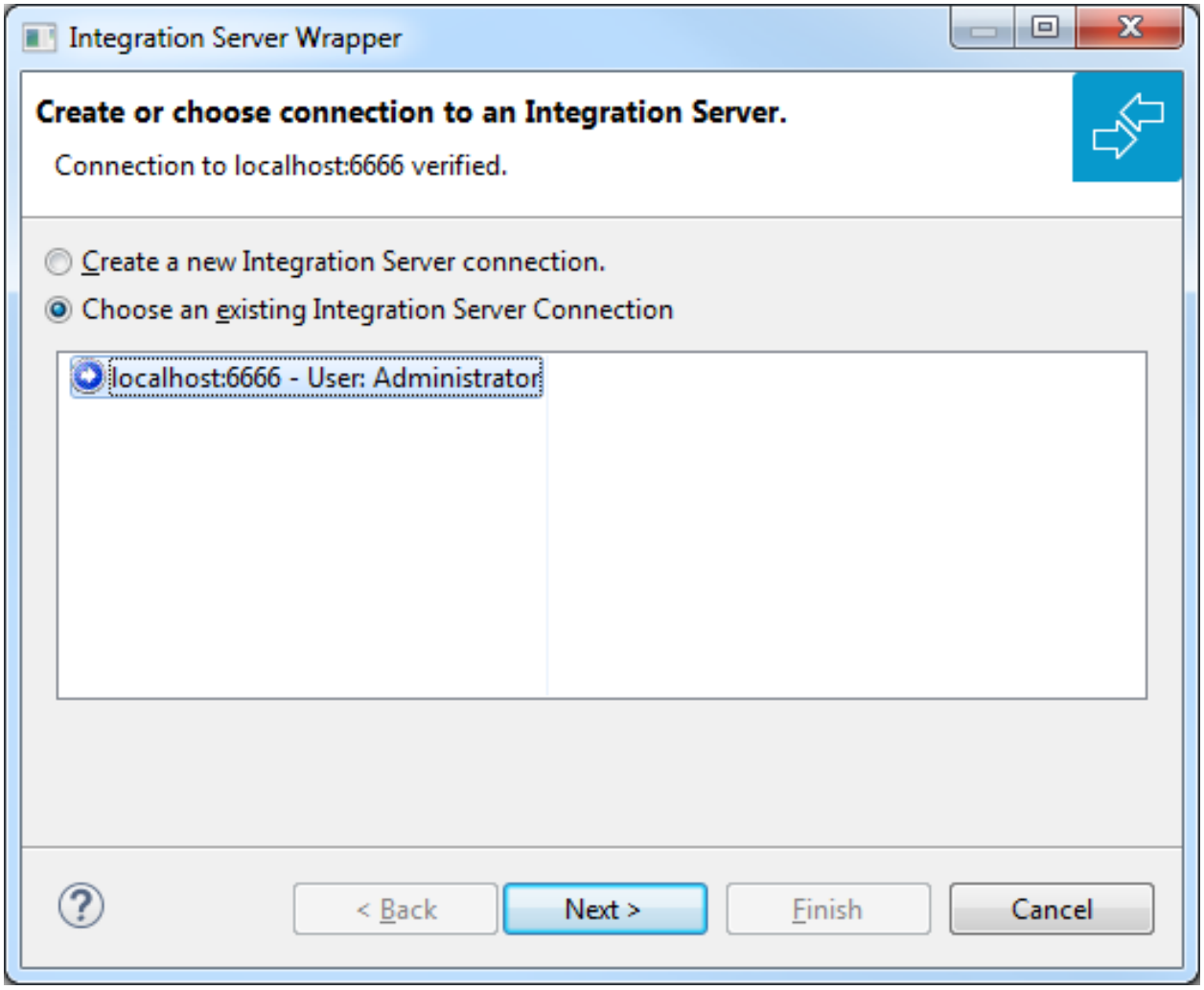
1 Define the new Integration Server connection on the wizard page.

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: Create or Update an Adapter Connection*.

Step 2b: Use an Existing Integration Server Connection



To use an existing Integration Server connection

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

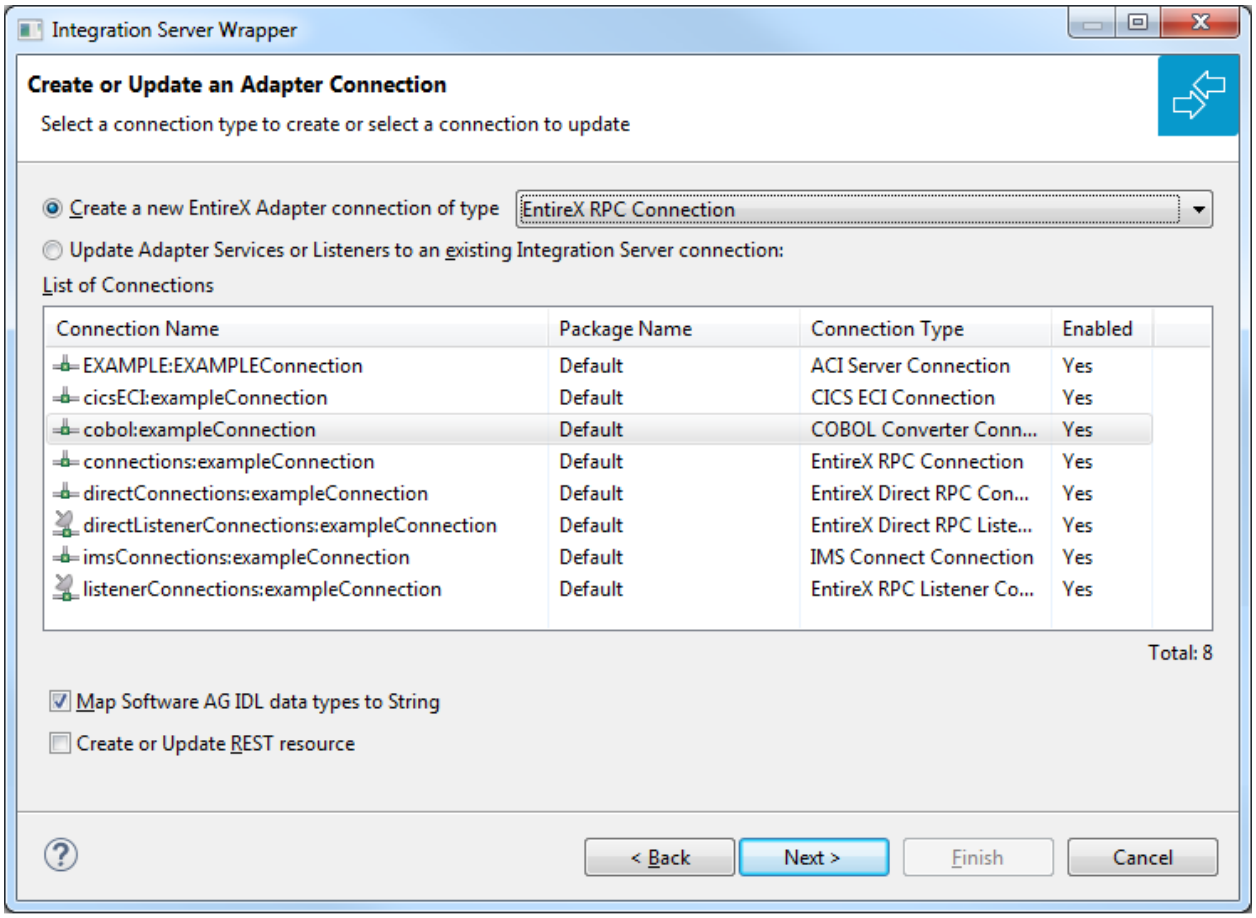
The selected connection is verified by a `ping` command. If the response is valid, the **Next** button is enabled. If invalid, an error message is displayed.

2 Continue with *Step 3: Create or Update an Adapter Connection*.

To create an additional Integration Server connection

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

Step 3: Create or Update an Adapter Connection



In this step you can either create a new Integration Server connection or update adapter services to an existing Integration Server connection.

To create a new connection

1 Select a connection type from the drop down list. Connection types are described under *EntireX Adapter Connections* in the EntireX Adapter documentation and ***Introduction to the Integration Server Wrapper***.

Note: The list of connection types is filtered: connection types that require a license are only shown if a corresponding license file is available. Reliable RPC connections are only shown if all IDL programs contain only IN parameters. Also, if a server mapping file is available, only those connection types that support the interface type specified in the server mapping file are shown.

2 Click **Next** and continue with *Step 4a: Create a Connection and Related Adapter Services*.

Or:

Step 4c: Create a Connection and Related Adapter Listener, depending on the selected connection type.

To update an existing connection or listener

1 Select a connection from **List of Connections**. The list of connections is filtered by connection type as described in the note above.

As a result, you are informed on how many adapter services or adapter listeners will be created, modified or left unchanged.

The update process can be characterized as follows:

- The metadata is updated for each IDL program.
- An adapter service is created for each new IDL program or one adapter listener is created.
- An existing adapter service or adapter listener is updated if it is contained in the IDL file for the update.
- A connection remains unchanged with respect to its type and settings (broker ID, server address, user ID, etc.).

2 Click **Finish**.

The checkbox **Map Software AG IDL data types to String** specifies how to map IDL data types to Integration Server data types. The default is to map all data types (except binary) to String. See *Mapping IDL Data Types to IS Data Types*. If the IDL file has been extracted from an Integration Server package, this option is not available because the value was set by the extraction and should not be changed.

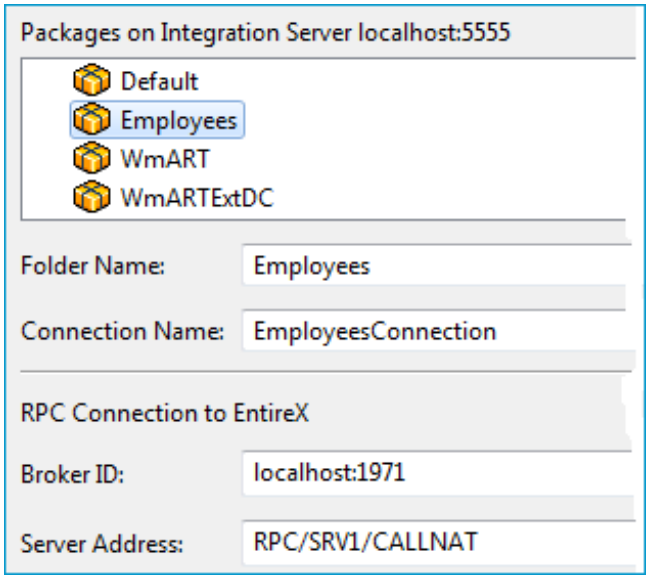
To create or update a REST resource

- Check **Create or Update REST resource** if you want to create REST V2 resources for your connection. This functionality is not available for Listener connections. An additional dialog will appear to define the destination of the resource See *Step 4b: Create or Update a REST Resource*.

Example

A connection is created with IDL programs `A` and `B`. Later, the update operation uses IDL programs `A` and `C`. After the update, the service for `A` is modified, the service for `B` is unchanged, and the service for `C` is new.

Step 4a: Create a Connection and Related Adapter Services



To create a connection and related adapter services

1 Select a package for the created objects.

2 Define a folder name. If the folder does not exist, it will be created.

3 Define a connection name.

4 Define the parameters of the connection type. For details, see the *EntireX and your webMethods Integration Server Applications*.

As a result, the folder will contain the connection and the adapter services (one for each IDL program). The name of a service is the same as the respective IDL program.

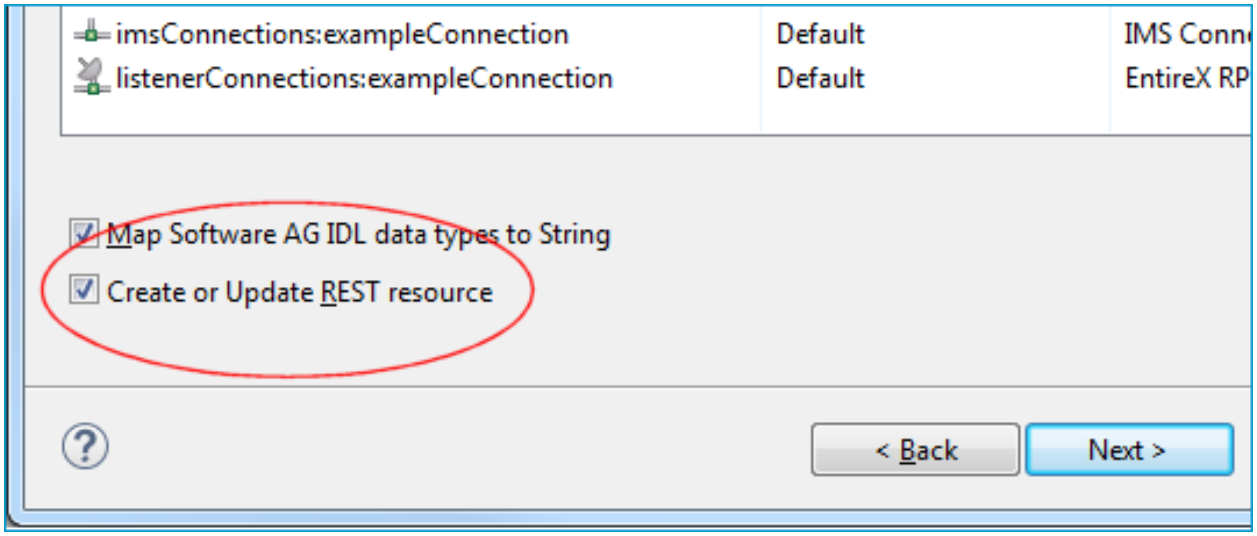
The default settings for the adapter services are:

- the **Default** package; if not available, the first package
- the IDL library name for the **Folder Name**
- the IDL library name with the suffix "Connection" for the **Connection Name**

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.

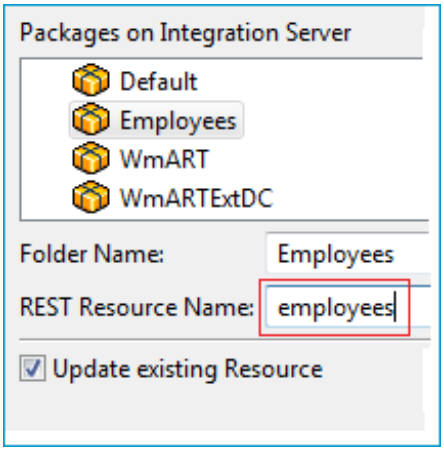
Step 4b: Create or Update a REST Resource

The following screen appears if checkbox **Create or Update REST resource** is checked in Step 3.



To create or update a REST resource

1 Select a package.



2 Define a folder name. If the folder does not exist, it will be created.

3 Define the REST Resource name.

4 Check **Update existing Resource** if you want to add additional services to an existing resource.

For more information see:

- *REST Developer's Guide* in the webMethods Integration Server documentation for information on using the generated REST resources
- *Configuring a REST Resource Using the URL Template-Based Approach* in the online help provided under *Software AG Designer Guides* > *webMethods Service Development Help* > *Working with REST*

Step 4c: Create a Connection and Related Adapter Listener

To create a connection and related listener

1 Select a package for the created objects.

2 Define a folder name. If the folder does not exist, it will be created.

3 Define a connection name.

4 Define the parameters of the connection type.

As a result, the folder will contain the connection and the adapter listener.

The default settings for new listener are:

- the **Default** package; if not available, the first package
- the IDL library name for the **Folder Name**
- the IDL library name with the suffix "Connection" for the **Connection Name**
- the IDL library name with the suffix "Listener" for the **Listener Name**

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.

Step 5: Finish the Wizard

To finish the Wizard

- Choose **Finish**.

  As a result, the folder contains multiple objects as listed under the corresponding step.

  Note: See *Software AG IDL to Integration Server Mapping*.

4 Software AG IDL to Integration Server Mapping

- Mapping IDL Data Types to IS Data Types 28
- Mapping Groups 30
- Mapping Arrays 30

This chapter describes the specific mapping of Software AG IDL data types, groups and arrays to the webMethods Integration Server. Please note also the remarks and hints on the IDL data types valid for all language bindings found under *Software AG IDL File* in the IDL Editor documentation.

Mapping IDL Data Types to IS Data Types

In the table below, the following metasymbols and informal terms are used for the IDL.

- The metasymbols "[" and "]" enclose optional lexical entities.
- The informal term *number* (or in some cases *number1*.*number2*) is a sequence of numeric characters, for example 123.

Software AG IDL	Description	Integration Server Data Type		Note
		using Map to String	using Map to Java Wrapper Types	
A[*number*]	Alphanumeric	java.lang.String	java.lang.String	1
AV	Alphanumeric variable length	java.lang.String	java.lang.String	
AV[*number*]	Alphanumeric variable length with maximum length	java.lang.String	java.lang.String	
B*number*	Binary	byte[]	byte[]	
BV	Binary variable length	byte[]	byte[]	
BV[*number*]	Binary variable length with maximum length	byte[]	byte[]	
D	Date	java.lang.String	java.lang.String	3,5
F4	Floating point (small)	java.lang.String	java.lang.Float	
F8	Floating point (large)	java.lang.String	java.lang.Double	
I1	Integer (small)	java.lang.String	java.lang.Integer	
I2	Integer (medium)	java.lang.String	java.lang.Integer	
I4	Integer (large)	java.lang.String	java.lang.Intger	
K*number*	Kanji	java.lang.String	java.lang.String	1

Software AG IDL	Description	Integration Server Data Type		Note
		using Map to String	using Map to Java Wrapper Types	
KV	Kanji variable length	java.lang.String	java.lang.String	
KV[*number*]	Kanji variable length with maximum length	java.lang.String	java.lang.String	
L	Logical	java.lang.String	java.lang.Boolean	
N*number1*[.*number2*]	Unpacked decimal	java.lang.String	java.lang.String/java.lang.Long	2,6
NU*number1*[.*number2*]	Unpacked decimal unsigned	java.lang.String	java.lang.String/java.lang.Long	2,6
P*number1*[.*number2*]	Packed decimal	java.lang.String	java.lang.String/java.lang.Long	2,6
PU*number1*[.*number2*]	Packed decimal unsigned	java.lang.String	java.lang.String/java.lang.Long	2,6
T	Time	java.lang.String	java.lang.String	4,5
U*number*	Unicode	java.lang.String	java.lang.String	1
UV	Unicode variable length	java.lang.String	java.lang.String	
UV*number*	Unicode variable length with maximum length	java.lang.String	java.lang.String	

Notes:

1. You have the option to keep leading and trailing whitespace characters. Default is to trim these whitespace characters. This can be configured individually for each adapter service.
2. You have the option to keep or remove leading zeros or the decimal point. Default is to remove leading zeros and to keep the decimal point. This can be configured individually for each adapter service.
3. The default pattern for formatting the date string in the pipeline is "MMM d, yyyy" using the locale "en", e.g. "Mar 2, 2016".
4. The default pattern for formatting the time string in the pipeline is "MMM d, yyyy h:mm:ss a" using the locale "en", e.g. "Mar 2, 2016 11:15:10 AM".
5. The default pattern can be changed in the **Settings** dialog of the EntireX Adapter. See *Adapter Settings*.

6. Decimal numbers are mapped to `Long` only if *`number1`*<=19 and *`number2`*=0. Otherwise they are mapped to `String`.

Mapping Groups

Groups are mapped to documents.

Mapping Arrays

- One-dimensional arrays are mapped to String lists (`java.lang.String[]`). Or see above.
- Two-dimensional arrays are mapped to String tables (`java.lang.String[][]`). Or see above.
- Three-dimensional arrays are mapped to `java.lang.String[][][]`. Or see above.

Note: Binary data is mapped to `byte[]`. For example, a one-dimensional array is mapped to byte list (`byte[]`) etc.

5 Integration Server Preferences

- Integration Server Connections 32
- Setting Integration Server Preferences 34

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.

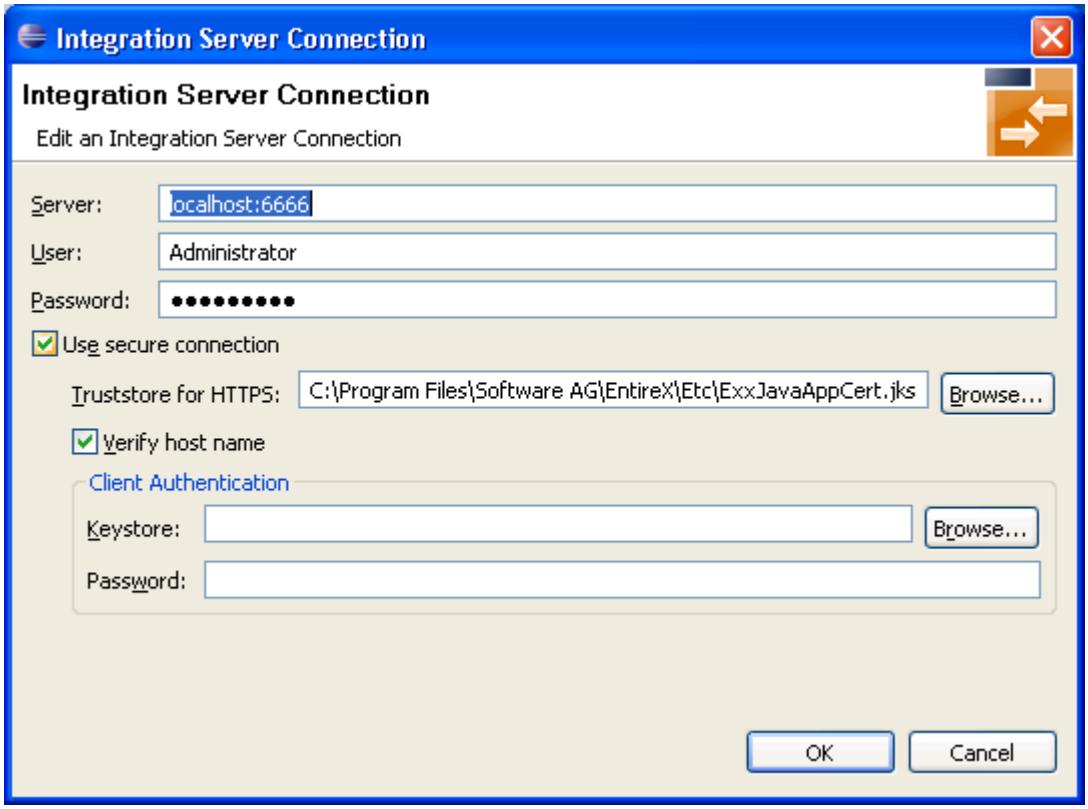
Integration Server Connections

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

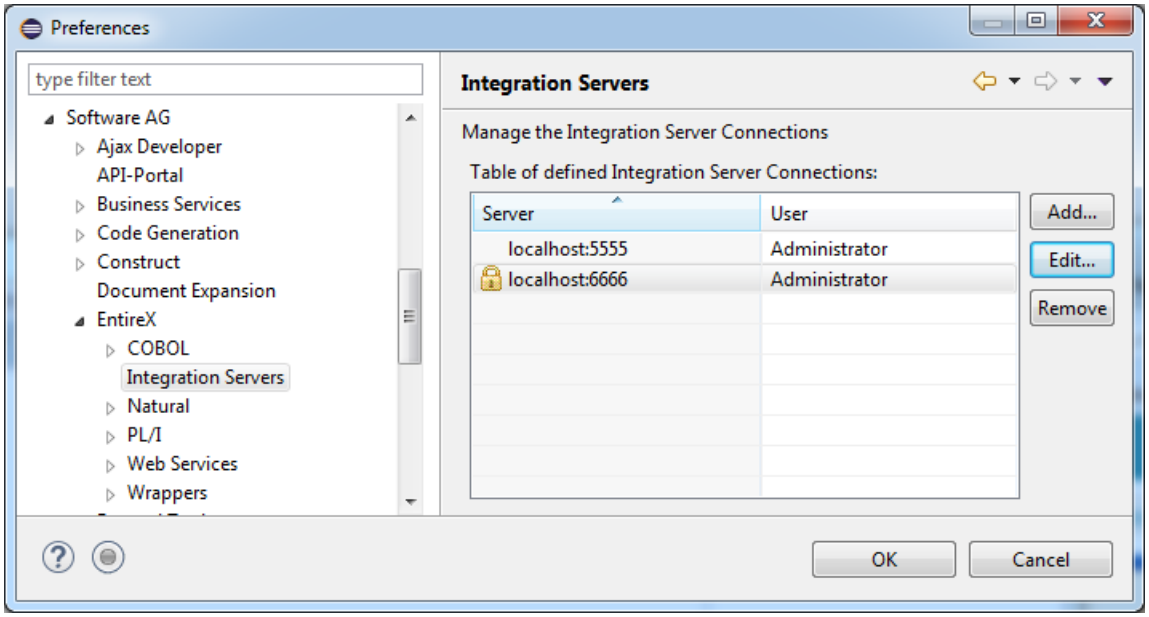
This information can be specified in the following dialog:



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.