software AG

webMethods EntireX

webMethods Integration Server Wrapper

Version 9.6

April 2014

webMethods EntireX

This document applies to webMethods EntireX Version 9.6.

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

Document ID: EXX-EEXXXAIWRAPPER-96-20140628

Table of Contents

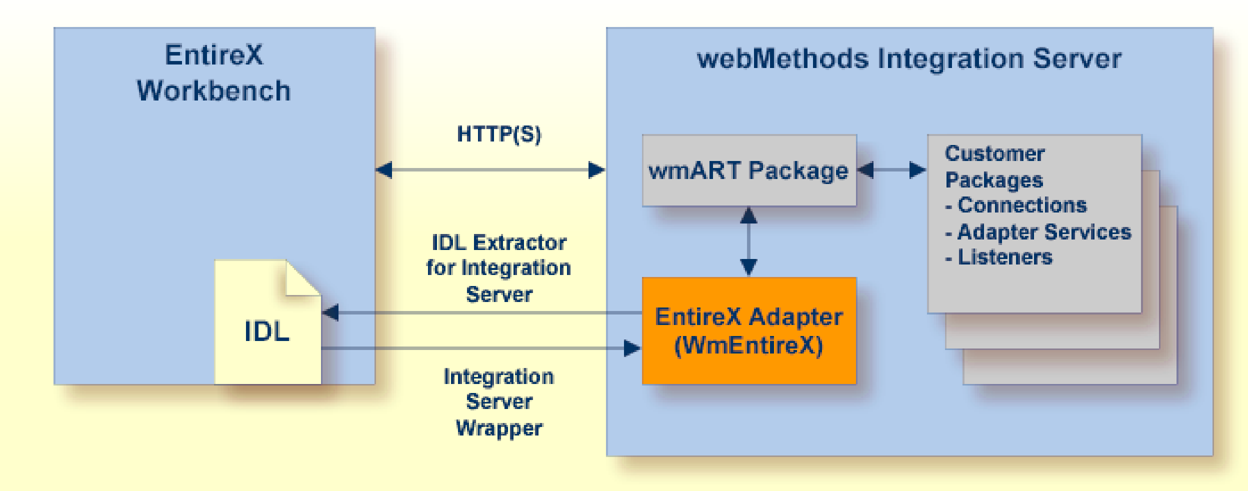
webMethods Integration Server Wrapper

The webMethods Integration Server Wrapper generates Integration Server adapter services and listeners from a Software AG IDL file within an Integration Server connection definition.

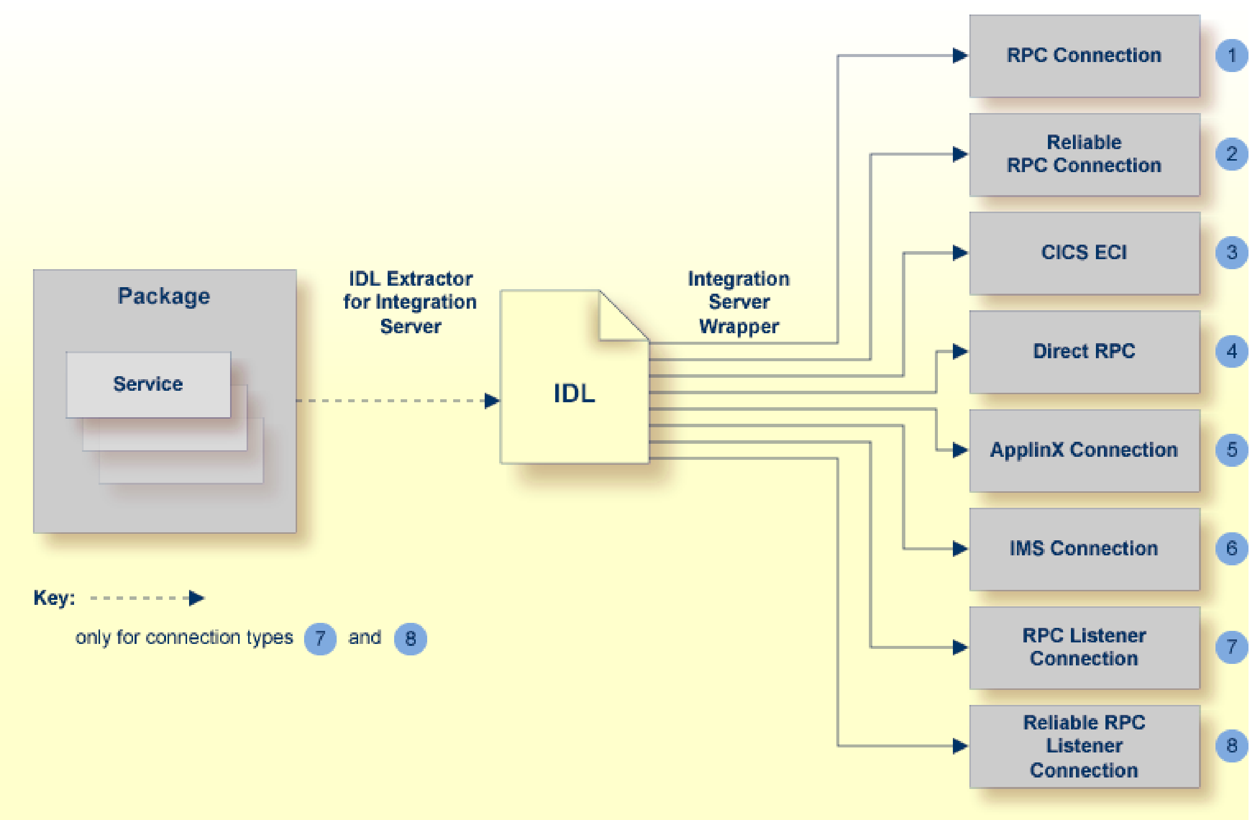
1 Introduction to the Integration Server Wrapper

Scope

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

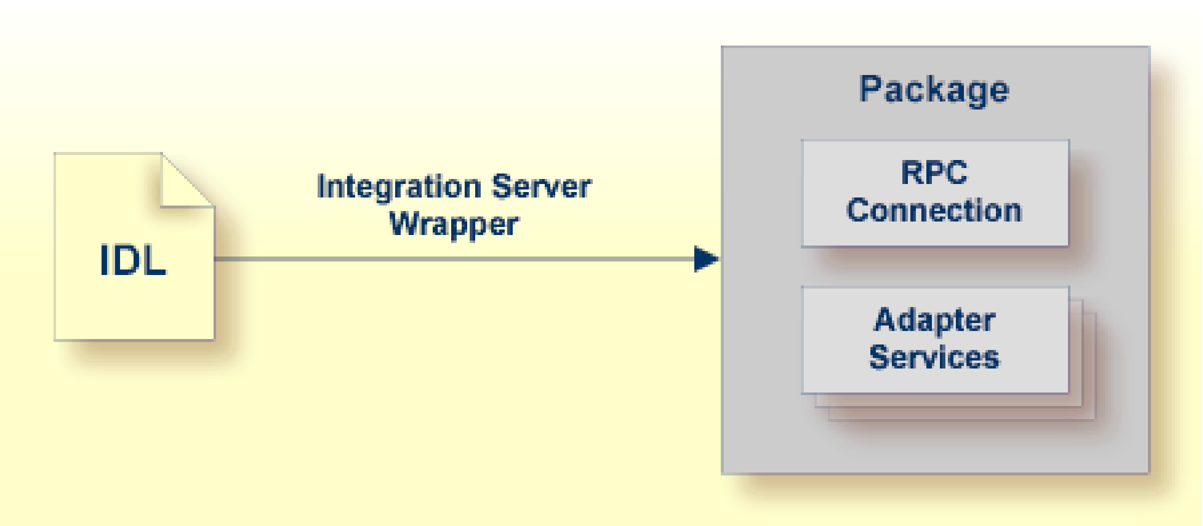

EntireX Workbench
IDL
HTTP(S)
IDL Extractor for Integration Server
Integration Server Wrapper
webMethods Integration Server
wmART Package
Customer Packages
- Connections
- Adapter Services
- Listeners
EntireX Adapter (WmEntireX)

The Integration Server Wrapper enables users to 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", "ApplinX" or "IMS Connect". A server consists of a connection of type "RPC Listener" or "Reliable RPC Listener".



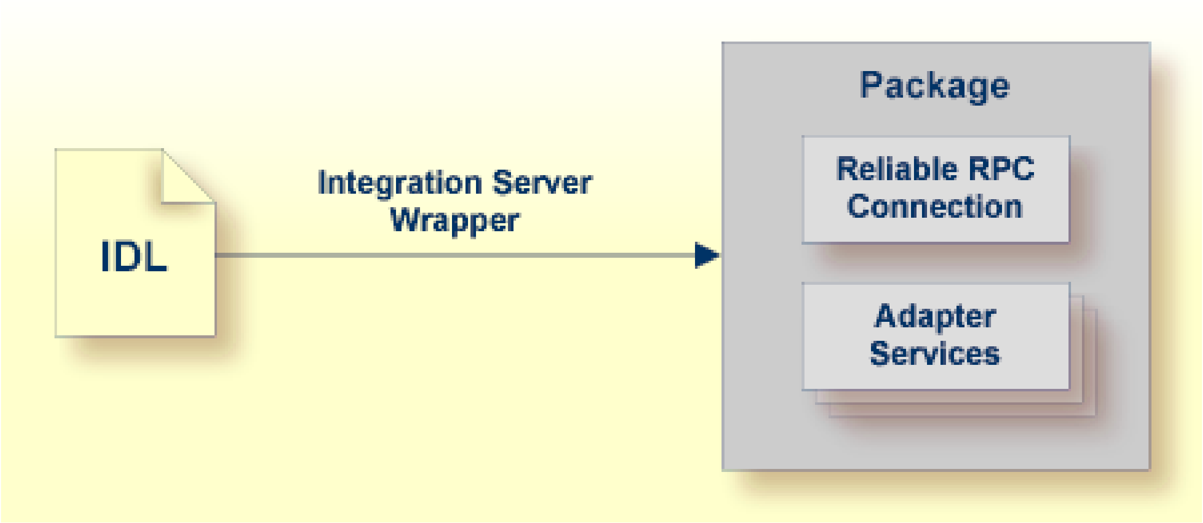
It is possible to create the following objects:

- **RPC Connection**



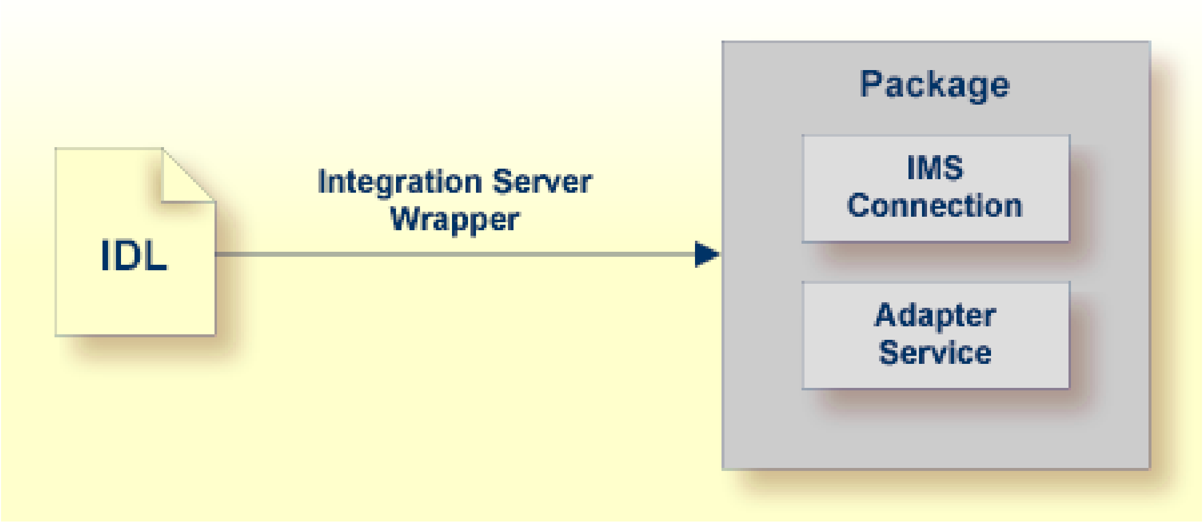
See *Step 4a: Define Adapter Services for an RPC Connection*.

- **Reliable RPC Connection**



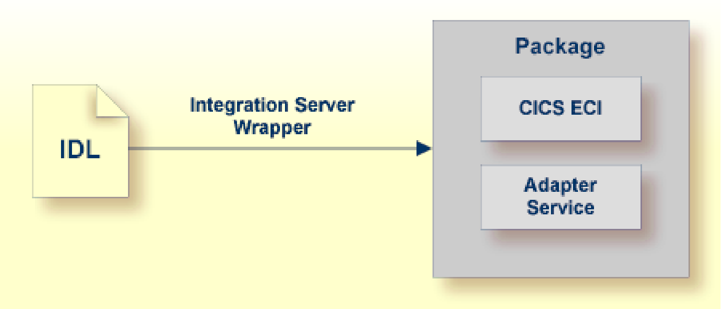
See *Step 4a: Define Adapter Services for an RPC Connection*.

- **Connection to IMS Connect**



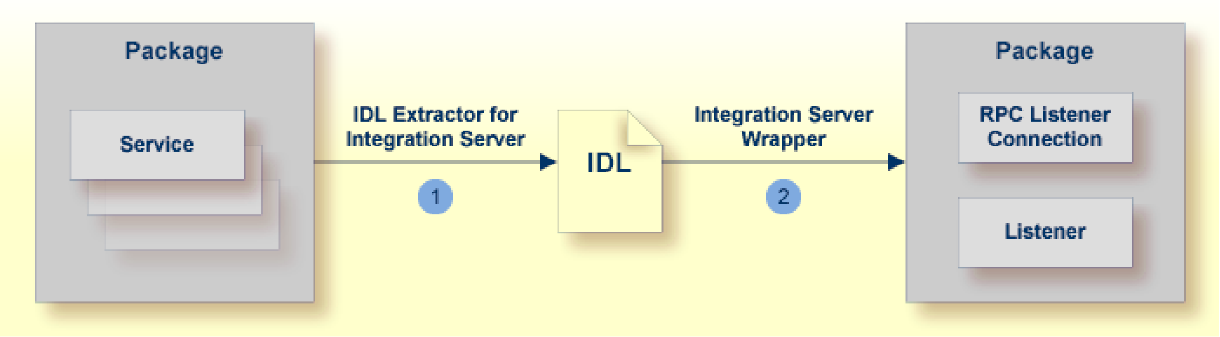
See *Step 4a: Define Adapter Services for an RPC Connection*.

- **CICS ECI Connection**



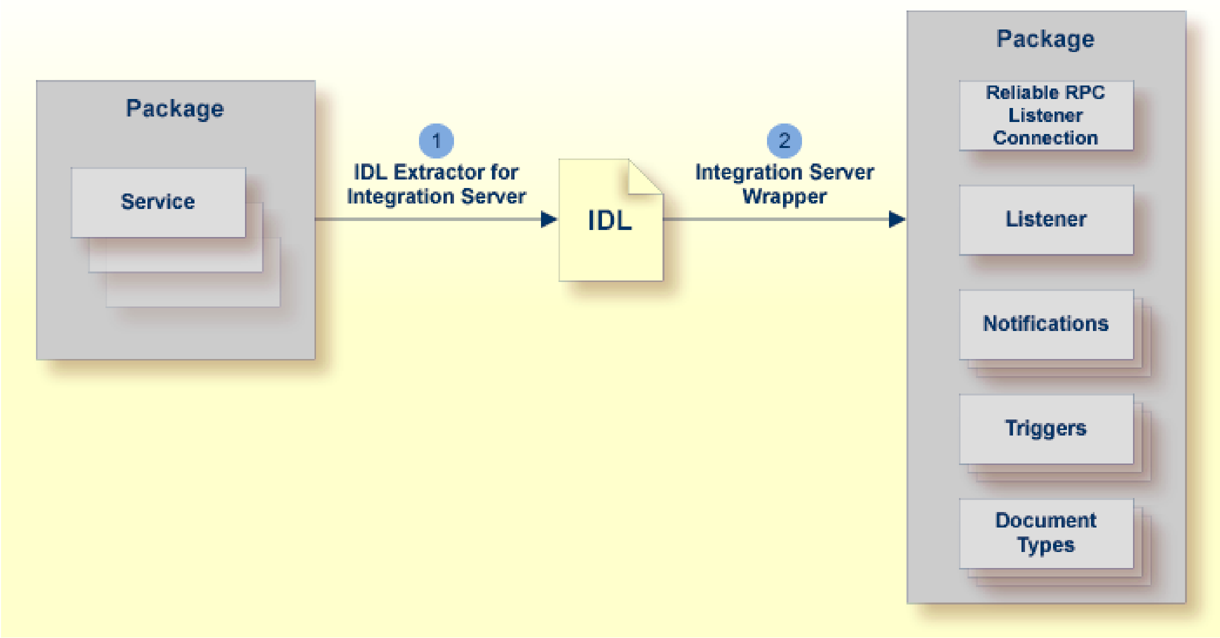
See *Step 4a: Define Adapter Services for an RPC Connection*.

- **RPC Listener**


Package
Service
IDL Extractor for Integration Server
1
IDL
Integration Server Wrapper
2
Package
RPC Listener Connection
Listener

1. See *Using the IDL Extractor for Integration Server*.
2. See *Step 4b: Define Adapter Services for an RPC Listener or a Reliable RPC Listener*.

- **Reliable RPC Listener**


Package
Service
1
IDL Extractor for Integration Server
IDL
2
Integration Server Wrapper
Package
Reliable RPC Listener Connection
Listener
Notifications
Triggers
Document Types

1. See *Using the IDL Extractor for Integration Server*.
2. See ***Step 4b: Define Adapter Services for an RPC Listener or a Reliable RPC Listener***.

Prerequisites

Four components are required for integration:

- the EntireX Adapter installed inside the Integration Server
- the EntireX Broker (not for Direct RPC connections, connections to IMS Connect, or CICS ECI Connections, or ApplinX)
- the EntireX RPC Server (not for Connections to IMS Connect or CICS ECI Connections, or ApplinX)
- the EntireX Workbench for design time (not for ApplinX connections)

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

For ApplinX connections we have the following prerequisites:

- the EntireX Adapter installed inside the Integration Server
- the ApplinX server
- the ApplinX Workbench (inside Software AG Designer)

2 Using the Integration Server Wrapper

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 **Generate webMethods IS Connection from Software AG IDL...**.

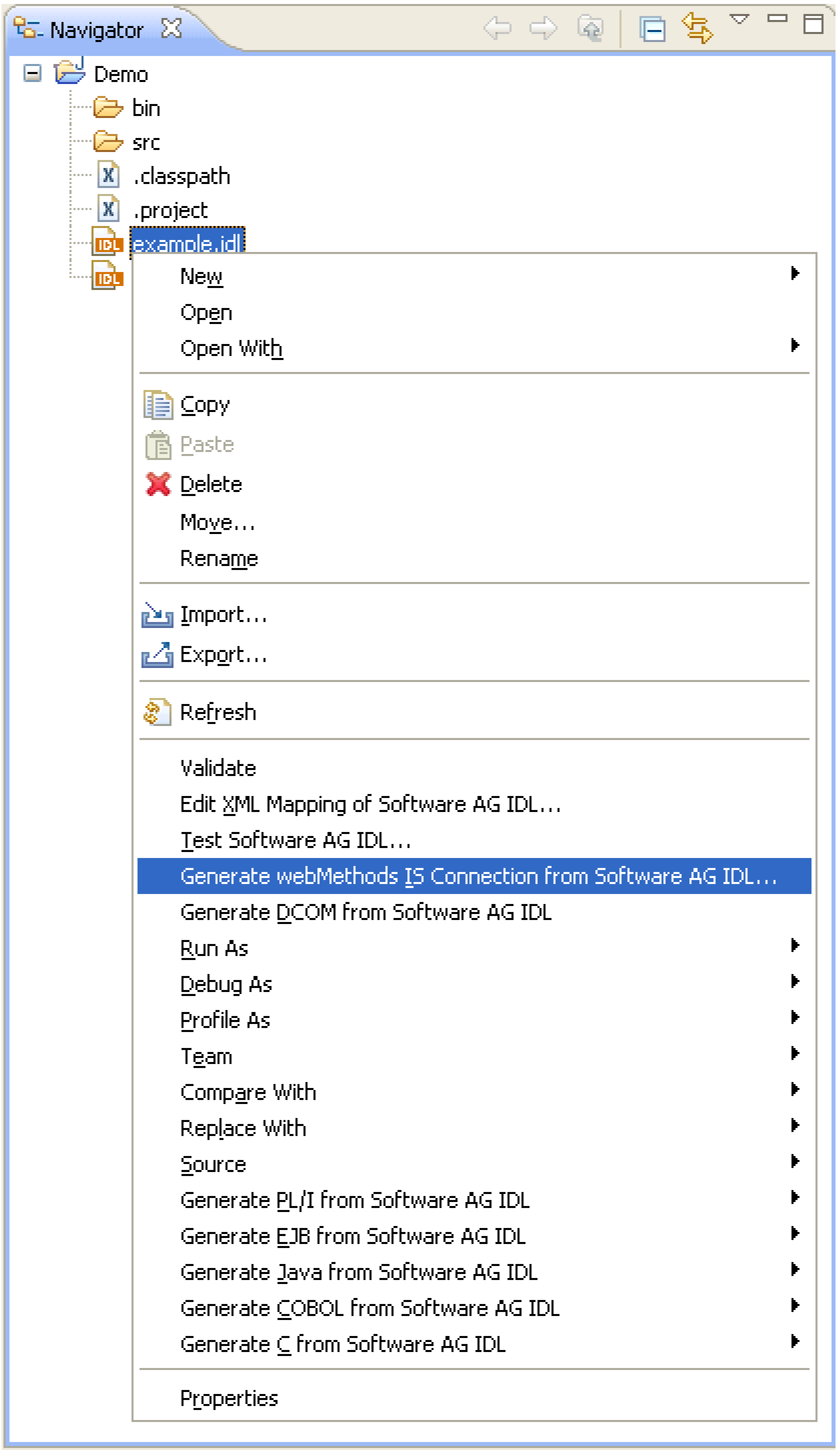
This starts the wizard with a list of existing Integration Server Wrapper connections.

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.

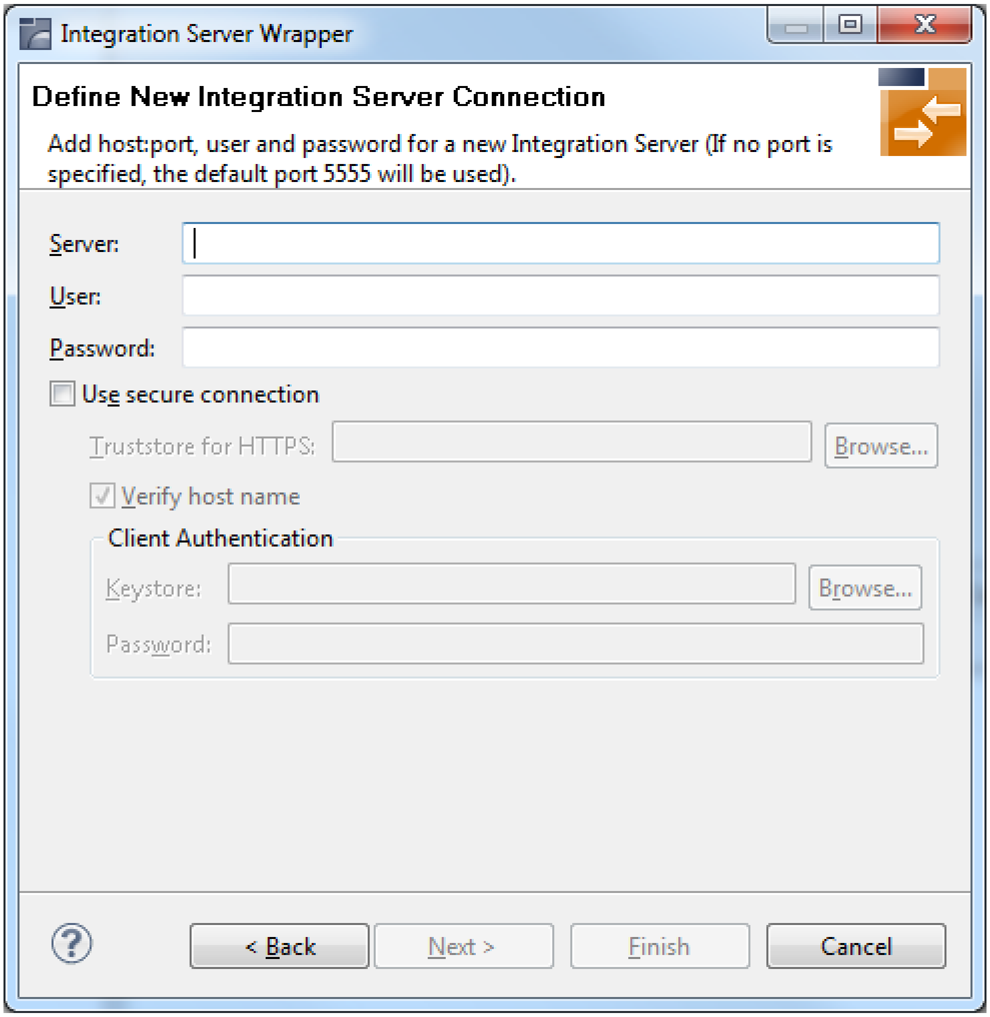
2 If you are using the wizard for the first time without any predefined Integration Server connections, continue with ***Step 2a: Create a New Integration Server Connection***.

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***.


Navigator
Demo
bin
src
.classpath
.project
example.idl
New
Open
Open With
Copy
Paste
Delete
Move...
Rename
Import...
Export...
Refresh
Validate
Edit XML Mapping of Software AG IDL...
Test Software AG IDL...
Generate webMethods IS Connection from Software AG IDL...
Generate DCOM from Software AG IDL
Run As
Debug As
Profile As
Team
Compare With
Replace With
Source
Generate PL/I from Software AG IDL
Generate EJB from Software AG IDL
Generate Java from Software AG IDL
Generate COBOL from Software AG IDL
Generate C from Software AG IDL
Properties

Step 2a: Create a New Integration Server Connection



Integration Server Wrapper

Define New Integration Server Connection

Add host:port, user and password for a new Integration Server (If no port is specified, the default port 5555 will be used).

Server:

User:

Password:

□ Use secure connection

Truststore for HTTPS:

☑ Verify host name

Client Authentication

Keystore:

Password:

To create a new Integration Server connection

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

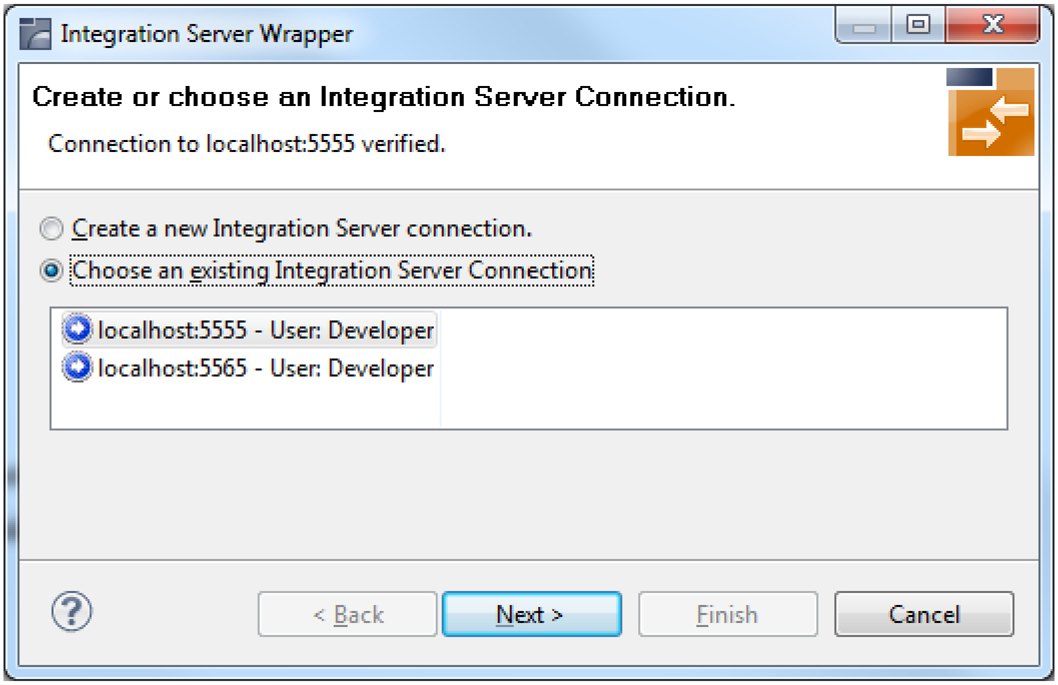
Notes:

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 ***Integration Server Preferences***.

2 Choose **Next** and continue with ***Step 3: Select the Connection Type***.

Step 2b: Use an Existing Integration Server Connection



Create or choose an Integration Server Connection.

Connection to localhost:5555 verified.

○ Create a new Integration Server connection.

● Choose an existing Integration Server Connection

localhost:5555 - User: Developer

localhost:5565 - User: Developer

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.

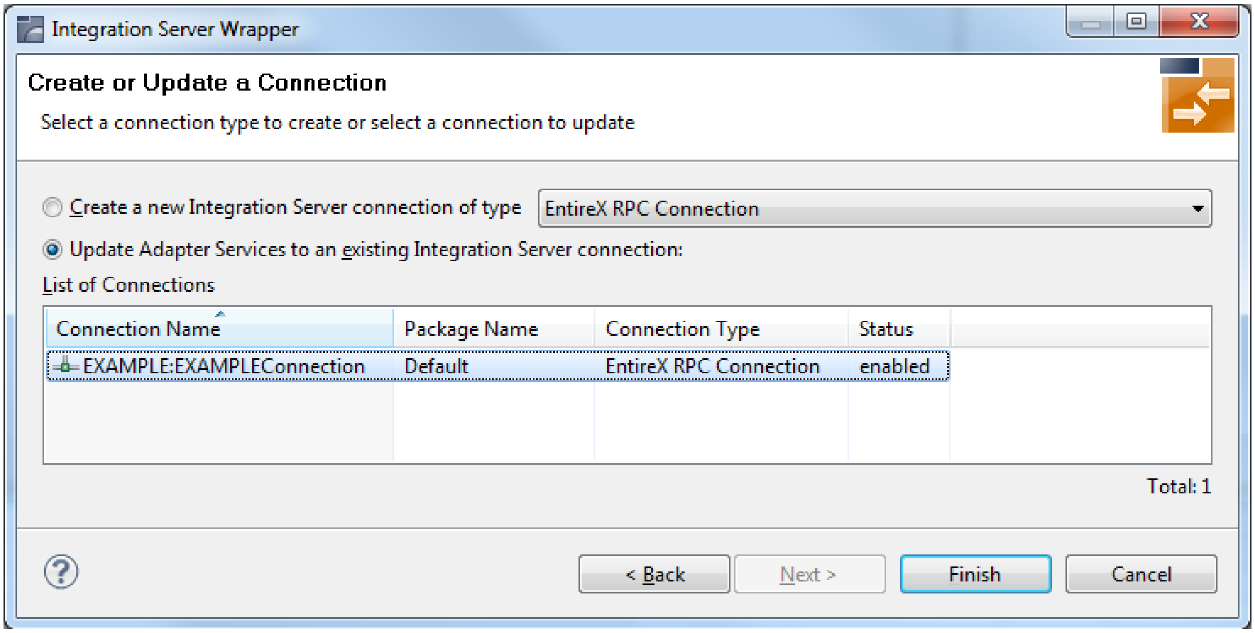
2 Continue with *Step 3: Select the Connection Type*.

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: Select the Connection Type

Note: This section describes how to select the connection type for EntireX Adapter version 9.0 or above. For earlier versions, see *Select the Connection Type for EntireX Adapter 8.2 or Below*.



Integration Server Wrapper

Create or Update a Connection

Select a connection type to create or select a connection to update

○ Create a new Integration Server connection of type: EntireX RPC Connection

● Update Adapter Services to an existing Integration Server connection:

List of Connections

Connection Name	Package Name	Connection Type	Status
EXAMPLE:EXAMPLEConnection	Default	EntireX RPC Connection	enabled

Total: 1

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.

2 Click **Next** and continue with *Step 4a: Define Adapter Services for an RPC Connection*.

Or:

Step 4b: Define Adapter Services for an RPC Listener or a Reliable RPC Listener, depending on the selected connection type.

To update an existing connection

1 Select a connection from **List of Connections**.

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

The update process can be characterized as follows:

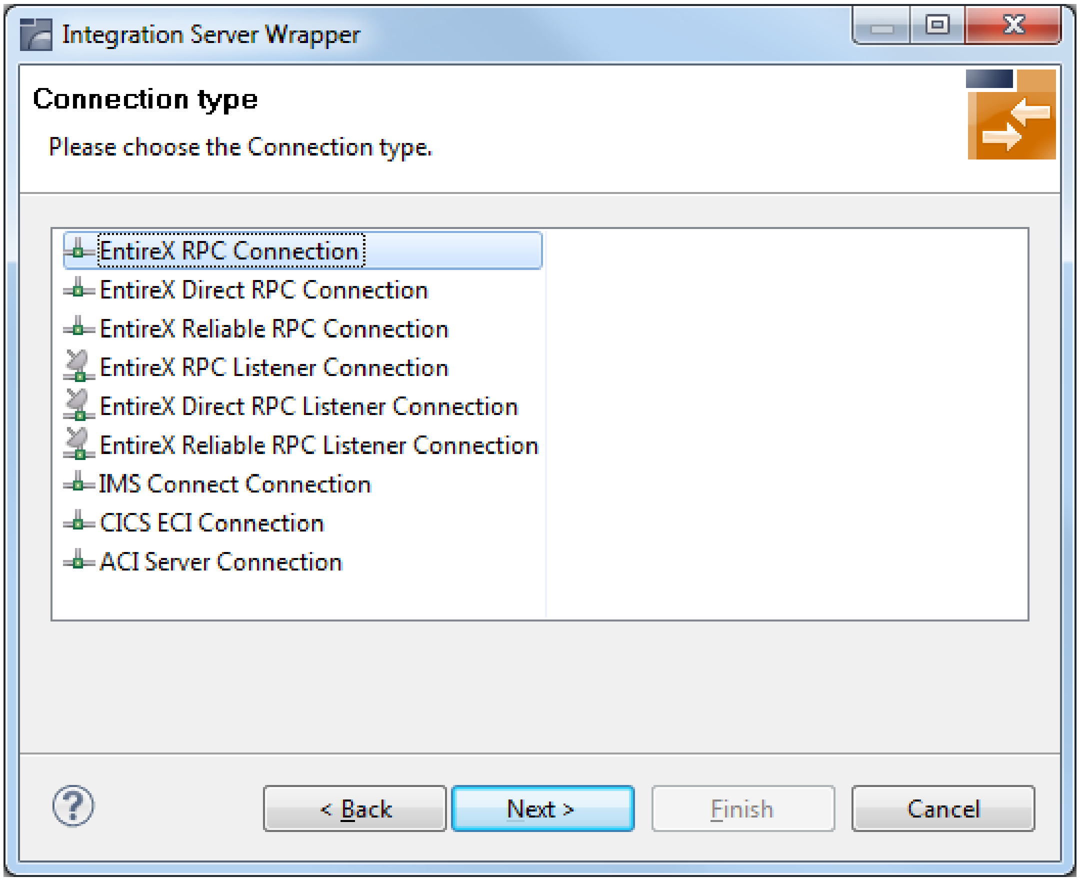
- The metadata is updated for each IDL program.
- An adapter service is created for each new IDL program.
- An existing adapter service is updated if it is contained in the IDL file for the update.
- A connection remains unchanged with respect to its type and settings (broker ID, server address, user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

2 Click **Finish**.

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.

Select the Connection Type for EntireX Adapter 8.2 or Below


Integration Server Wrapper
Connection type
Please choose the Connection type.
EntireX RPC Connection
EntireX Direct RPC Connection
EntireX Reliable RPC Connection
EntireX RPC Listener Connection
EntireX Direct RPC Listener Connection
EntireX Reliable RPC Listener Connection
IMS Connect Connection
CICS ECI Connection
ACI Server Connection
< Back
Next >
Finish
Cancel

To select a connection type forEntireX Adapter 8.2 or below

1 Select a **Connection type** from the list of available types.

Notes:

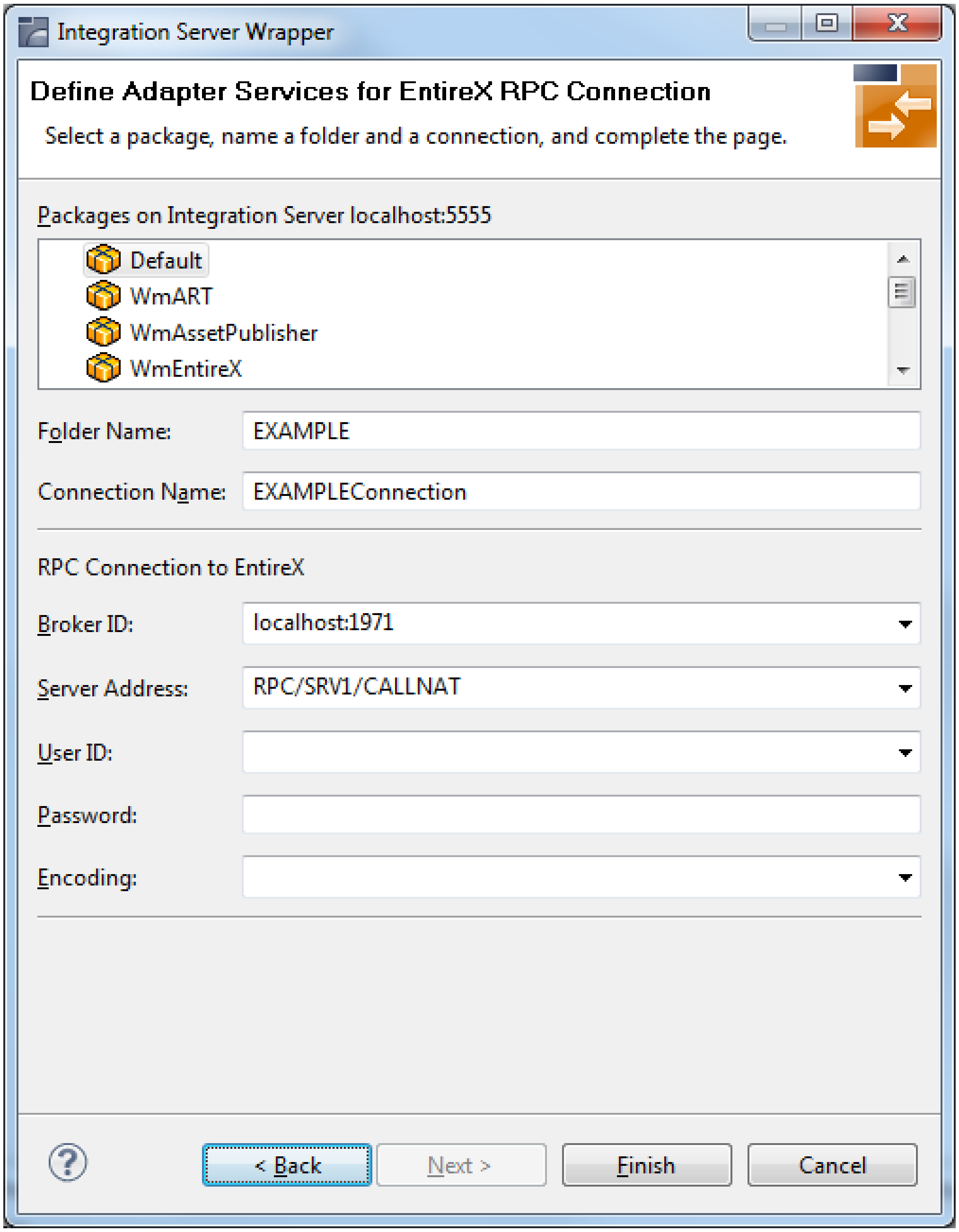
1. Some connection types require a license.
2. The connection types for **Reliable RPC** require an IDL file with `IN` parameters only. If the selected IDL file contains `OUT` parameters, the connection types for **Reliable RPC** are not available.
3. Listeners can be generated only for connection types **RPC**, **Direct RPC** and **Reliable RPC**.

2 Click **Next** and continue with *Step 4a: Define Adapter Services for an RPC Connection*.

Or:

Step 4b: Define Adapter Services for an RPC Listener or a Reliable RPC Listener, depending on the selected connection type.

Step 4a: Define Adapter Services for an RPC Connection



Define Adapter Services for EntireX RPC Connection

Select a package, name a folder and a connection, and complete the page.

Packages on Integration Server localhost:5555

- Default
- WmART
- WmAssetPublisher
- WmEntireX

Folder Name: EXAMPLE

Connection Name: EXAMPLEConnection

RPC Connection to EntireX

Broker ID: localhost:1971

Server Address: RPC/SRV1/CALLNAT

User ID:

Password:

Encoding:

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 Adapter documentation under *webMethods > Mainframe Integration* on the **Software AG Product Documentation** website.

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.

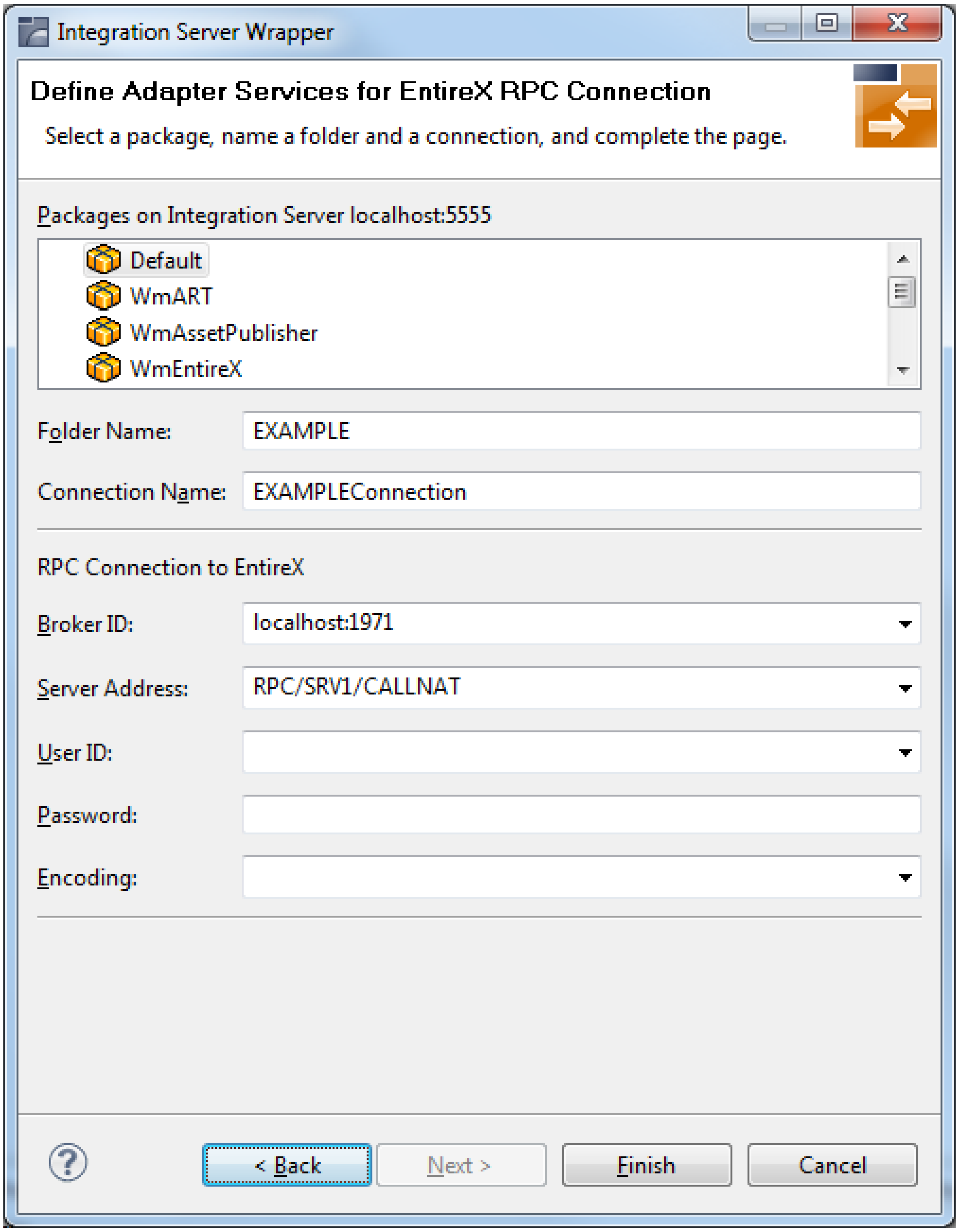
The default settings for new RPC adapter services are:

- 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**

Notes:

1. The check box **Overwrite existing Objects in Integration Server** is useful for re-generating objects created previously. However, you cannot overwrite an RPC Listener Connection or a reliable RPC Listener Connection with a connection of a different type. If the connection is deleted with the Adapter Administration UI, it is not possible to overwrite the objects. In this case, you have to delete the adapter services in the Designer.
2. 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: Define Adapter Services for an RPC Listener or a Reliable RPC Listener



Integration Server Wrapper

Define Adapter Services for EntireX RPC Connection

Select a package, name a folder and a connection, and complete the page.

Packages on Integration Server localhost:5555

- Default
- WmART
- WmAssetPublisher
- WmEntireX

Folder Name: EXAMPLE

Connection Name: EXAMPLEConnection

RPC Connection to EntireX

Broker ID: localhost:1971

Server Address: RPC/SRV1/CALLNAT

User ID:

Password:

Encoding:

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 Adapter documentation under *webMethods > Mainframe Integration* on the **Software AG Product Documentation** website.

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 new RPC 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**

Notes:

1. The check box **Overwrite existing Objects in Integration Server** is useful for re-generating objects created previously. However, you cannot overwrite an RPC Listener Connection or a reliable RPC Listener Connection with a connection of a different type. If the connection is deleted with the Adapter Administration UI, it is not possible to overwrite the objects. In this case, you have to delete the adapter services in the Designer.
2. 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 *Mapping Software AG IDL to Integration Server Data Types*.

3 Mapping Software AG IDL to Integration Server Data Types

- All primitive data types of the Software AG IDL (except B and BV) are mapped to `java.lang.String`.
- Only data types B and BV (with or without maximum length) are mapped to `byte[]`.
- With data types N, NU, P, and PU, 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.
- Data types A, K, and U (for fixed length) 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.
- Groups are mapped to documents.
- One dimensional arrays are mapped to String lists (`java.lang.String[]`).
- Two dimensional arrays are mapped to String tables (`java.lang.String[][]`).
- Three dimensional arrays are mapped to `java.lang.String[][][]`.
- Data type D: the format of the string in the pipeline is `java.text.DateFormat.getDateInstance(DateFormat.MEDIUM, Locale.ENGLISH).format(date))`, where *`date`* is of type `java.util.Date`.
- Data type T: the format of the string in the pipeline is `java.text.DateFormat.getDateTimeInstance(DateFormat.MEDIUM, DateFormat.MEDIUM, Locale.ENGLISH).format(date))`, where *`date`* is of type `java.util.Date`.

4 Handling SVM Files

A server mapping file (SVM) enables the RPC server to correctly support special COBOL syntax such as `REDEFINE`s, `JUSTIFIED`, `SYNCHRONIZE` and `OCCURS DEPENDING ON` clauses, `LEVEL-88` fields, etc. If one of these elements is used, the EntireX Workbench automatically extracts an SVM file in addition to the IDL (interface definition language), or an SVM file is generated by the COBOL Wrapper for a server skeleton. The SVM file is used at runtime to marshal and unmarshal the RPC data stream.

SVM Files in the EntireX Workbench

In the *EntireX Workbench*, an SVM file has to relate to an appropriate IDL file. Therefore, you always have to keep the IDL file and the SVM file together in the same folder.

If there is an SVM file and a corresponding IDL file,

- at least one of the IDL programs in the corresponding IDL file requires server-mapping information to correctly call the target server. For those IDL programs, there is an SVM entry (line) in the Workbench SVM file.
- deployment of the SVM file to the RPC server is mandatory, see *Server Mapping Deployment*.

If there is an IDL file but no corresponding SVM file,

- there is no IDL program that requires server mapping information.

SVM Files in EntireX Adapter

SVM files may be used for IMS Connect and CICS ECI connections only. The SVM files are wrapped into the Integration Server adapter service itself. For all connections except ACI Server connections the SVM files need to de deployed to the remote RPC server. See *Server Mapping Deployment*.

Do not change the location of the generated SVM file. It has to be kept in the same folder as the IDL file and will be picked up automatically together with the IDL file when an adapter connection for IMS Connect or CICS ECI is generated.

Source Control of SVM Files

Because SVM entries within an SVM file contain text data only, a Workbench SVM file is text-based (although it is not intended for human consumption). Therefore, you can include it in your source control management together with the IDL file and the COBOL source(s) as a triplet that should always be kept in sync.

Compare SVM Files

For SVM files in the *EntireX Workbench* format, you can use a third party file/text compare tool to check if two files are identical.

The SVM entries (corresponding to lines in a Workbench SVM file) contain a creation timestamp at offset 276 (decimal) in the format `YYYYMMDDHHIISST`. The precision is 1/10 of a second.

When is an SVM File Required?

For the IDL Extractor for COBOL

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
CICS with DFHCOMMAREA Calling Convention and `IN` different to `OUT`	all		yes	*CICS with `DFHCOMMAREA` Calling Convention* under *Introduction to the IDL Extractor for COBOL* | *CICS `DFHCOMMAREA`* under *COBOL Parameter Selection*
CICS Channel Container Calling Convention	all		yes	*CICS with Channel Container Calling Convention*
CICS with DFHCOMMAREA Large Buffer Interface	all		yes	*CICS with `DFHCOMMAREA` Large Buffer Interface*
IMS MPP Message Interface (IMS Connect)	all		yes	*IMS MPP Message Interface (IMS Connect)*
IMS BMP with Standard Linkage Calling Convention	all		yes	*IMS BMP with Standard Linkage Calling Convention*

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
Micro Focus with Standard Linkage Calling Convention	`BINARY` clause		yes	*Micro Focus with Standard Linkage Calling Convention*
all	`OCCURS DEPENDING ON` clause		yes	*Tables with Variable Size - `DEPENDING ON` Clause* under *COBOL to IDL Mapping* in the IDL Extractor for COBOL documentation
all	`REDEFINES` clause		yes	*`REDEFINE` Clause*
all	`TRAILING [SEPARATE]` clause		yes	*`SIGN LEADING` and `TRAILING SEPARATE` Clause*
all	`LEADING [SEPARATE]` clause		yes	*`SIGN LEADING` and `TRAILING SEPARATE` Clause*
all	`ALIGNED RIGHT` attribute		yes	
all	all	Rename of program	yes	*The Software AG IDL Tree Pane* under *Mapping Editor User Interface* in the IDL Extractor for COBOL documentation
all	all	Map to operation	yes	*Context Menu* under *The COBOL Parameters Pane*
all	all	Map to constant	yes	*Context Menu*
all	all	Suppress	yes	*Context Menu*
other combinations			no	

For the COBOL Wrapper

This depends on the interface type chosen and the IDL type:

Interface Type	IDL Type	COBOL Wrapper	SVM Required	More Information
CICS with DFHCOMMAREA Large Buffer Interface	all		yes	*CICS with DFHCOMMAREA Large Buffer Interface* under *COBOL Server Interface Types*
CICS with Channel Container Calling Convention	all		yes	*CICS with Channel Container Calling Convention*

Interface Type	IDL Type	COBOL Wrapper	SVM Required	More Information
IMS BMP with Standard Linkage Calling Convention	all		yes	*IMS BMP with Standard Linkage Calling Convention*
Micro Focus	I2 or I4		yes	*Micro Focus with Standard Linkage Calling Convention* | *IDL Data Types* under *Software AG IDL File* in the IDL Editor documentation
all	IDL unbounded array		yes	*`array-definition`* under *Software AG IDL Grammar* in the IDL Editor documentation
all	IDL unbounded group		yes	`group-parameter-definition` under *Software AG IDL Grammar* in the *IDL Editor* documentation
all	all	IDL program name is not a valid COBOL name and is therefore adapted, or the COBOL program name is customized	yes	*Customize Automatically Generated Server Names*
other combinations			no	

Is There a Way to Smoothly Introduce SVM Files?

All EntireX RPC servers can be executed without SVM files. There is no need to install the SVM container (see ***SVM Files in EntireX Adapter***) as long as you do not use features that require SVM files (see ***When is an SVM File Required?***). You can also call COBOL servers generated or extracted with previous versions of EntireX mixed with a COBOL server that requires SVM files. All EntireX RPC servers are backward compatible.

5 Integration Server Preferences

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.

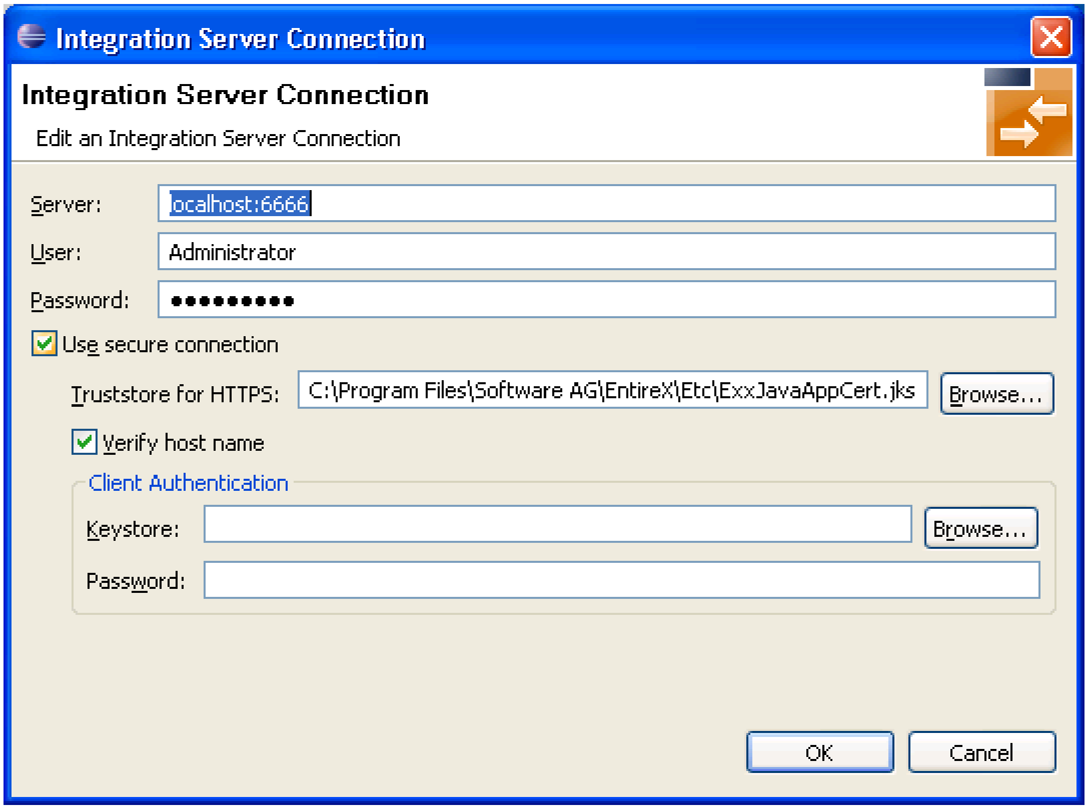
Integration Server Connections

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
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:

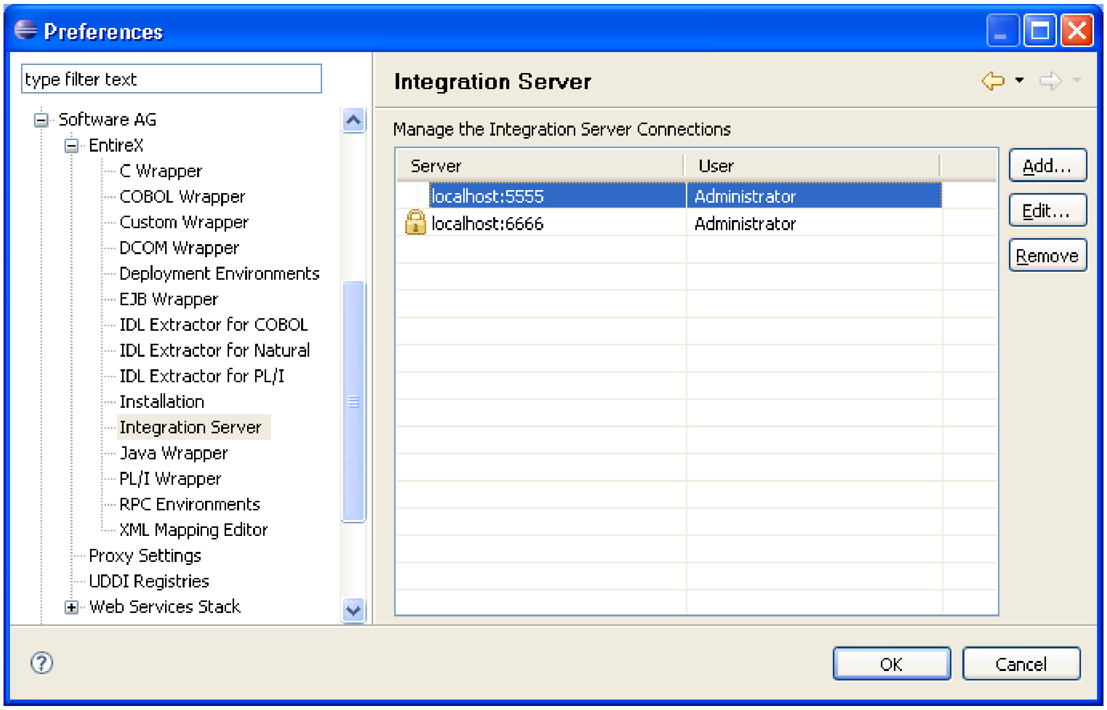
Password:

Browse...
Browse...
OK
Cancel

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 *Integration Server Preferences*.

Setting Integration Server Preferences



Manage the Integration Server Connections

Server	User
localhost:5555	Administrator
localhost:6666	Administrator

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.