Using the IDL Extractor for Integration Server

This chapter describes following sections:

- Step 1: Start the IDL Extractor for Integration Server

- Step 2a: Connect EntireX Workbench with Integration Server

- Step 2b: Connect EntireX Workbench with Integration Server using an Existing Connection

- Step 3: Select the Integration Server Package to Extract

- Step 4: Select the Connection Type

- Step 5: Define a Listener

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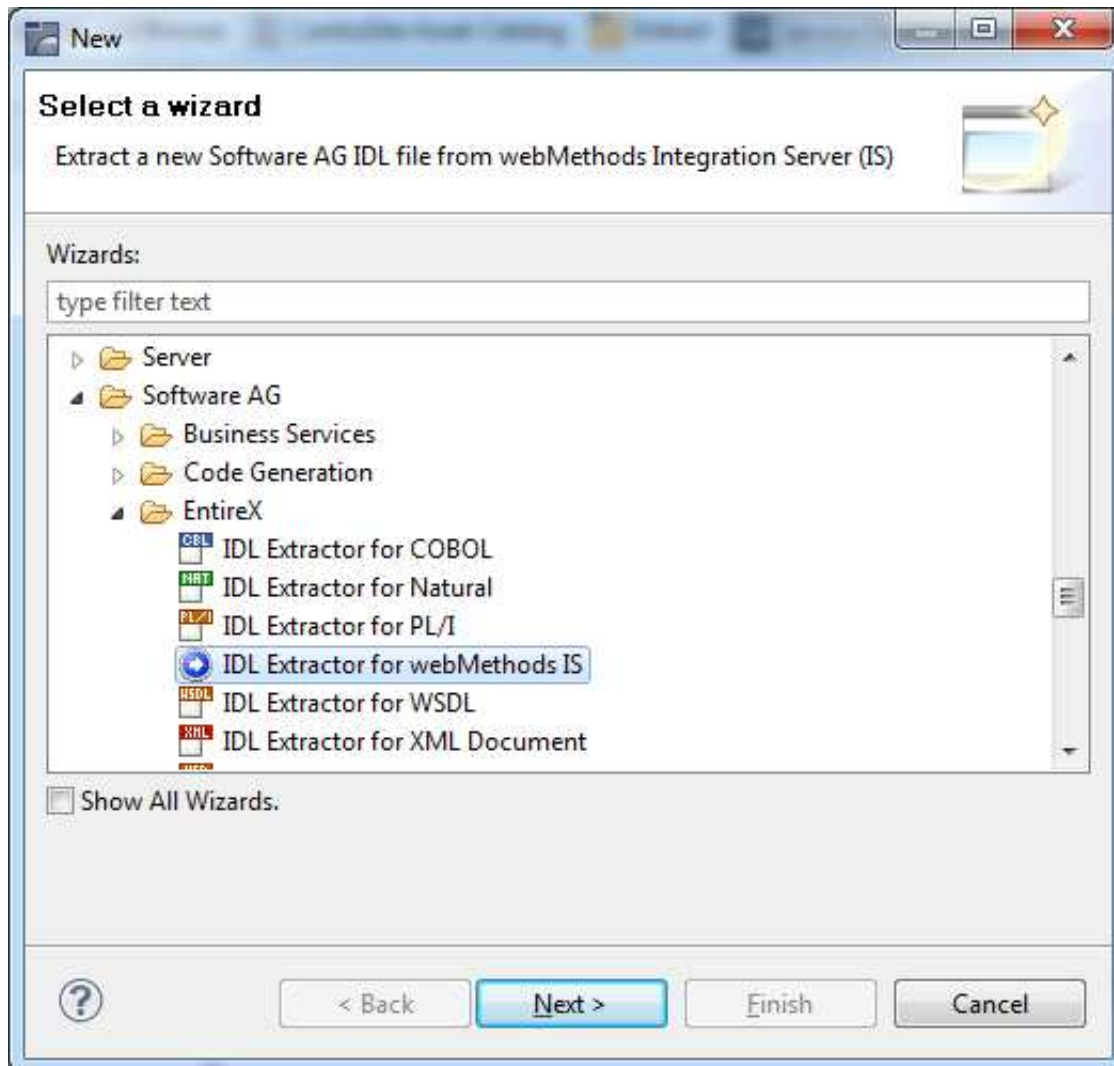
   Note:
   You can also chose **New** from the toolbar or context menu of a view showing resources. Also Ctrl-N starts the selection of the *New Wizards*.

2. If you are using the wizard for the first time without any predefined Integration Server connections, continue with *Step 2a: Connect EntireX Workbench 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 EntireX Workbench with Integration Server using an Existing Connection*.

Step 2a: Connect EntireX Workbench 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**.
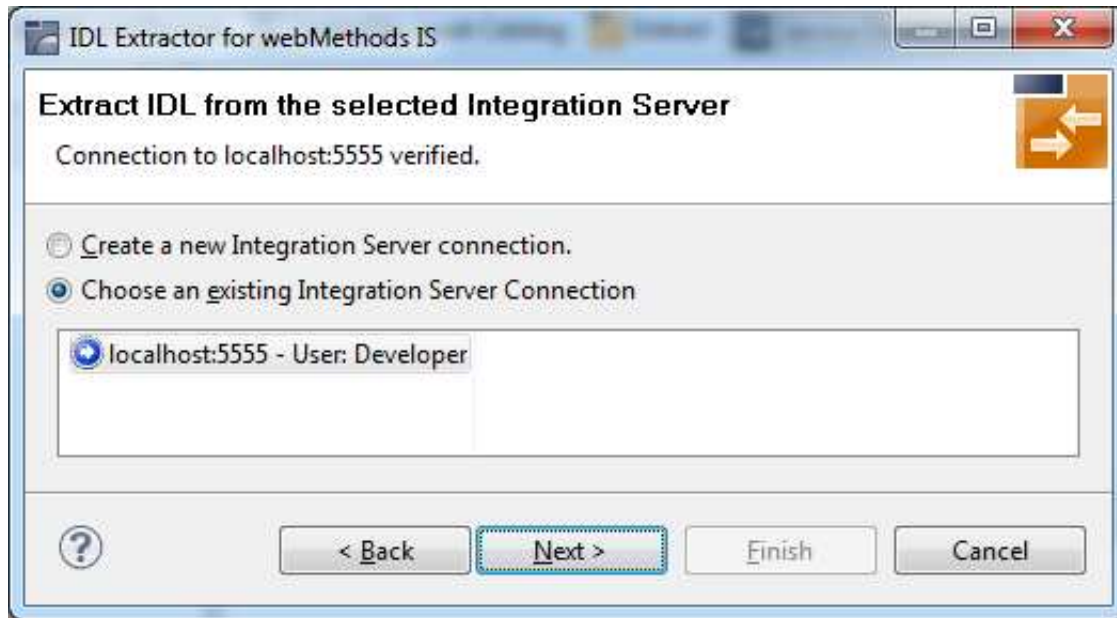   5. For managing Integration Server connections, see *Integration Server Preferences*.

2. Choose **Next** and continue with *Step 3: Select the Integration Server Package to Extract*.

Step 2b: Connect EntireX Workbench 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**

1. Select **Create a new Integration Server Connection** and continue with *Step 2a: Connect EntireX Workbench with Integration Server*.

Step 3: Select the Integration Server Package to Extract
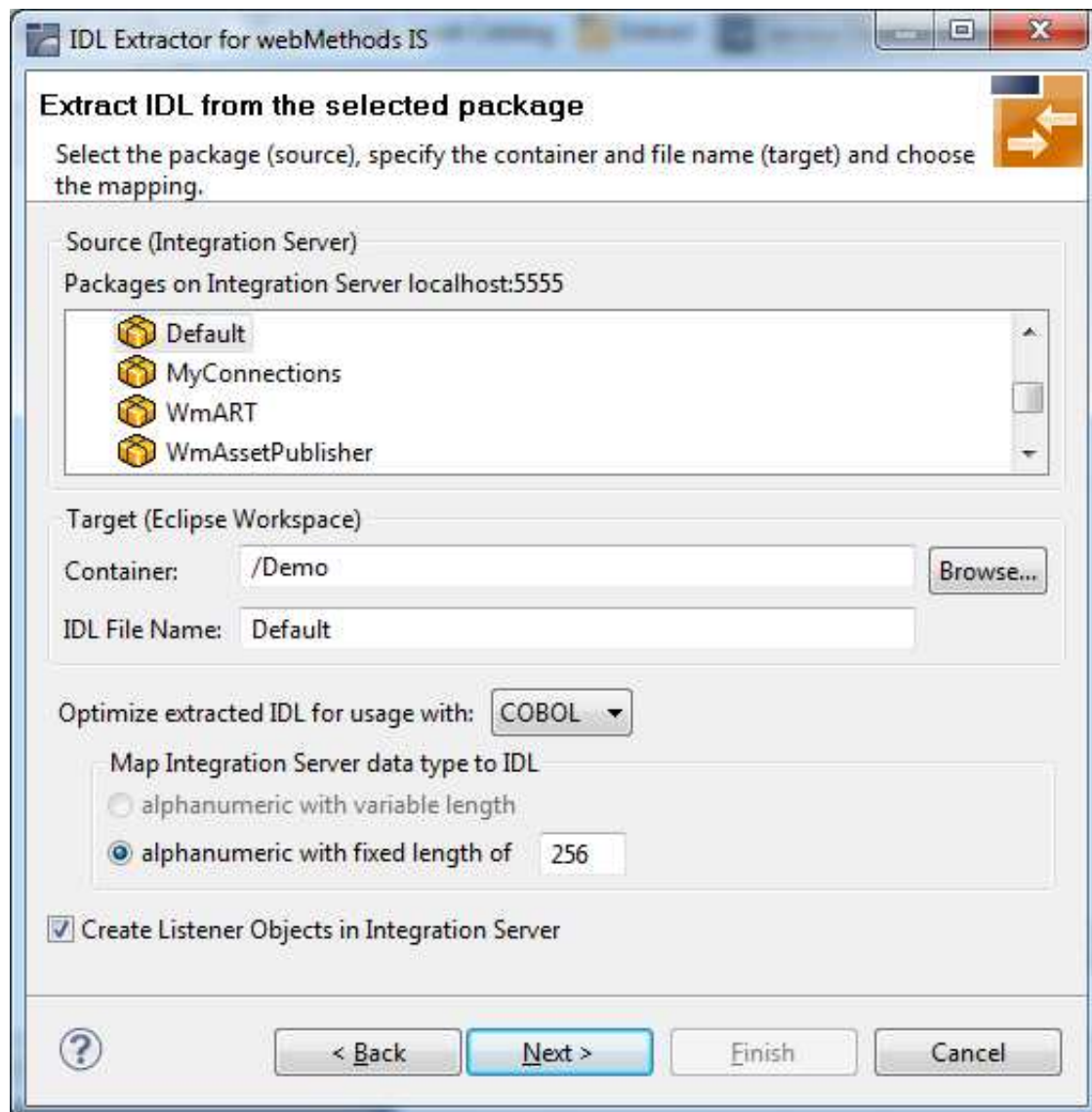
> **To extract the IDL from the selected package**

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 4b: Define Adapter Services for an RPC Listener or a Reliable RPC 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: Select the Connection Type
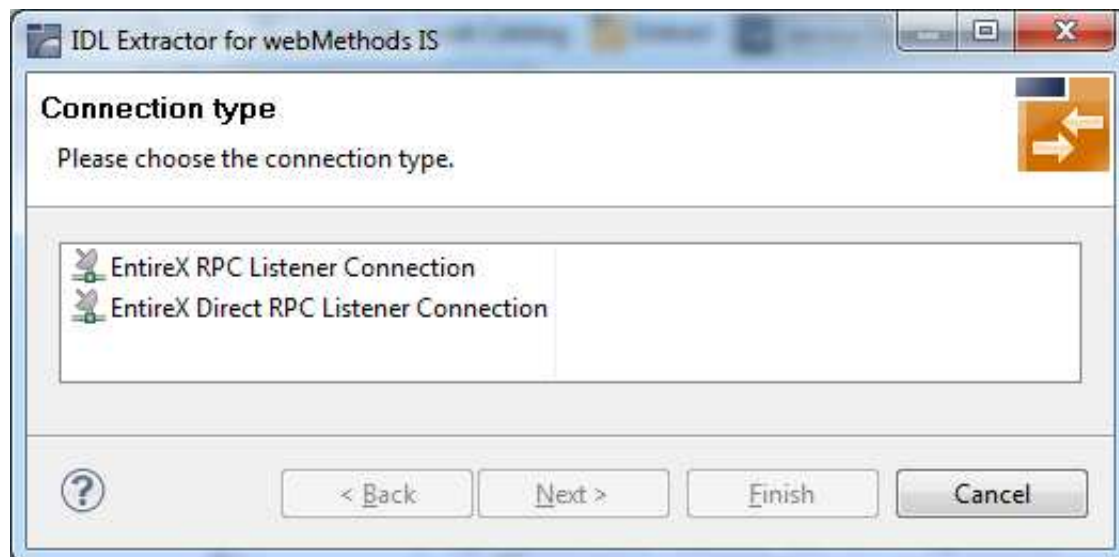

> **To select the connection type**

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

● 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.

Step 5: Define a Listener
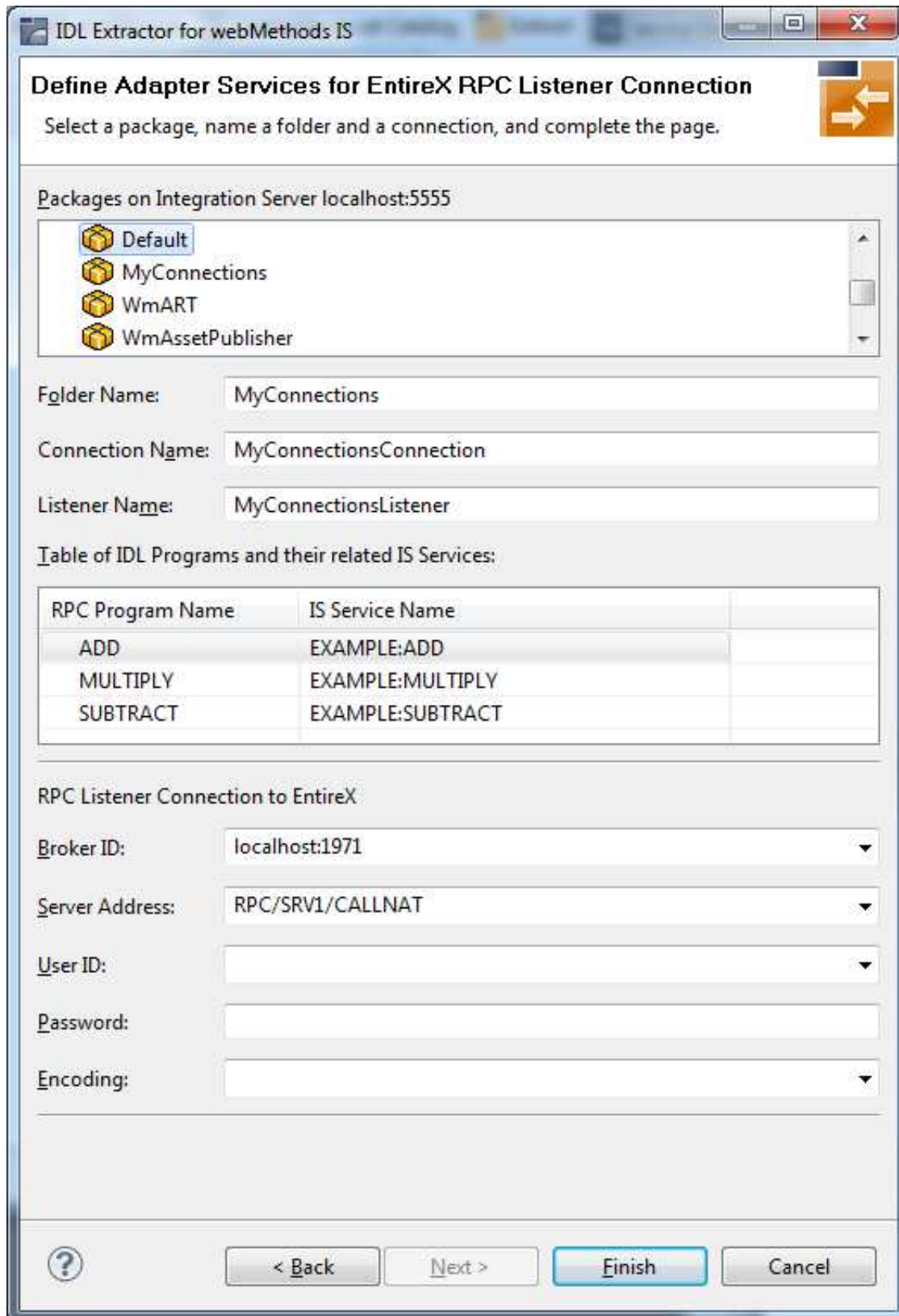
≫   **To define a listener**

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 name`Connection) and **Listener Name** (default: `library name`Listener).

3. If necessary, edit the Broker settings for the Listener (**Broker ID**, **Server Address**, **User ID**, **Password**, and **Encoding**).

4. The check box **Overwrite existing Objects in Integration Server** can be used to re-generate the objects after you have changed the IDL file.

    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.

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