webMethods EntireX

Common Integration Scenarios

Version 9.5 SP1

November 2013

This document applies to webMethods EntireX Version 9.5 SP1.

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

Document ID: EXX-EEXXSCENARIOS-95SP1-20140628

Table of Contents

Common Integration Scenarios

Connecting Natural	You have a Natural server and want to call this from a Web service client or from an Integration Server service.
Connecting Integration Server	You want to call the Integration Server Listener from a Natural or COBOL application.
Connecting COBOL	You have a COBOL server and want to call this from a Web service client or Integration Server service.
Connecting Web Services	You want to call a Web service from a Natural or COBOL application.

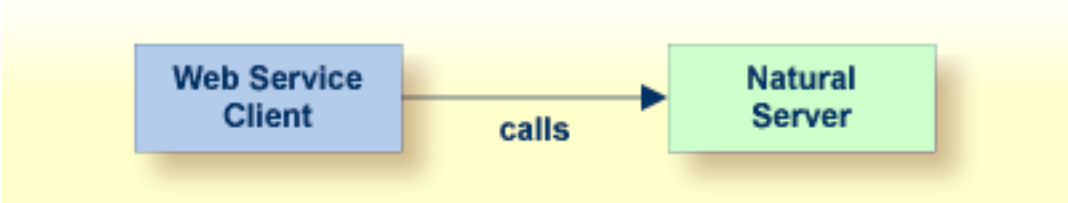
I Connecting Natural

Calling Natural from a Web Service
Calling Natural from Integration Server

See also *Calling Integration Server from Natural* | *Calling a Web Service from Natural*.

1 Calling Natural from a Web Service

Scenario: “I have a Natural server and I want to call this from a Web service client.”

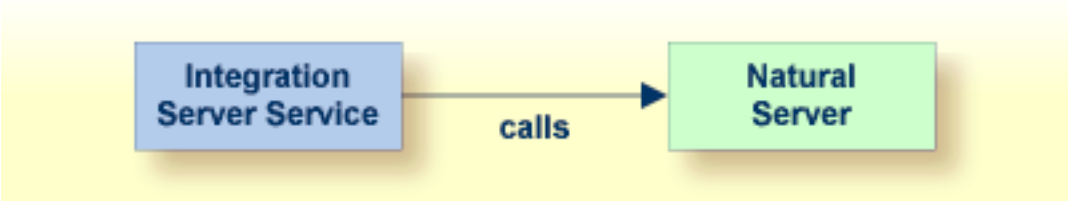


Solution: Select an existing Natural server (1) and generate the integration logic (2) to call it from a Web service client (3). See also the steps below.

1. Extract the interface of a Natural server. See *Using the Software AG IDL Extractor for Natural*.
2. Generate Web service client objects. See *Using the EntireX Web Services Wrapper*.
3. Test call from Web service client to Natural server.

2 Calling Natural from Integration Server

Scenario: “I have a Natural server and I want to call this from the Integration Server platform.”



Solution: Select an existing Natural server (1) and generate the integration logic (2) to call it from IS platform (3). See also the steps below.

1. Extract the interface of a Natural server. See *Using the Software AG IDL Extractor for Natural*.
2. Generate Integration Server adapter service and adapter connections. See *Using the Integration Server Wrapper*. As an alternative, perform steps (1) and (2) using an integrated wizard. See *Adapter Services Wrapper for Natural* in the EntireX documentation.
3. Test call from Integration Server service to Natural server.

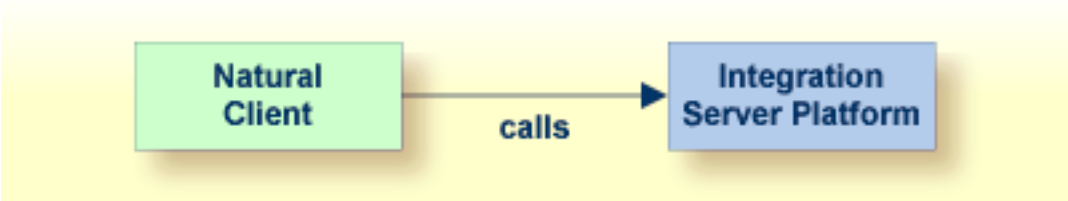
II Connecting Integration Server

- *Calling Integration Server from Natural*
- *Calling Integration Server from COBOL*

See also *Calling Natural from Integration Server* | *Calling COBOL from Integration Server*.

3 Calling Integration Server from Natural

Scenario: “I want to call the Integration Server listener from a Natural application.”

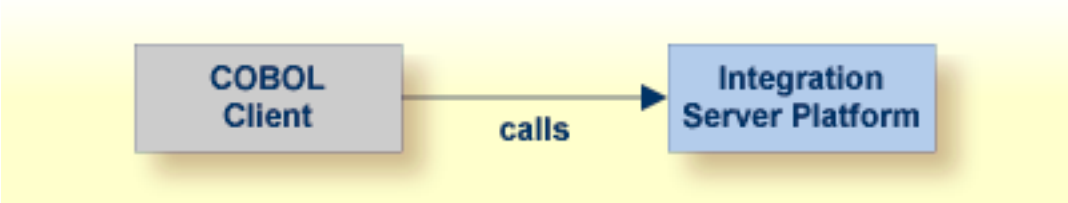


Solution: Select an existing IS service (1) and generate the integration logic (2) to call it from a Natural application (3). See also the steps below.

1. Read package from Integration Server and generate Integration Server adapter services and listeners. See *Using the IDL Extractor for Integration Server*.
2. Generate objects for Natural client application. See *Natural Wrapper*.
3. Test call from Natural client to Integration Server listener. See *Sample Generation Result for the Client Side* under *Using the Natural Wrapper*.

4 Calling Integration Server from COBOL

Scenario: “I want to call the Integration Server listener from a COBOL application.”



Solution: Select an existing IS service (1) and generate the integration logic (2) to call it from a COBOL application (3). See also the steps below.

1. Read package from Integration Server and generate Integration Server adapter services and listeners. See *Using the IDL Extractor for Integration Server*.
2. Generate objects for COBOL client application. See *COBOL Wrapper*.
3. Test call from COBOL client to Integration Server listener.

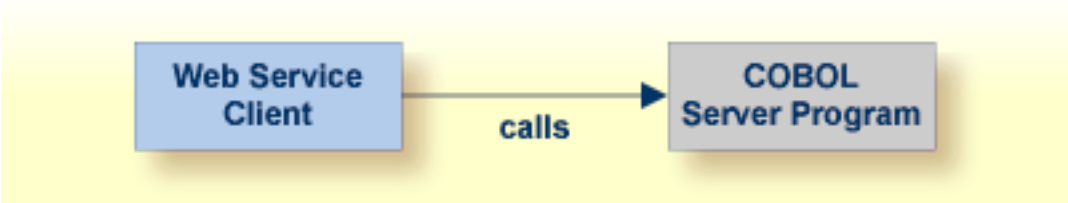
III Connecting COBOL

- *Calling COBOL from a Web Service*
- *Calling COBOL from Integration Server*

See also *Calling a Web Service from COBOL* | *Calling Integration Server from COBOL*.

5 Calling COBOL from a Web Service

Scenario: “I have a COBOL server program and I want to call this from a Web service client.”

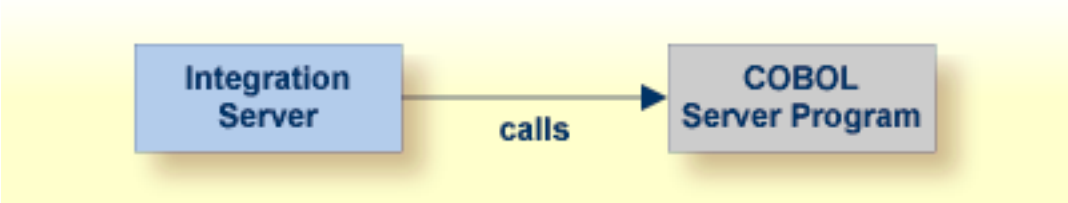


Solution: Select an existing COBOL server program (1) and generate the integration logic (2) to call it from a Web service client (3). See also the steps below.

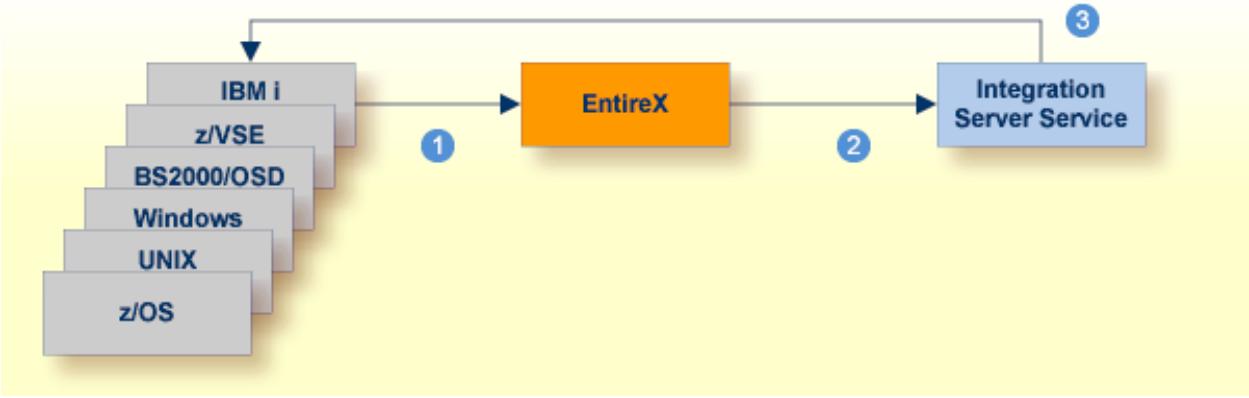
1. Extract the interface of the COBOL server program. See *Using the IDL Extractor for COBOL - Overview*.
2. Generate Web service client objects. See *Using the EntireX Web Services Wrapper*.
3. Test call from Web service client to COBOL server program.

6 Calling COBOL from Integration Server

Scenario: "I have a COBOL server program and I want to call this from the Integration Server platform."



Solution: Take an existing COBOL server (1) and generate the integration logic (2) to call it from the IS platform (3).



The COBOL server can be called from different operating systems. Continue with the appropriate scenario:

- *Calling COBOL on z/OS from Integration Server*
- *Calling COBOL on UNIX from Integration Server*

- *Calling COBOL on Windows from Integration Server*
- *Calling COBOL on BS2000/OSD from Integration Server*
- *Calling COBOL on z/VSE from Integration Server*
- *Calling COBOL on IBM i from Integration Server*

7 Calling COBOL on z/OS from Integration Server

Under z/OS, a COBOL server can be called in different environments:

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

Continue with the appropriate scenario:

- *Calling COBOL on z/OS CICS from Integration Server*
- *Calling COBOL on z/OS IMS from Integration Server*
- *Calling COBOL on z/OS Batch from Integration Server*

8 Calling COBOL on z/OS CICS from Integration Server

There are different styles (interface types) for calling a COBOL server.

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

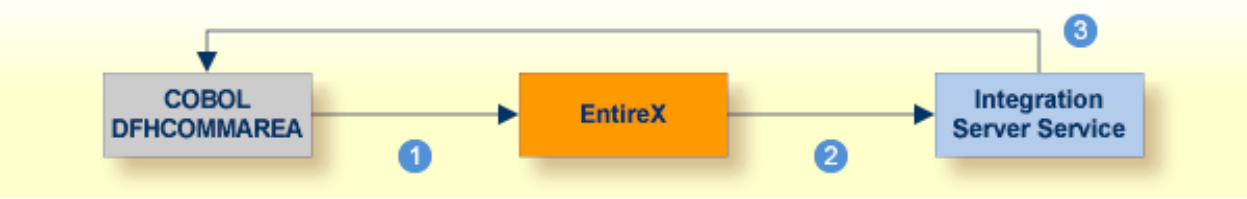
It is important to know the interface type of your COBOL server. If you are unsure, consult a COBOL CICS specialist or see description of interface type in the IDL Extractor for COBOL documentation for details and examples: `DFHCOMMAREA` *Calling Convention* | *Channel Container Calling Convention* | `DFHCOMMAREA` *Large Buffer Interface*.

When you are sure which interface type you are using, continue with the appropriate scenario:

- *Calling COBOL DFHCOMMAREA on z/OS CICS from Integration Server*
- *Calling COBOL Channel Container on z/OS CICS from Integration Server*
- *Calling COBOL Large Buffer on z/OS CICS from Integration Server*

9 Calling COBOL DFHCOMMAREA on z/OS CICS from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL DFHCOMMAREA server. For illustration and examples on such a server, see *CICS with* `DFHCOMMAREA` *Calling Convention*.
- You have access to the related COBOL sources and copybooks. The minimum requirement is the `DATA DIVISION` of the interface. The sources and copybooks must be files on your PC or remotely stored in PDS or CA Librarian data set and accessed via the *Batch RPC Server* (see the Batch RPC Server documentation).
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the CICS RPC Server see *CICS RPC Server*
  - For the *EntireX Direct RPC* connection method you need the CICS RPC Server, see *CICS RPC Server*
  - For the *EntireX CICS ECI* connection method you need to configure the CICS ECI TCP/IP service within your CICS region. See *Preparing IBM CICS for ECI* in the webMethods EntireX Adapter documentation.

1: Extract the Interface of a COBOL Server

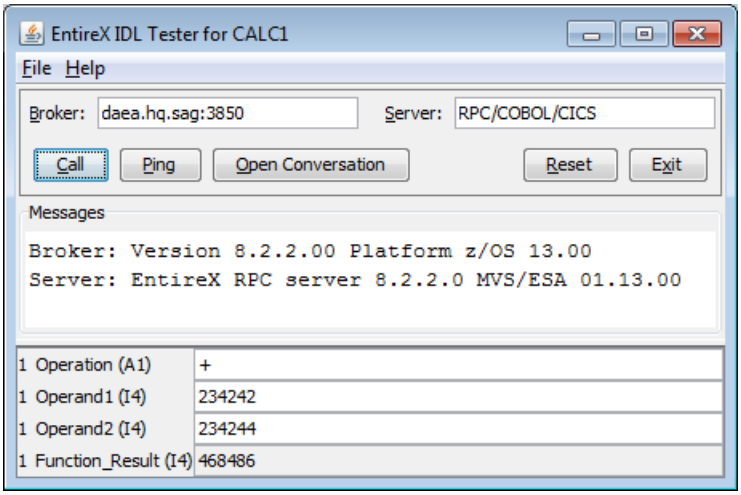
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

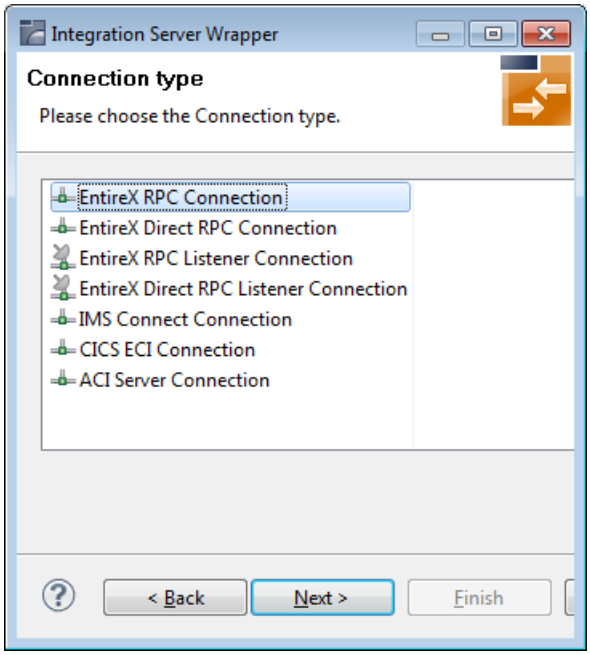


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

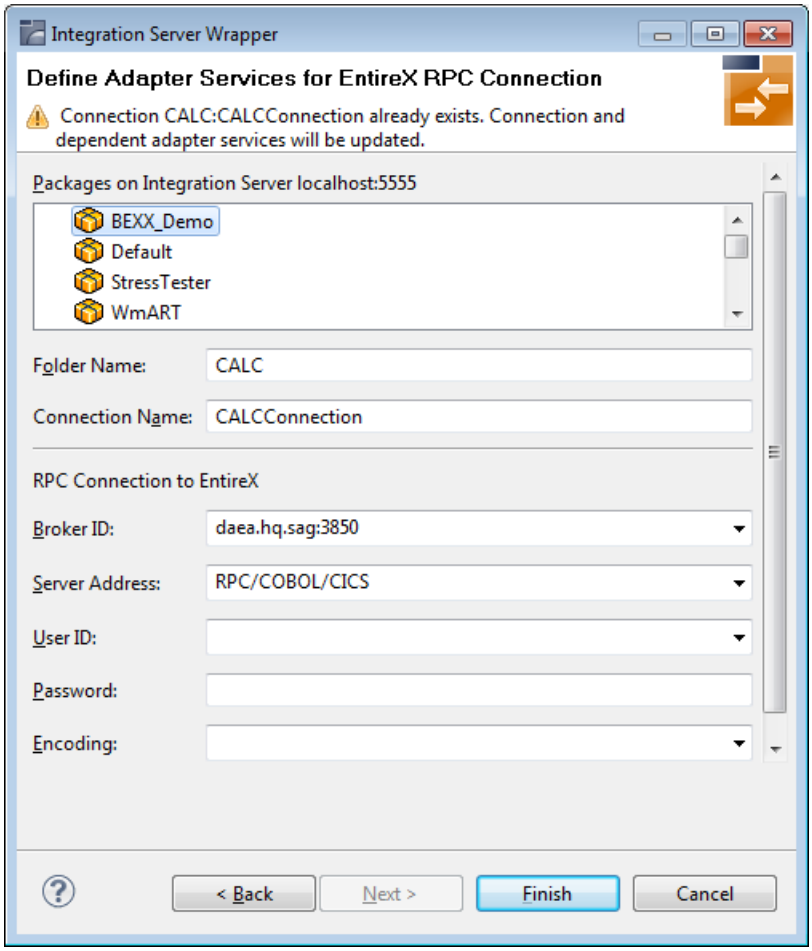
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



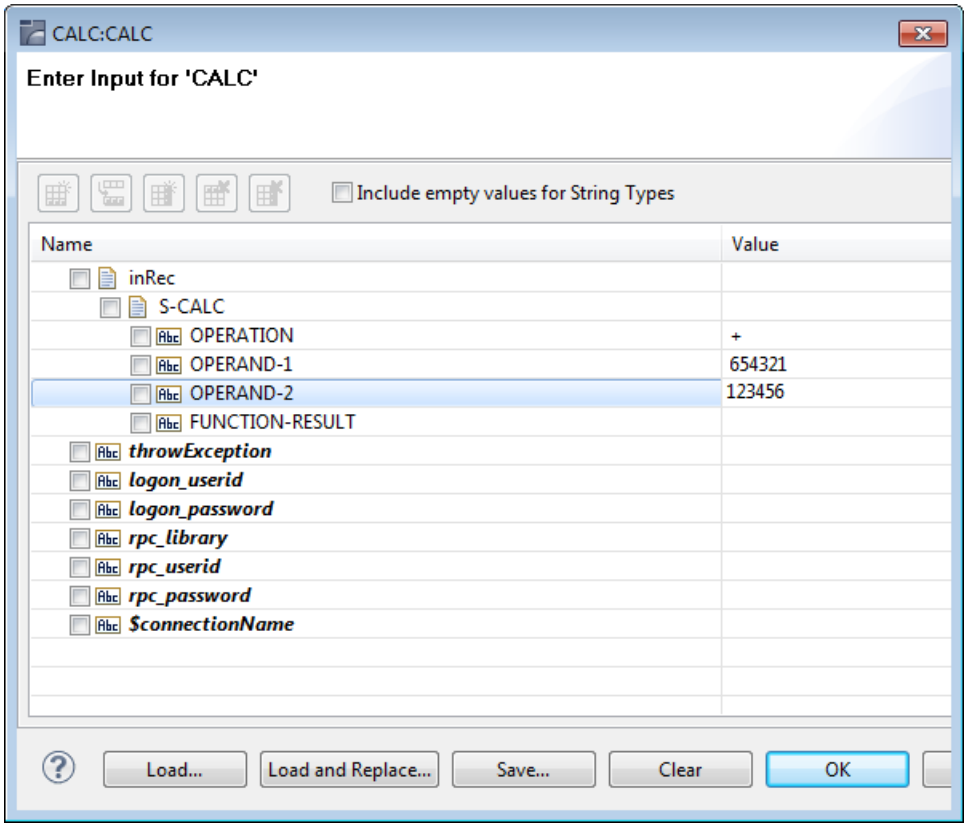
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:

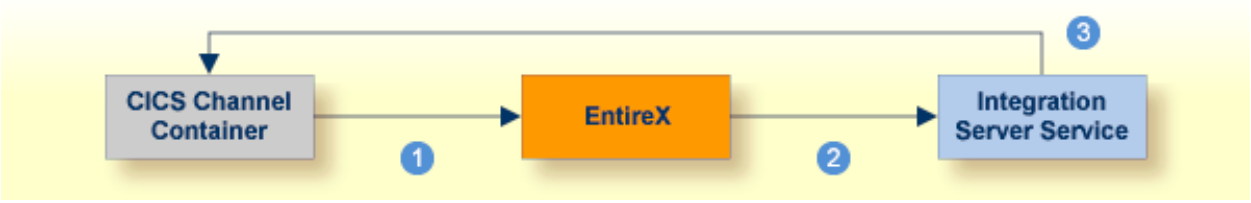


This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

10 Calling COBOL Channel Container on z/OS CICS from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL Channel Container server. For illustration and examples on such a server, see *CICS with Channel Container Calling Convention*.
- You have access to the related COBOL sources and copybooks. The minimum requirement is the `DATA DIVISION` of the interface. The sources and copybooks must be files on your PC or remotely stored in PDS or CA Librarian data set and accessed via the *Batch RPC Server* (see the Batch RPC Server documentation).
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the CICS RPC Server see *CICS RPC Server*
  - For the *EntireX Direct RPC* connection method you need the CICS RPC Server, see *CICS RPC Server*

1: Extract the Interface of a COBOL Server

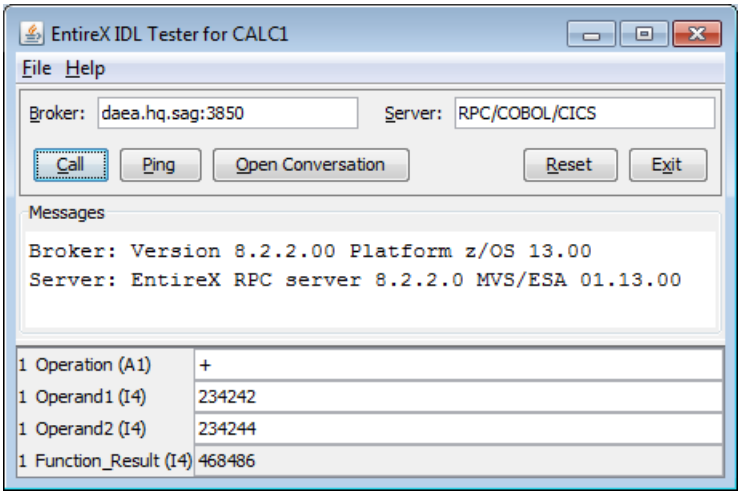
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

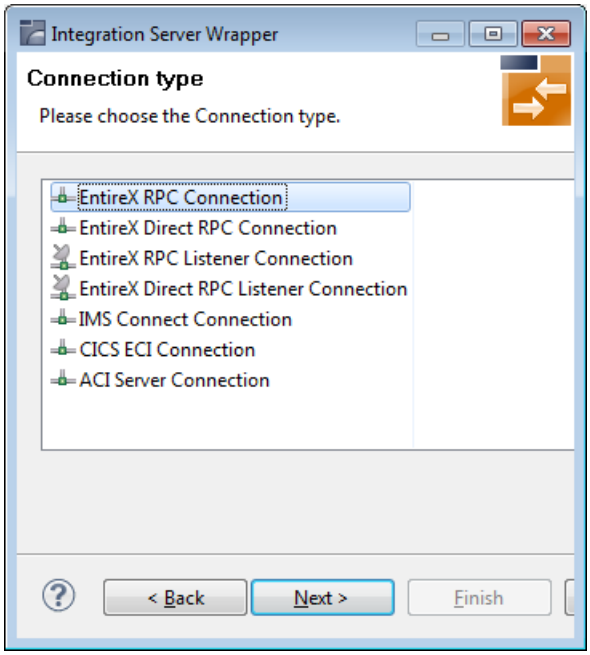


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

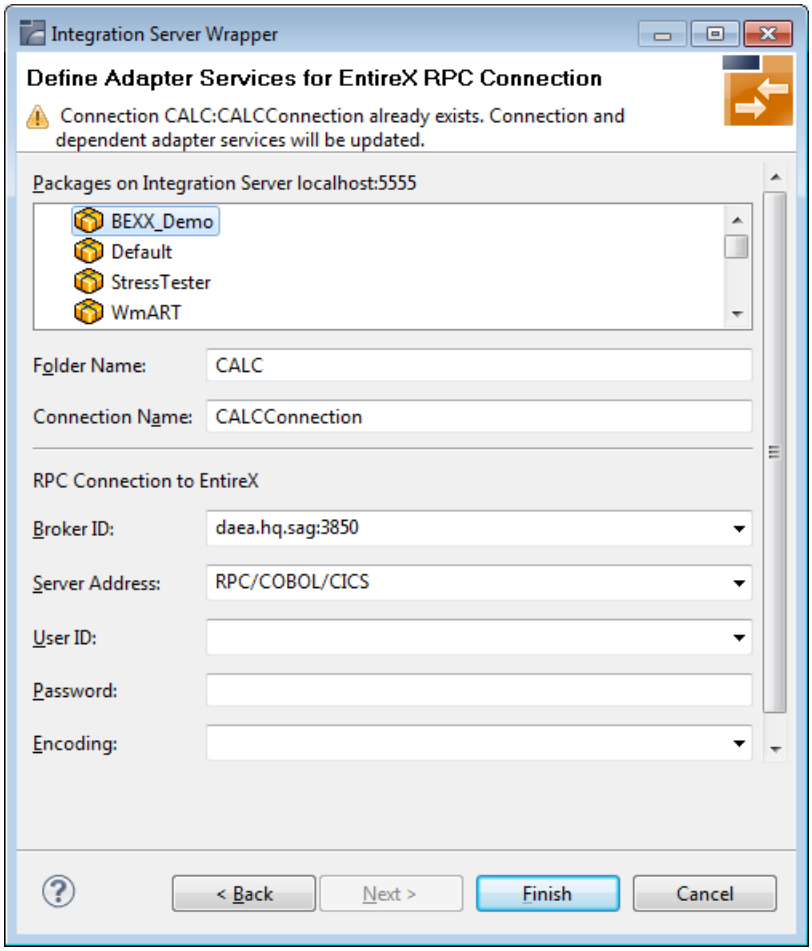
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



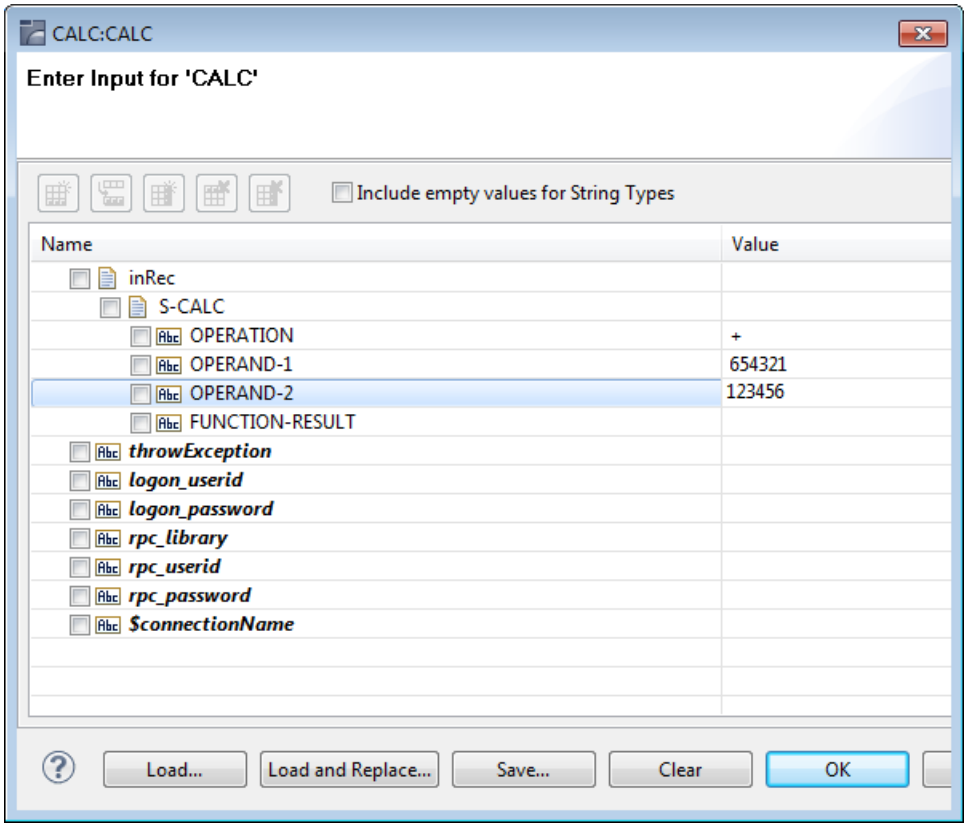
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:

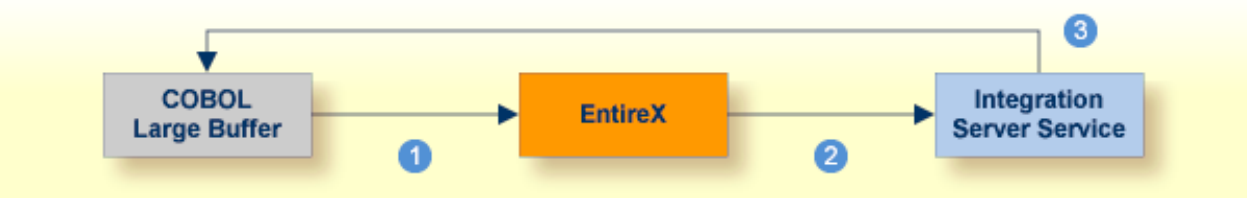


This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

11 Calling COBOL Large Buffer on z/OS CICS from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL Large Buffer server. For illustration and examples on such a server, see *CICS with* `DFHCOMMAREA` *Large Buffer Interface*.
- You have access to the related COBOL sources and copybooks. The minimum requirement is the `DATA DIVISION` of the interface. The sources and copybooks must be files on your PC or remotely stored in PDS or CA Librarian data set and accessed via the *Batch RPC Server* (see the Batch RPC Server documentation).
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the CICS RPC Server see *CICS RPC Server*
  - For the *EntireX Direct RPC* connection method you need the CICS RPC Server, see *CICS RPC Server*

1: Extract the Interface of a COBOL Server

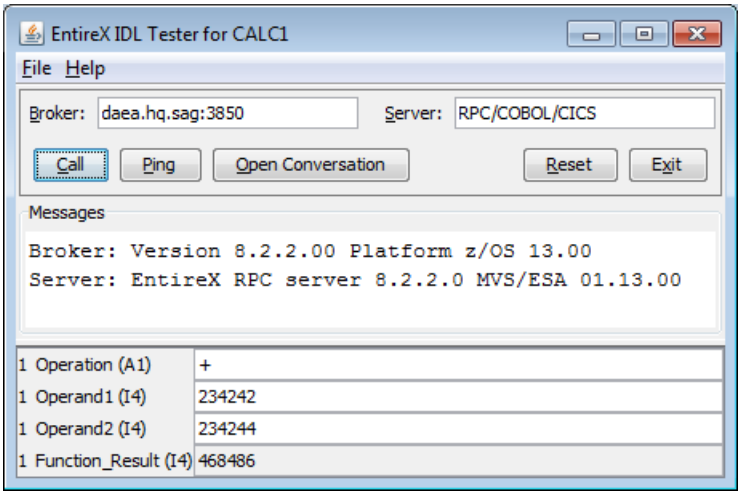
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

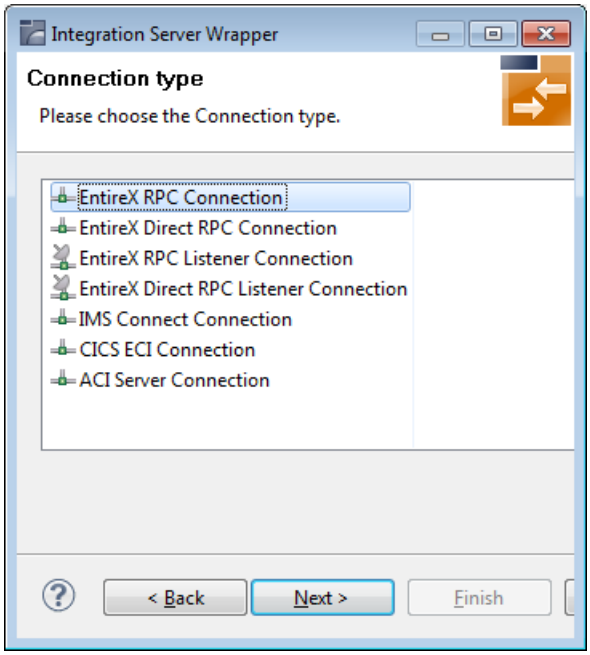


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

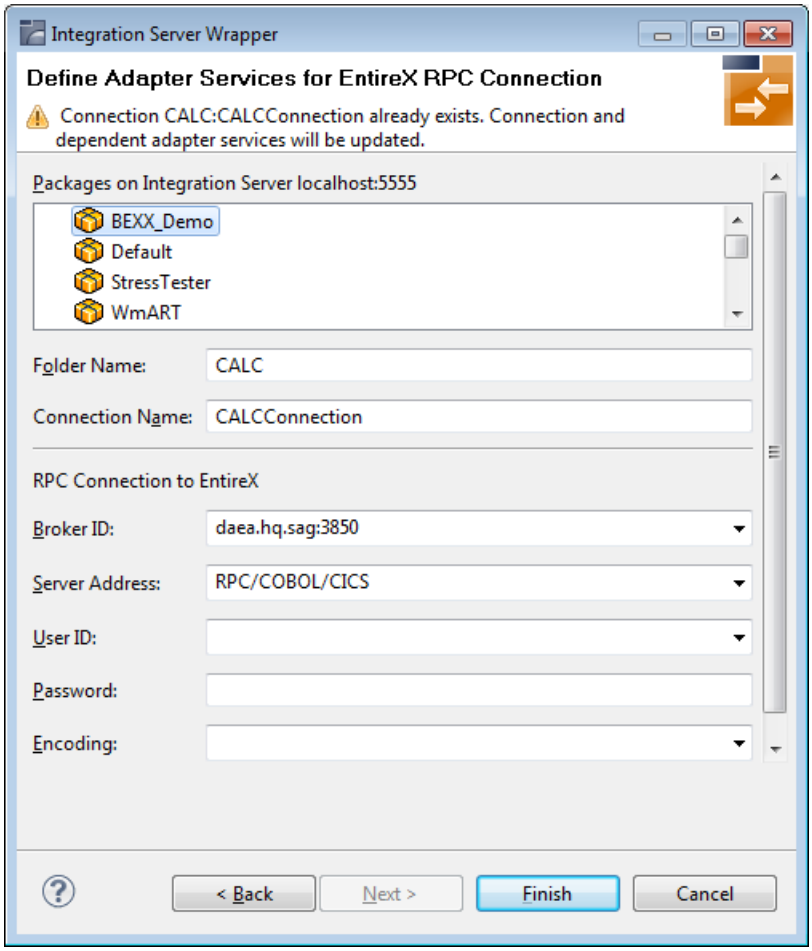
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



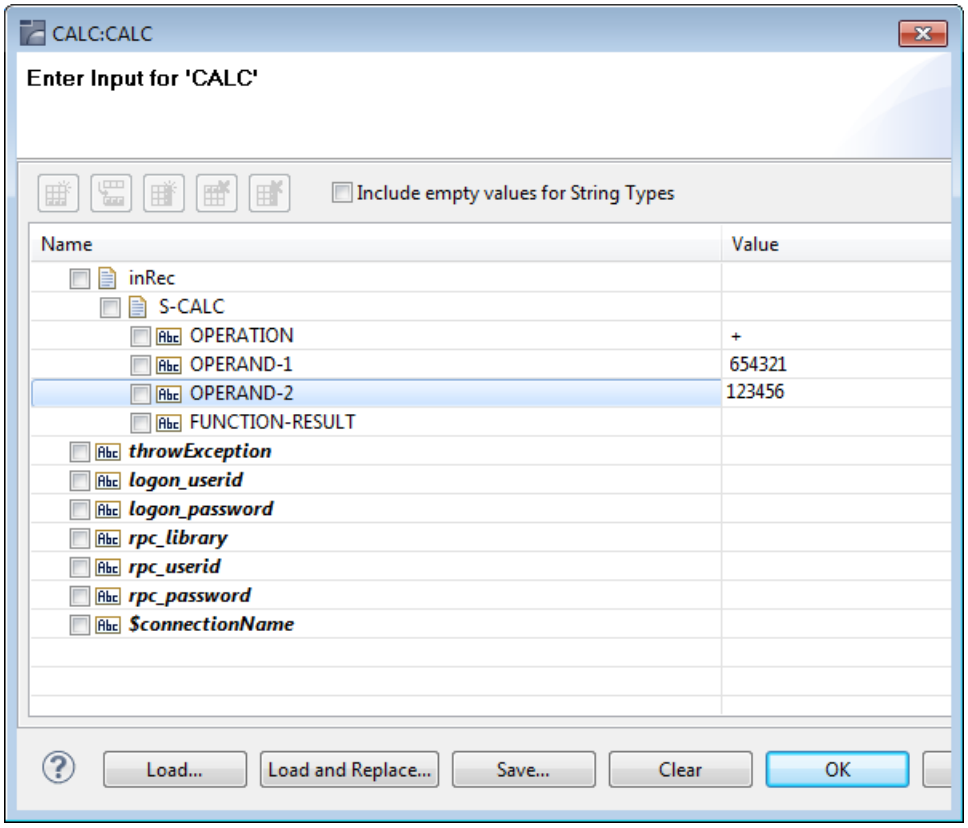
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

12 Calling COBOL on z/OS IMS from Integration Server

In IMS there are two styles of programs: message processing programs (MPP) and batch message processing programs (BMP). MPP programs can be called via IMS Connect.

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

It is important to know whether your COBOL server runs in MPP (online) mode or BMP (batch) mode. If you are unsure, consult a COBOL IMS specialist or see description of interface type in the IDL Extractor for COBOL documentation for details and examples: *IMS BMP | IMS MPP*.

When you are sure which programing style you are using, continue with the appropriate scenario:

- *Calling COBOL on z/OS IMS MPP (IMS Connect) from Integration Server*
- *Calling COBOL on z/OS IMS BMP (Batch) from Integration Server*

13 Calling COBOL on z/OS IMS MPP (IMS Connect) from Integration Server

Introduction

This scenario makes the following important assumptions:

- You have a working COBOL IMS MPP server. For illustration and examples on such a server, see *IMS MPP Message Interface (IMS Connect)*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You have IMS Connect Address Space running. This is needed to call the COBOL server program at runtime using the IMS Connect method. See *Preparing for IMS* in the webMethods EntireX Adapter documentation.

1: Extract the Interface of a COBOL Server

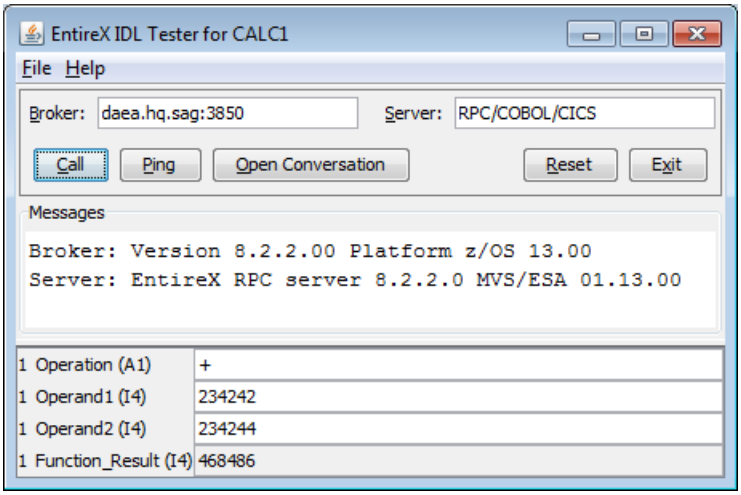
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

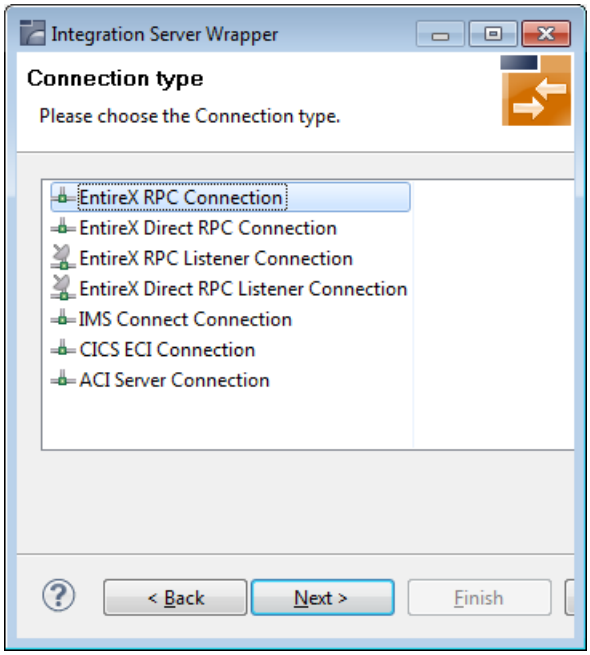


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

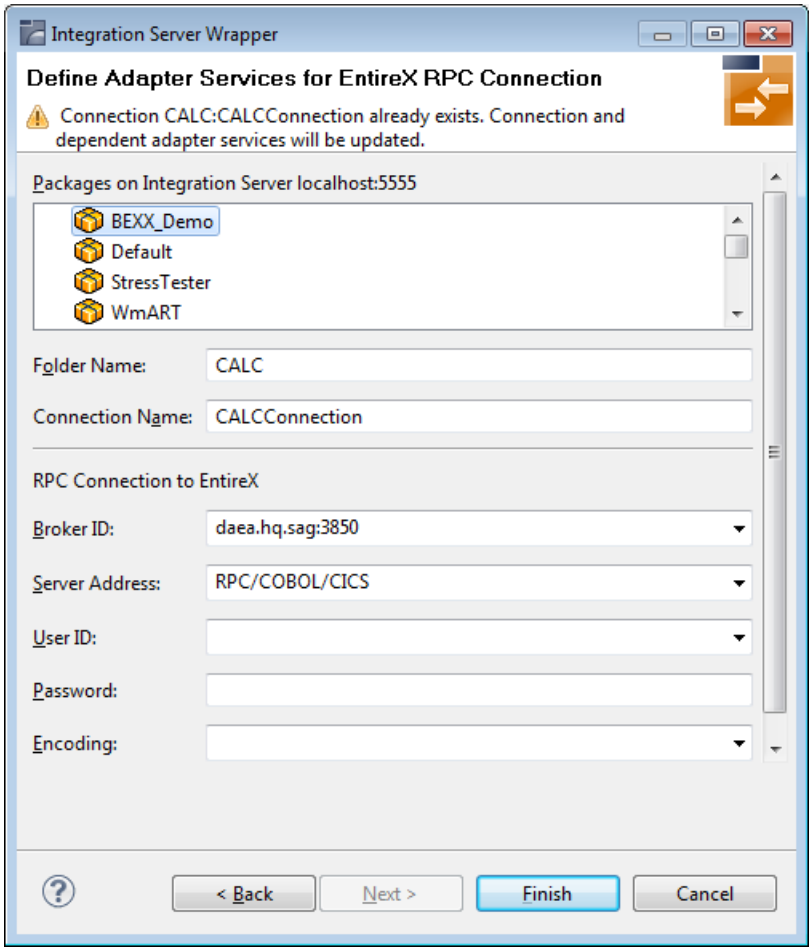
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



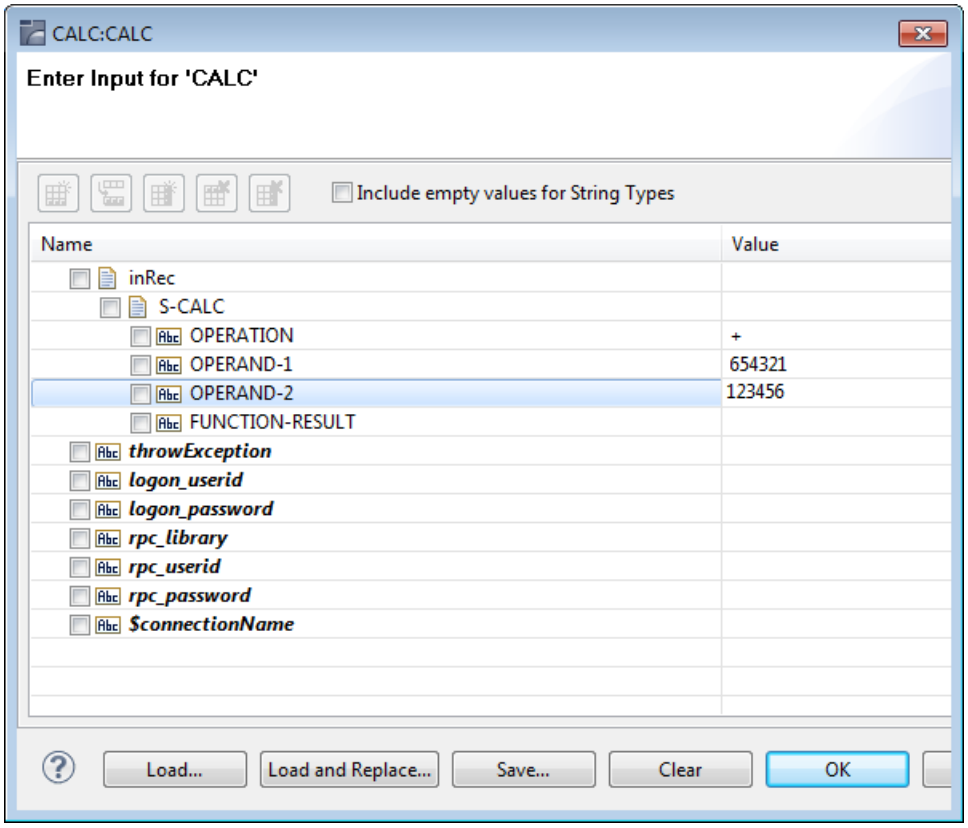
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

14 Calling COBOL on z/OS IMS BMP (Batch) from Integration Server

Introduction

This scenario makes the following important assumptions:

- You have a working COBOL IMS BMP server. For illustration and examples on such a server, see *IMS BMP with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks. The minimum requirement is the `DATA DIVISION` of the interface. The sources and copybooks must be files on your PC or remotely stored in PDS or CA Librarian data set and accessed via the *Batch RPC Server* (see the Batch RPC Server documentation).
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - an EntireX RPC server, see *Batch RPC Server*
  - For the *EntireX Direct RPC Connection* method you need the EntireX RPC server. See *Direct RPC* in the webMethods EntireX Adapter documentation and *Batch RPC Server*.

1: Extract the Interface of a COBOL Server

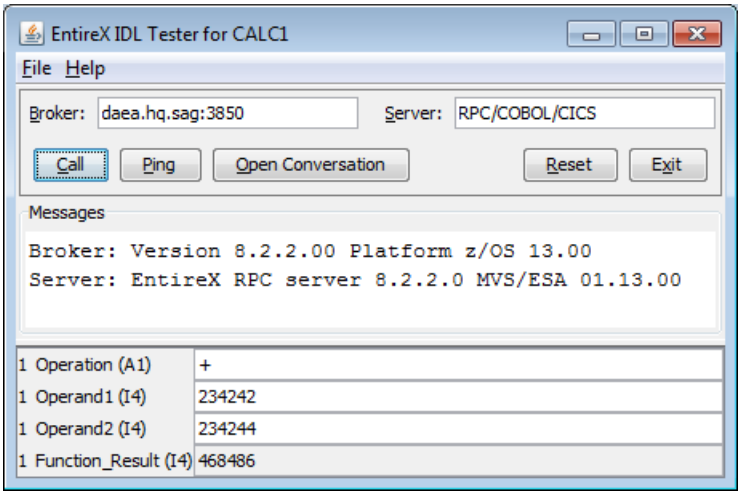
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

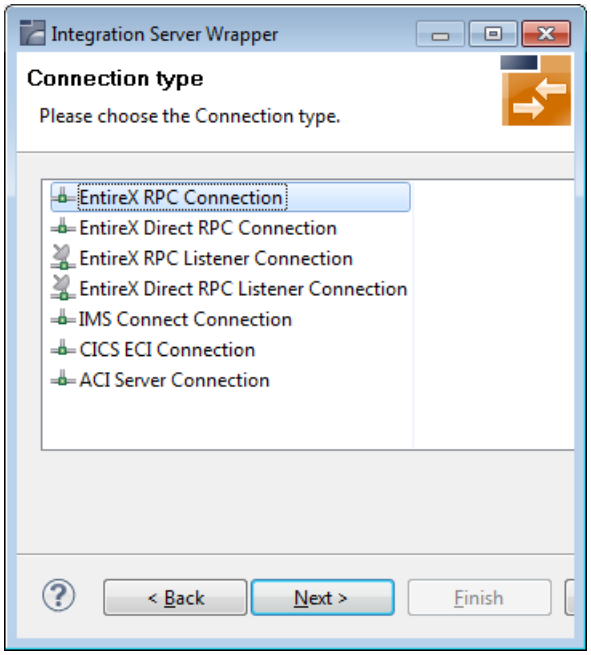


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

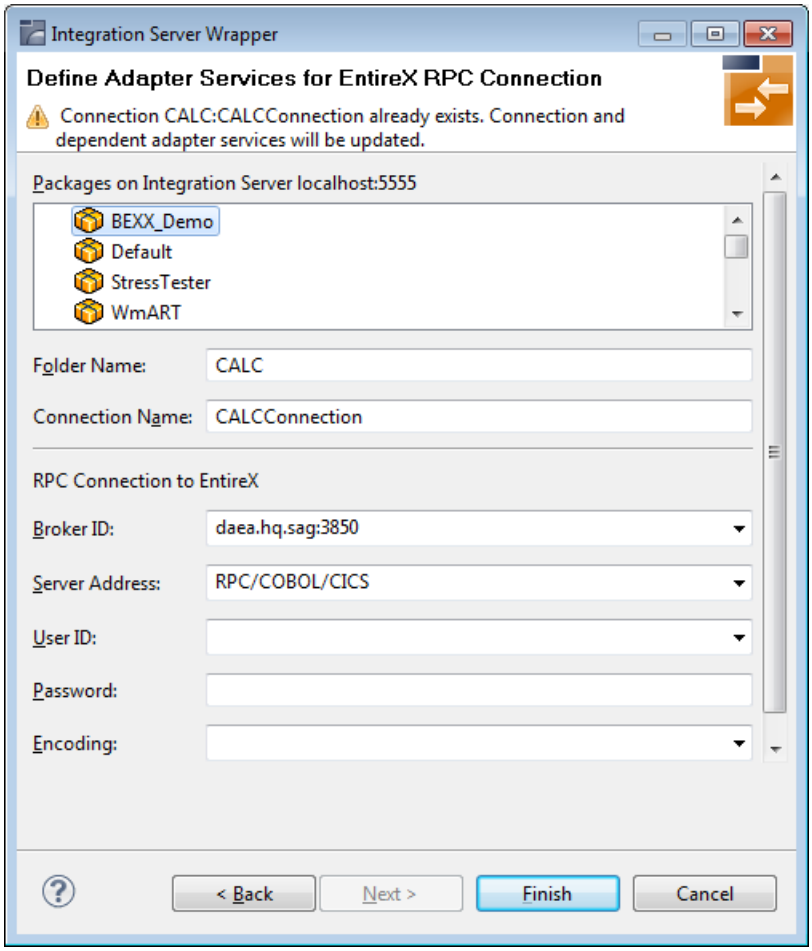
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



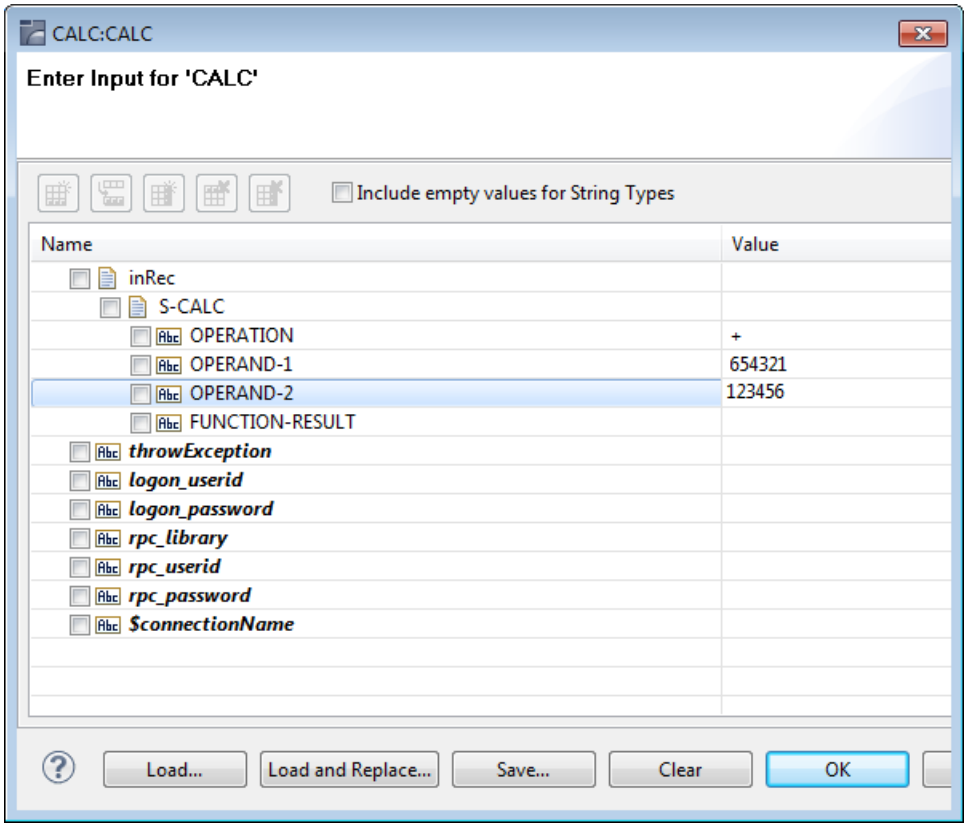
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:

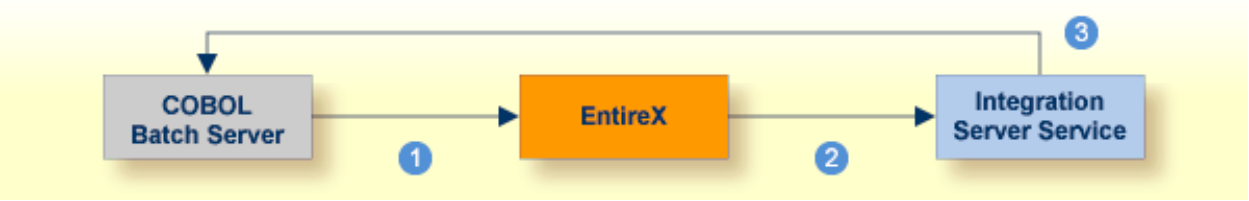


This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

15 Calling COBOL on z/OS Batch from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL batch server. For illustration and examples on such a server, see *Batch with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks. The minimum requirement is the `DATA DIVISION` of the interface. The sources and copybooks must be files on your PC or remotely stored in PDS or CA Librarian data set and accessed via the *Batch RPC Server* (see the Batch RPC Server documentation).
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - an EntireX RPC server, see *Batch RPC Server*
  - For the *EntireX Direct RPC Connection* method you need the EntireX RPC server. See *Direct RPC* in the webMethods EntireX Adapter documentation and *Batch RPC Server*.

1: Extract the Interface of a COBOL Server

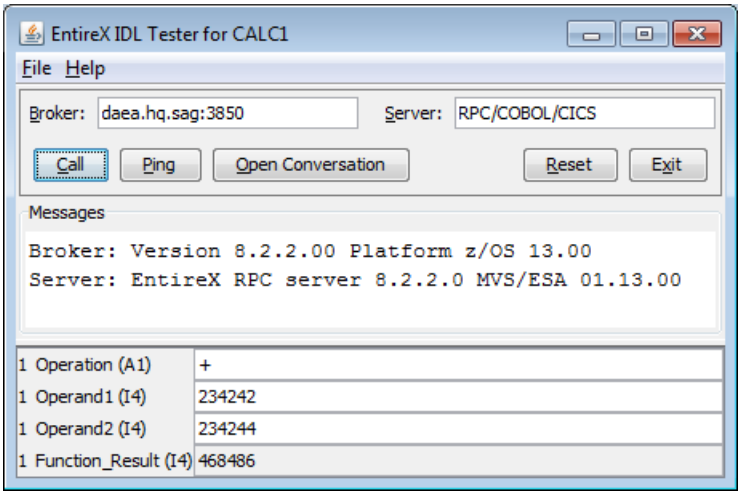
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

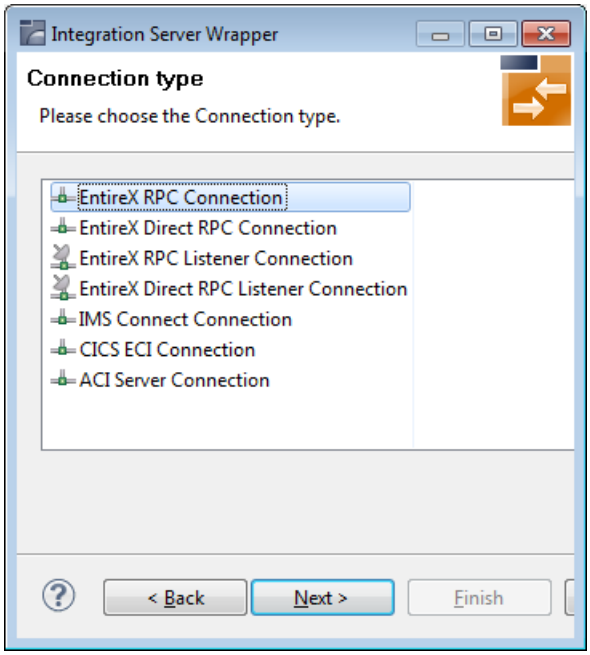


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

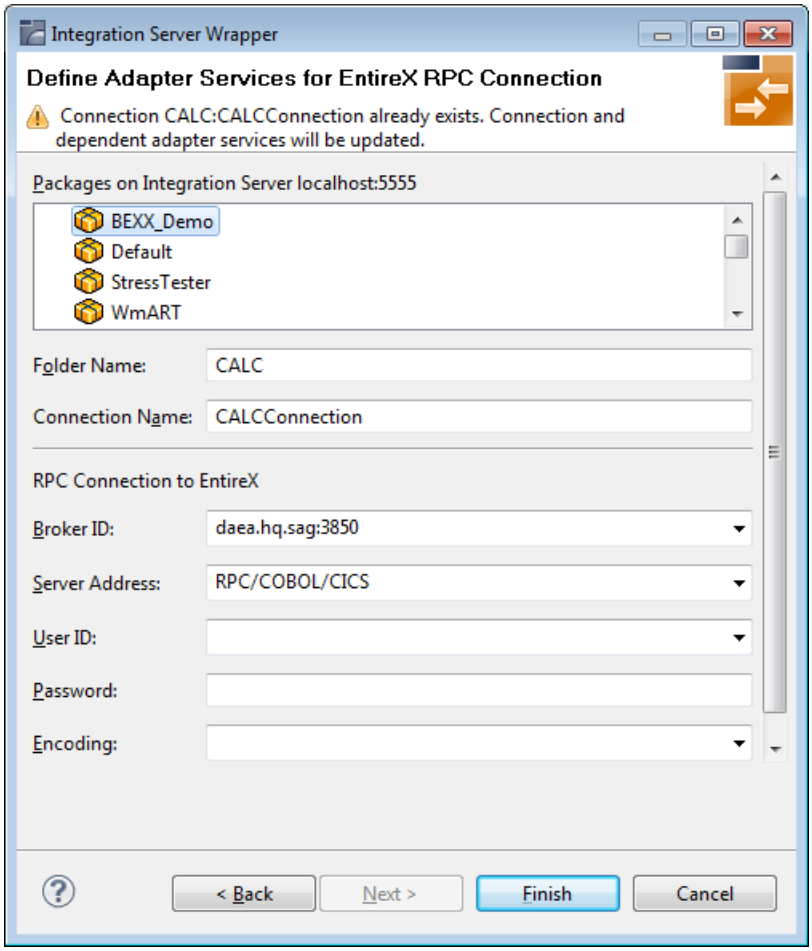
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



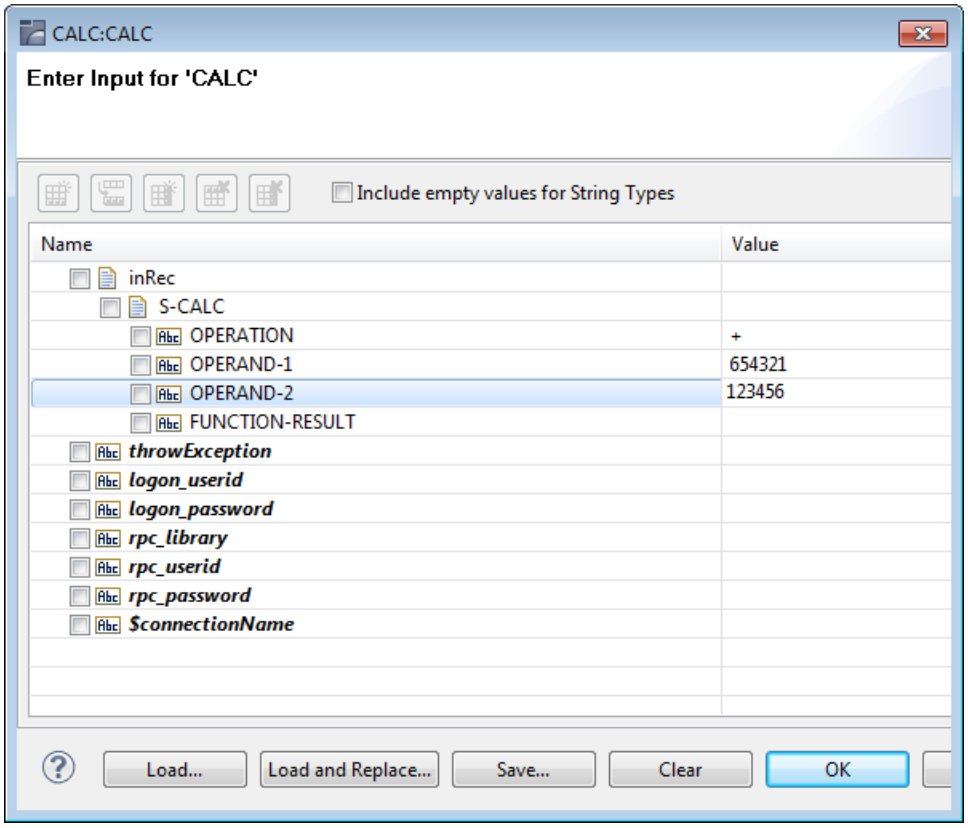
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

16 Calling COBOL on UNIX from Integration Server

Introduction

Under UNIX, a COBOL server running in Micro Focus environments can be called.

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

This scenario makes the following important assumptions:

- You have a working COBOL Micro Focus server. For illustration and examples on such a server, see *Micro Focus with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the EntireX Micro Focus COBOL RPC Server see *Micro Focus RPC Server*
  - For the *EntireX Direct RPC* connection method you need the EntireX Micro Focus COBOL RPC Server see *Micro Focus RPC Server*

1: Extract the Interface of a COBOL Server

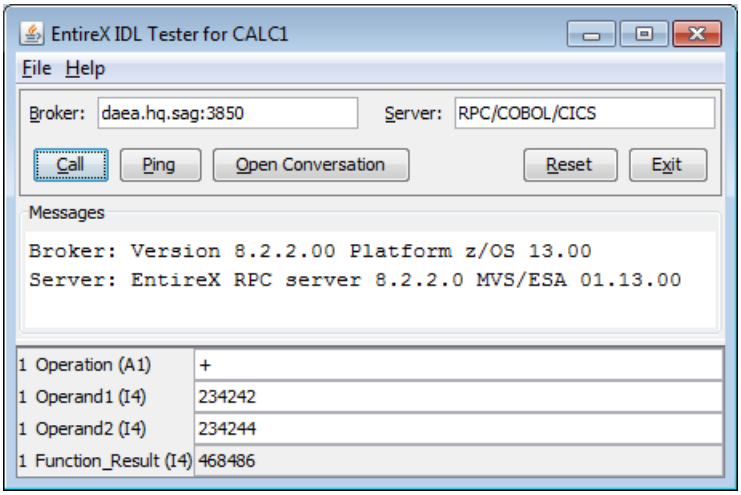
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

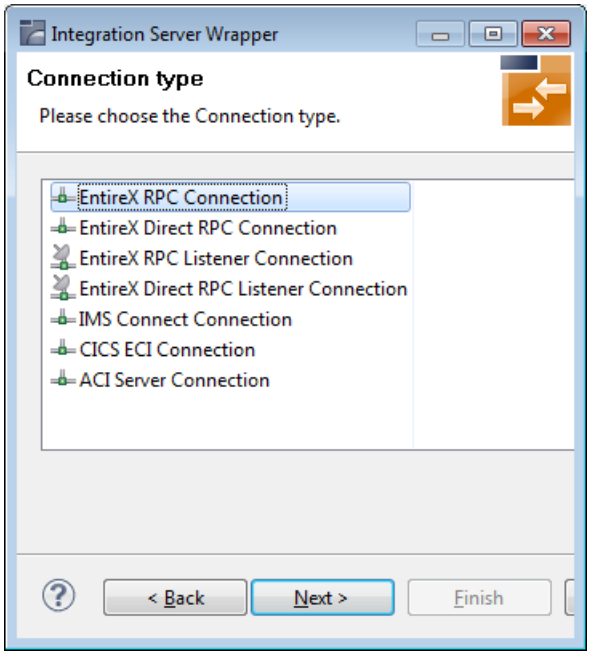


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

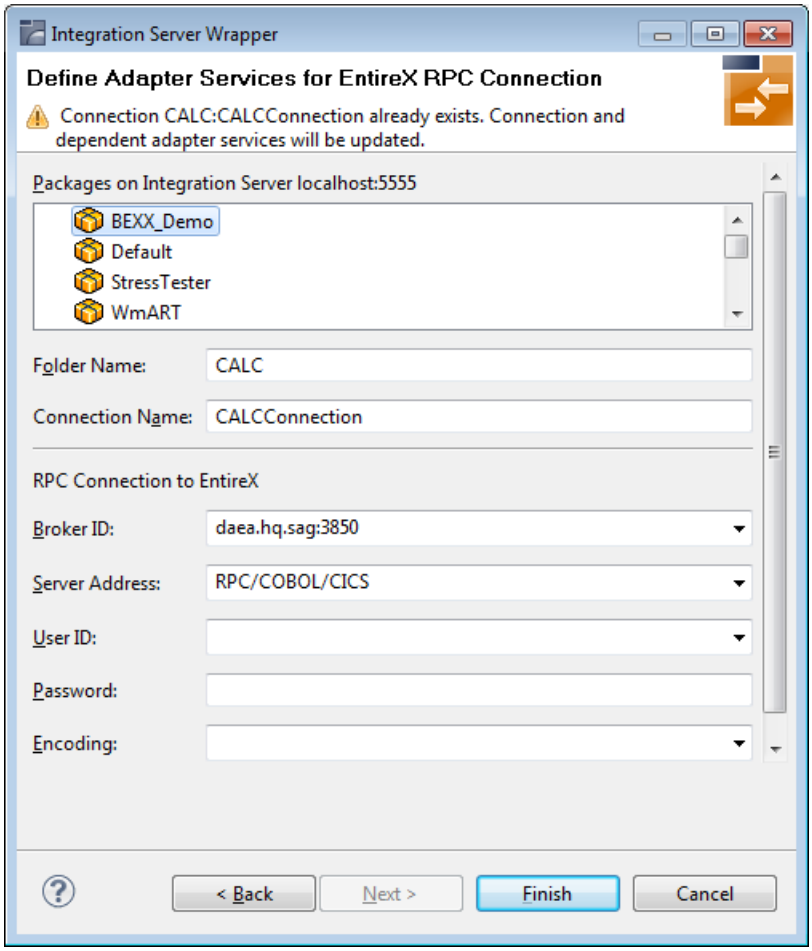
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



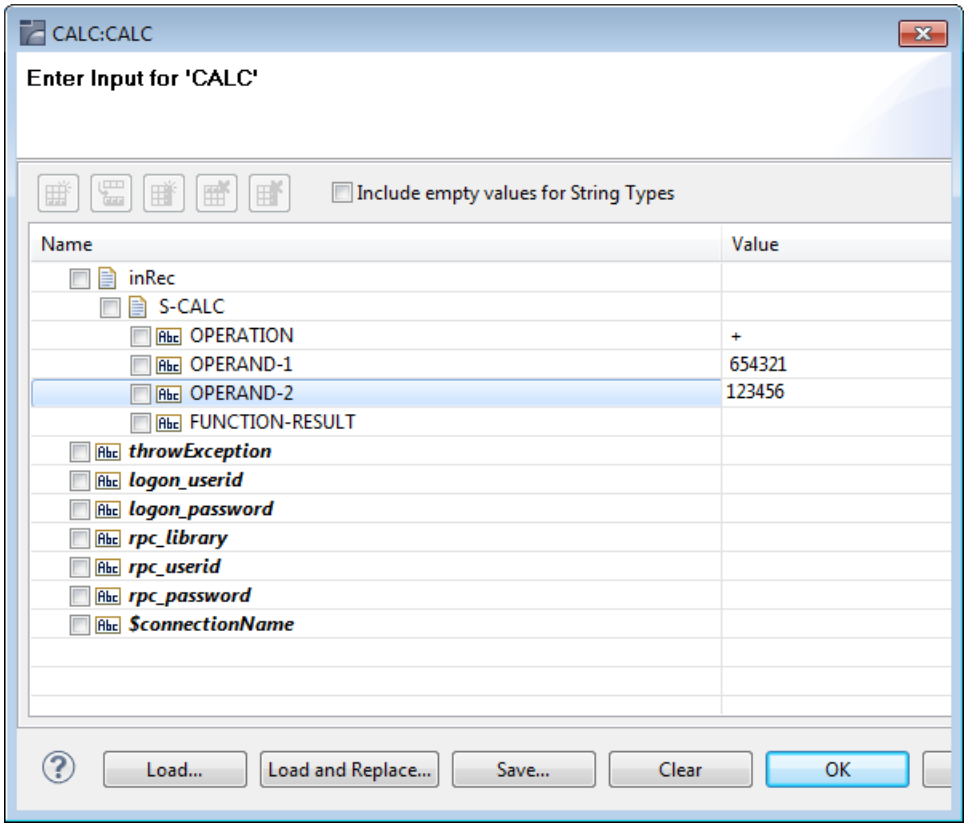
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

17 Calling COBOL on Windows from Integration Server

Introduction

Under Windows, a COBOL server running in Micro Focus environments can be called.

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

This scenario makes the following important assumptions:

- You have a working COBOL Micro Focus server. For illustration and examples on such a server, see *Micro Focus with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the EntireX Micro Focus COBOL RPC Server see *Micro Focus RPC Server*
  - For the *EntireX Direct RPC* connection method you need the EntireX Micro Focus COBOL RPC Server see *Micro Focus RPC Server*

1: Extract the Interface of a COBOL Server

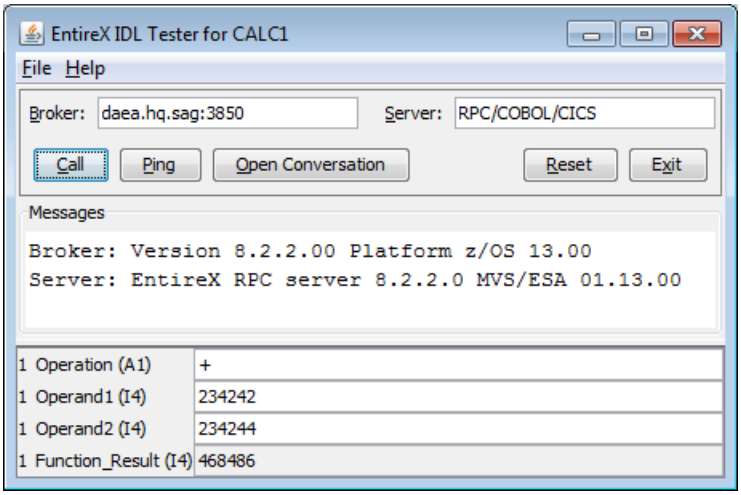
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

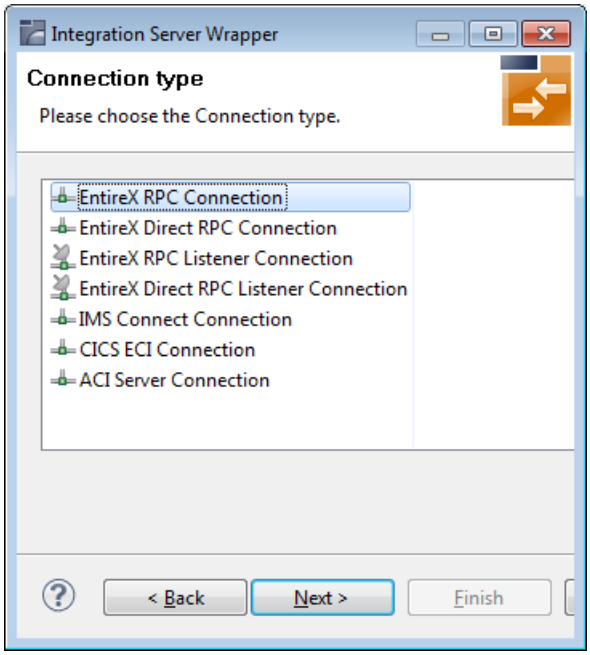


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

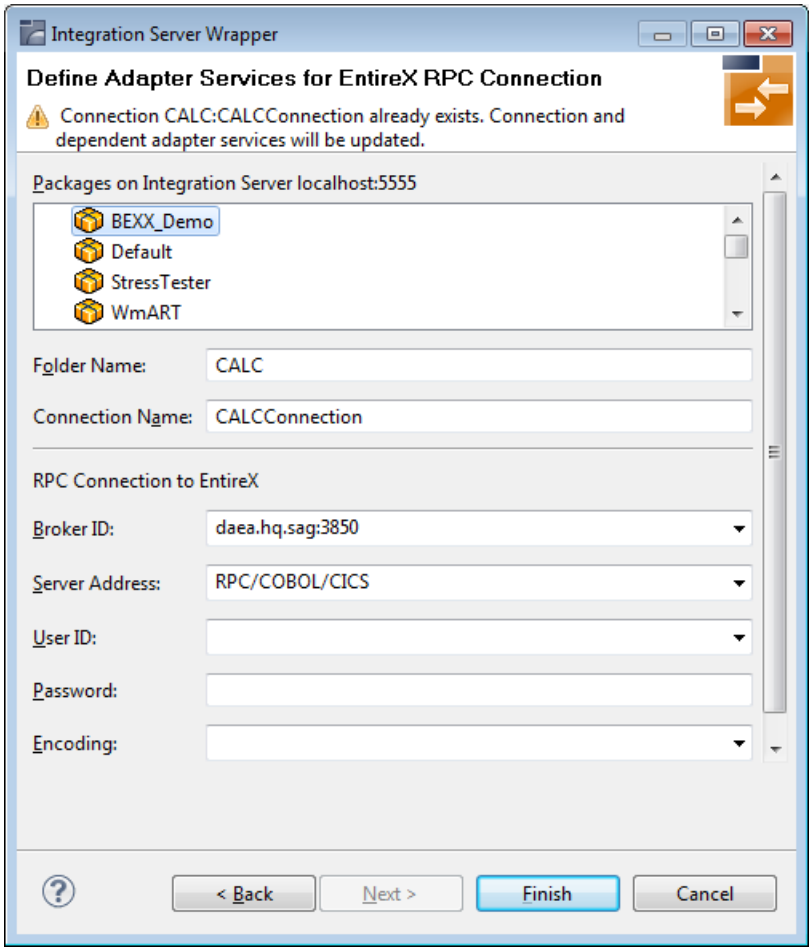
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



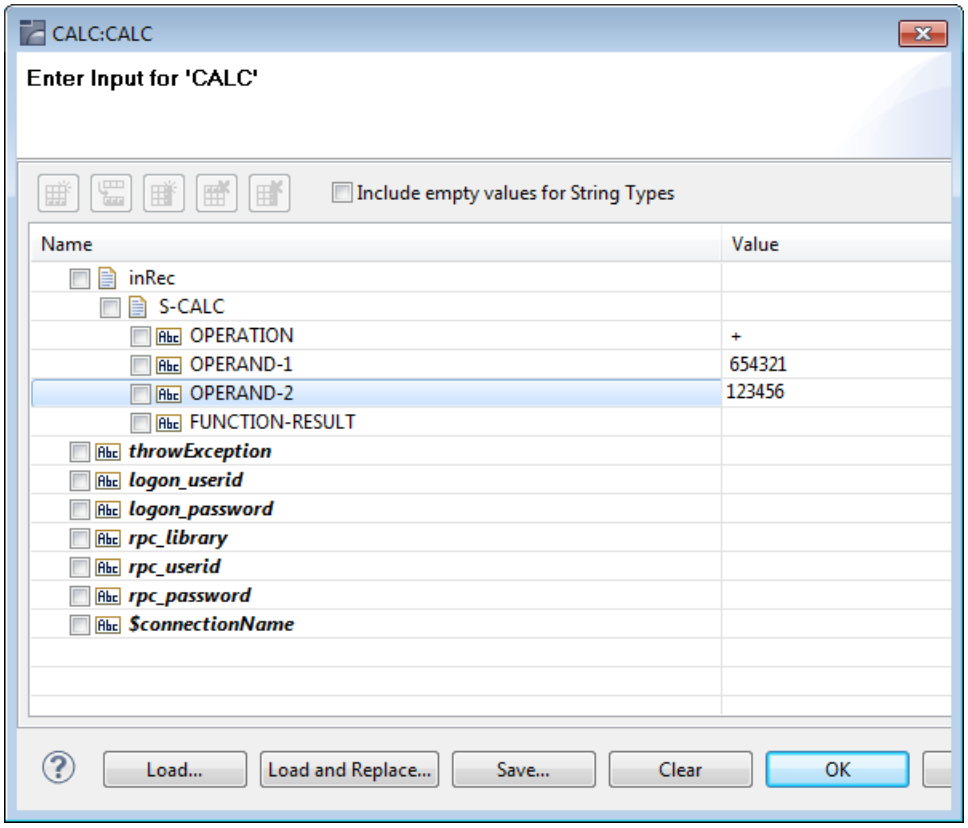
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:

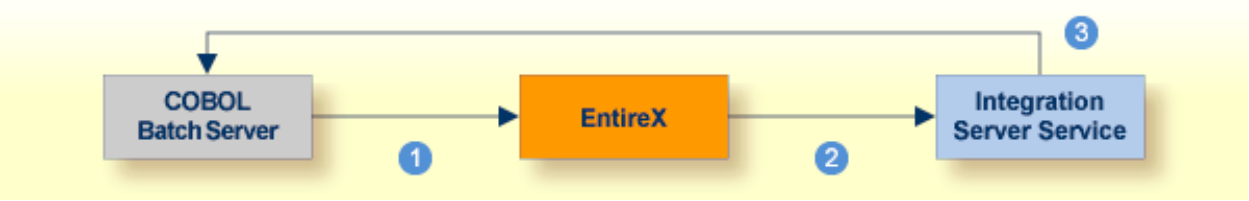


This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

18 Calling COBOL on BS2000/OSD from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL batch server. For illustration and examples on such a server, see *Batch with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface. The sources and copybooks must be files on your PC or remotely stored in LMS libraries and accessed via the *Extractor Service* (see the BS2000/OSD Administration documentation).
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC Connection* method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - an EntireX RPC server, see *Administering the BS2000/OSD Batch RPC Server* in the BS2000/OSD administration documentation
  - For the *EntireX Direct RPC* connection method you need the EntireX RPC server. See *Direct RPC* in the webMethods EntireX Adapter documentation and *Administering the BS2000/OSD Batch RPC Server* in the BS2000/OSD administration documentation.

1: Extract the Interface of a COBOL Server

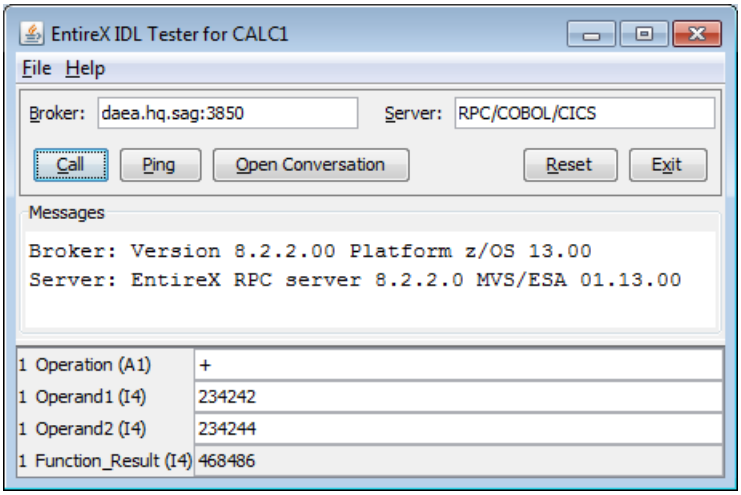
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

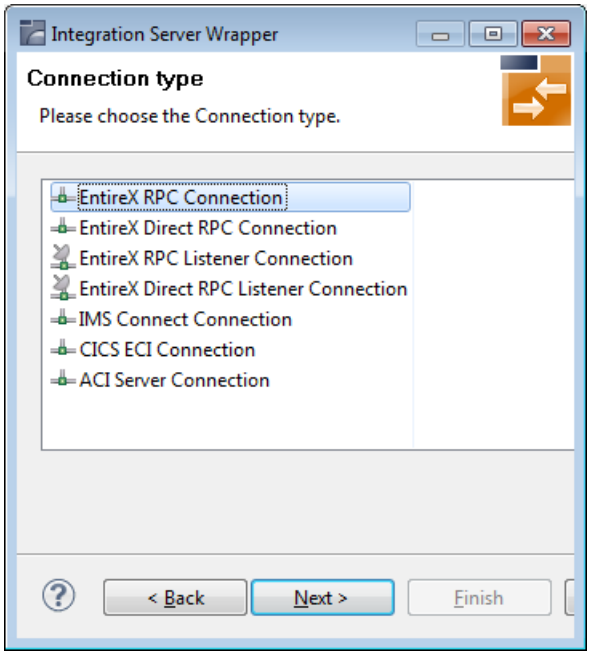


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

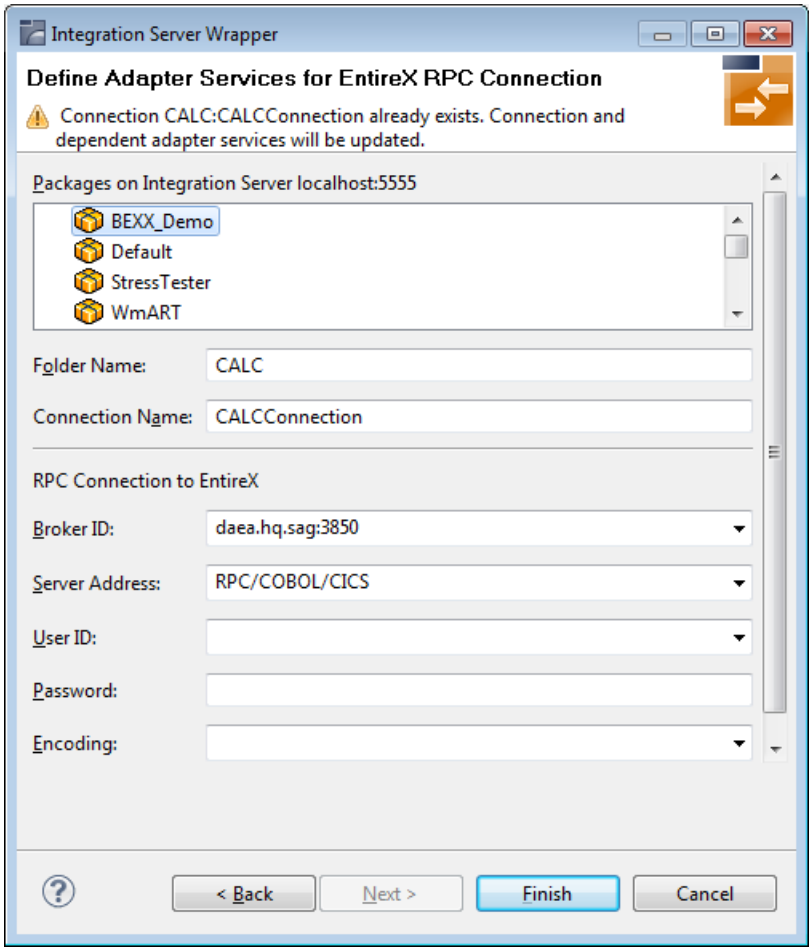
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



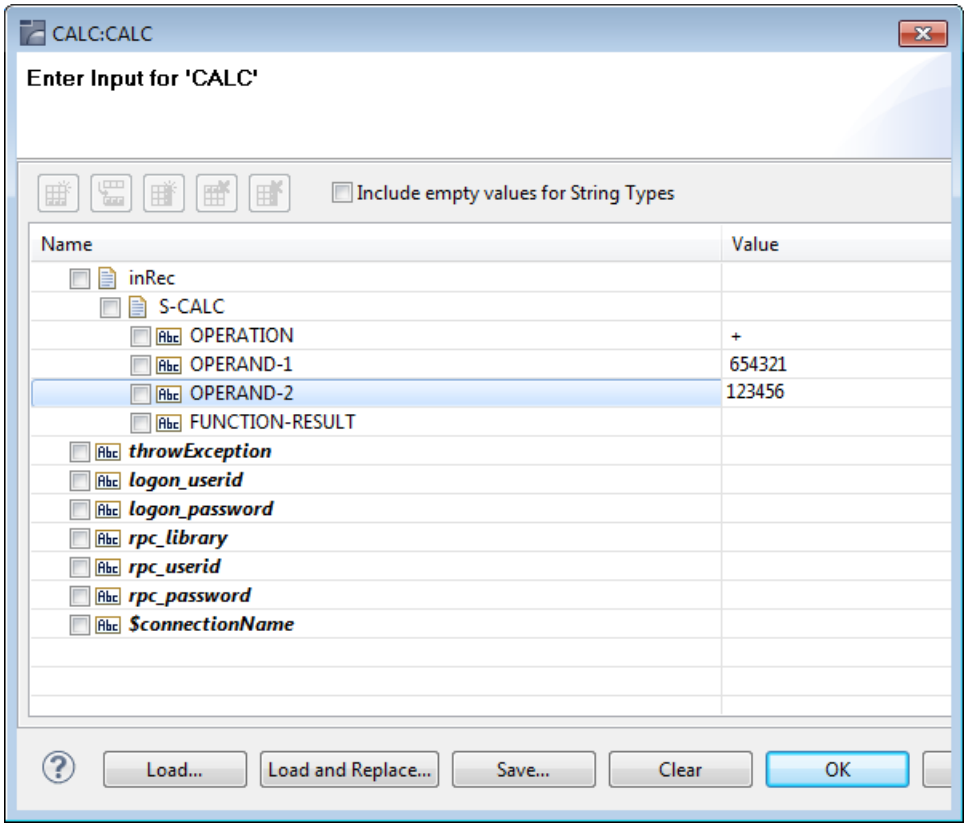
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

19 Calling COBOL on z/VSE from Integration Server

Under z/VSE, a COBOL server can be called in different environments:

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

Continue with the appropriate scenario:

- *Calling COBOL on z/VSE CICS from Integration Server*
- *Calling COBOL on z/VSE Batch Integration Server*

20 Calling COBOL on z/VSE CICS from Integration Server

There are different styles (interface types) for calling a COBOL server.

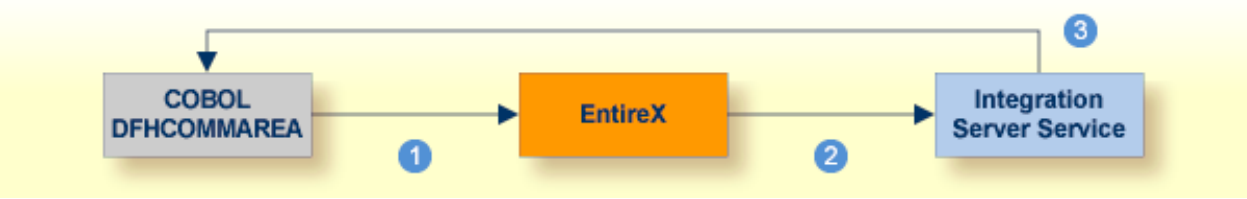
It is important to know the interface type of your COBOL server. If you are unsure, consult a COBOL CICS specialist or see description of interface type in the IDL Extractor for COBOL documentation for details and examples: `DFHCOMMAREA` *Calling Convention* | `DFHCOMMAREA` *Large Buffer Interface*.

When you are sure which programing style you are using, continue with the appropriate scenario:

- *Calling COBOL DFHCOMMAREA on z/VSE CICS from Integration Server*
- *Calling COBOL Large Buffer on z/VSE CICS from Integration Server*

21 Calling COBOL DFHCOMMAREA on z/VSE CICS from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL DFHCOMMAREA server. For illustration and examples on such a server, see *CICS with* `DFHCOMMAREA` *Calling Convention*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the EntireX CICS RPC Server (see the separate z/VSE documentation)
  - For the *EntireX Direct RPC* connection method you need the EntireX CICS RPC Server (see the separate z/VSE documentation)
  - For the *EntireX CICS ECI* connection method you need to configure the CICS ECI TCP/IP service within your CICS region. See *Preparing IBM CICS for ECI* in the webMethods EntireX Adapter documentation.

1: Extract the Interface of a COBOL Server

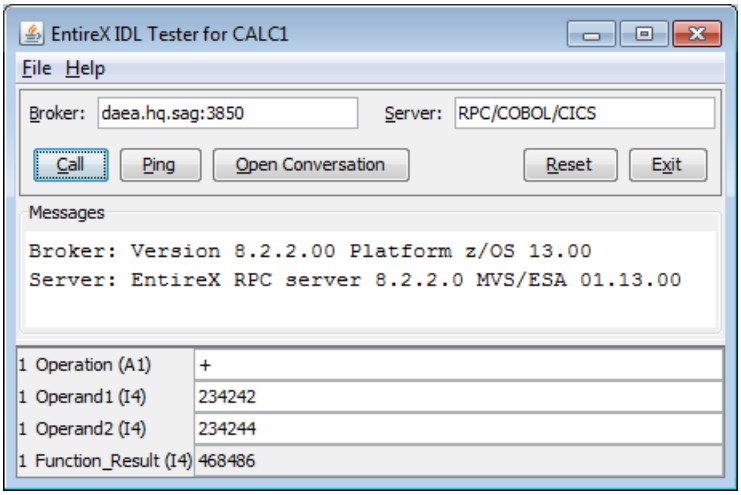
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

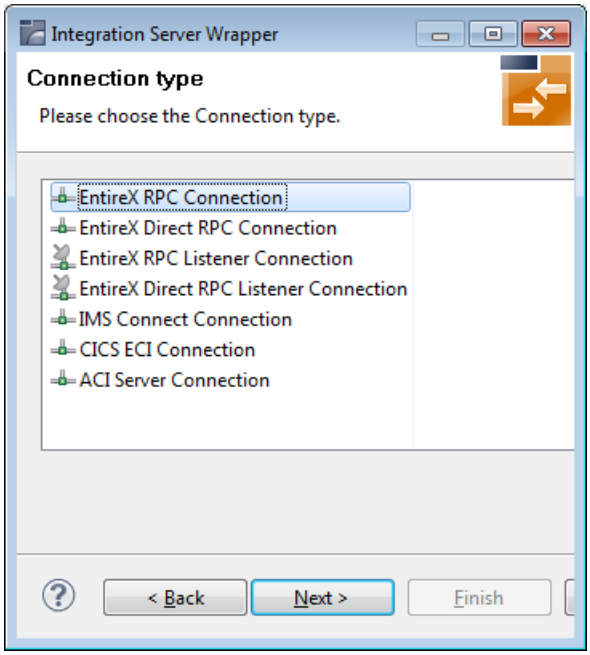


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

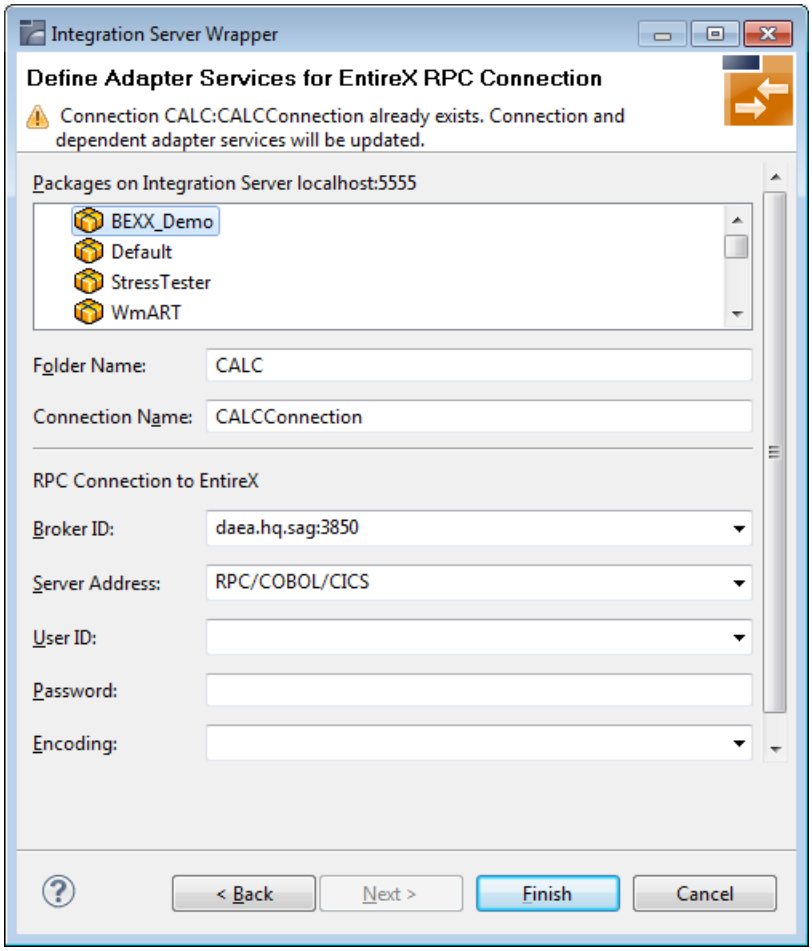
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



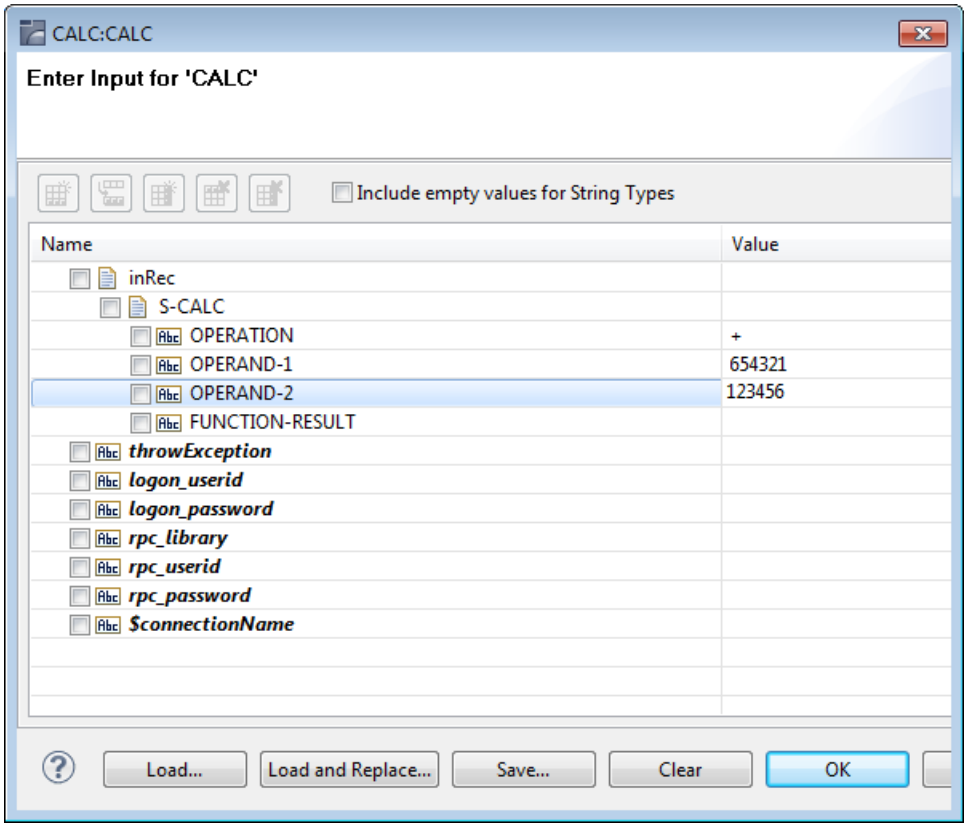
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:

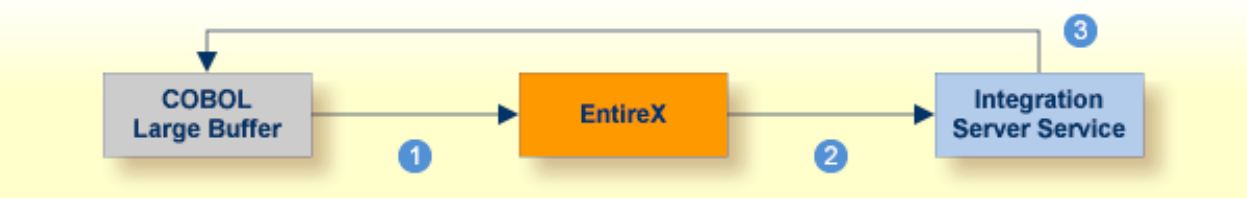


This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

22 Calling COBOL Large Buffer on z/VSE CICS from Integration Server

Introduction



This scenario makes the following important assumptions:

- You have a working COBOL Large Buffer server. For illustration and examples on such a server, see *CICS with* `DFHCOMMAREA` *Large Buffer Interface*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the EntireX CICS RPC Server (see the separate z/VSE documentation)
  - For the *EntireX Direct RPC* connection method you need the EntireX CICS RPC Server (see the separate z/VSE documentation)

1: Extract the Interface of a COBOL Server

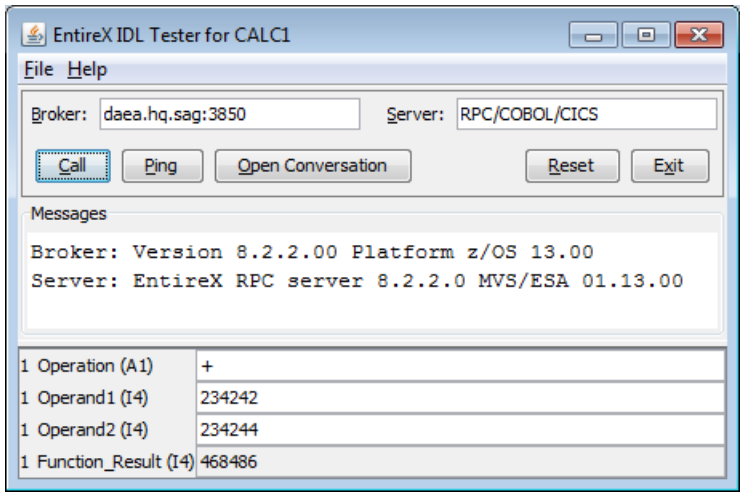
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

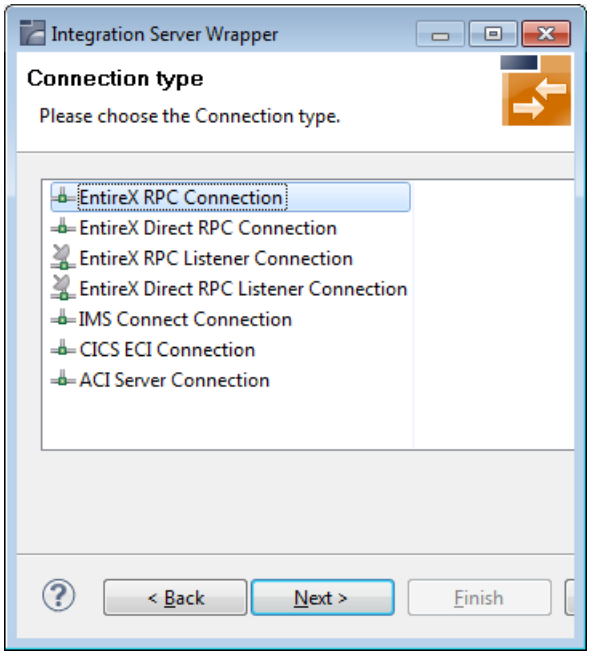


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

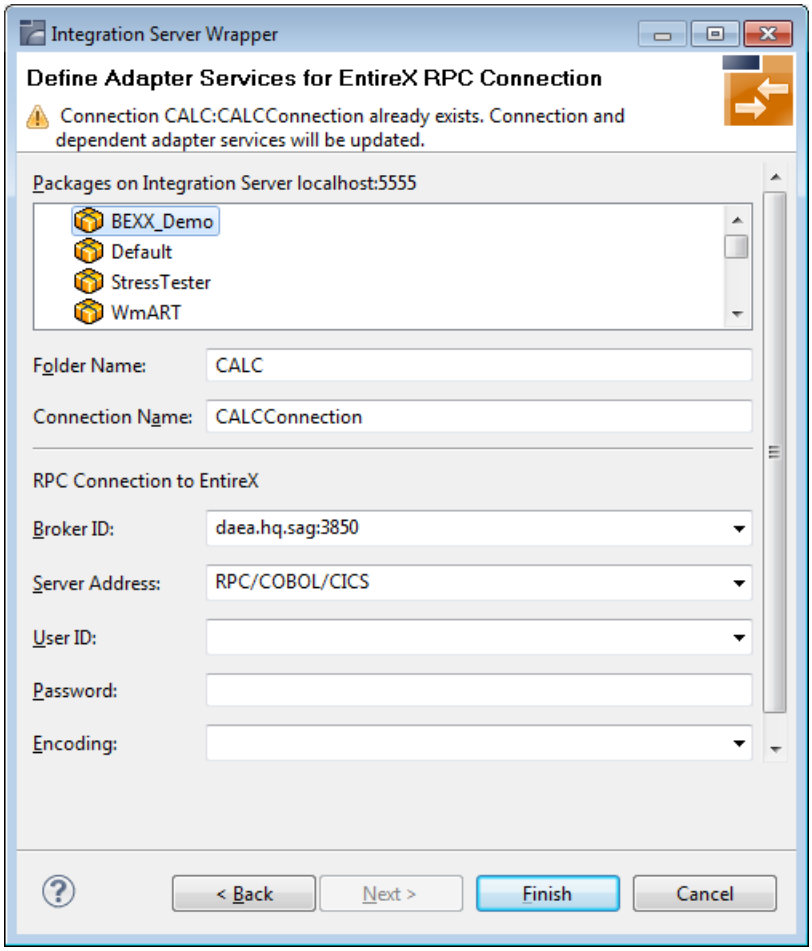
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



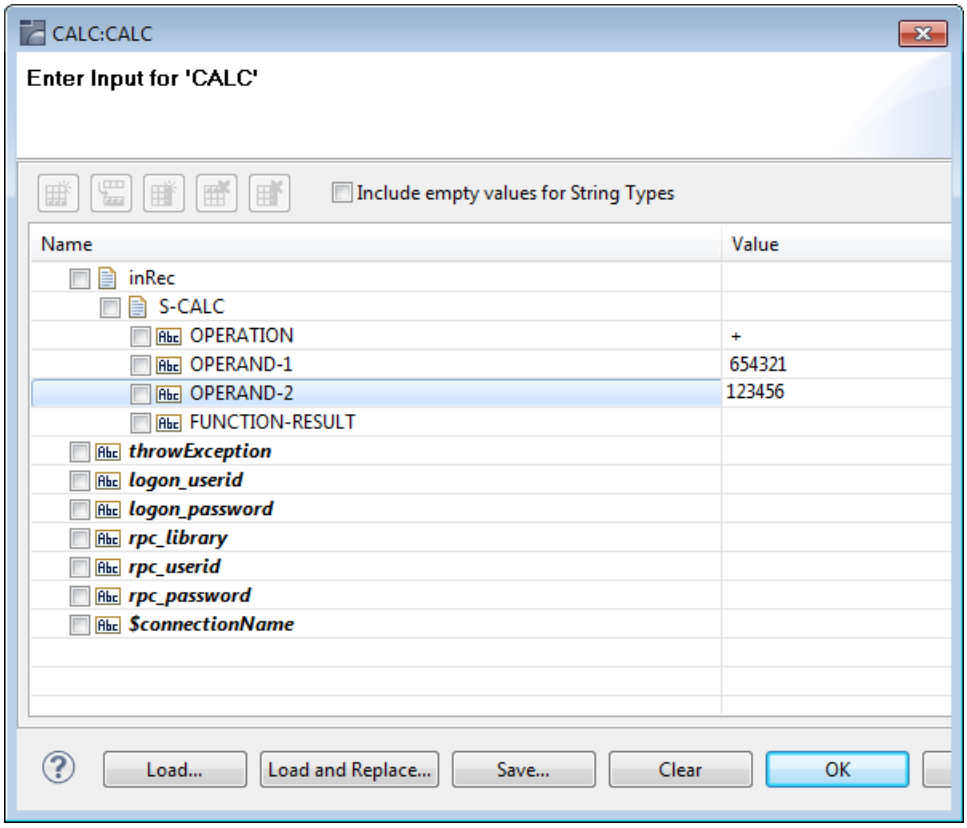
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

23 Calling COBOL on z/VSE Batch Integration Server

Introduction

Under z/VSE batch, a COBOL server with a standard call interface can be called.

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

This scenario makes the following important assumptions:

- You have a working COBOL batch server. For illustration and examples on such a server, see *Batch with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the EntireX RPC Batch Server (see the separate z/VSE documentation)
  - For the *EntireX Direct RPC* connection method you need the EntireX Batch RPC Server (see the separate z/VSE documentation)

1: Extract the Interface of a COBOL Server

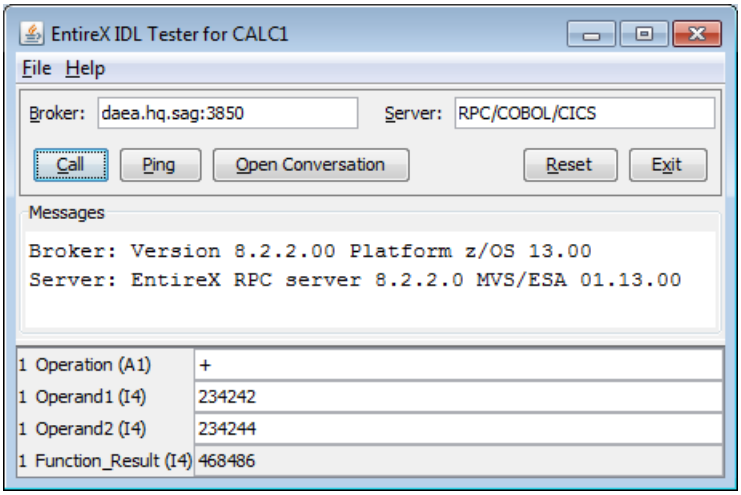
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

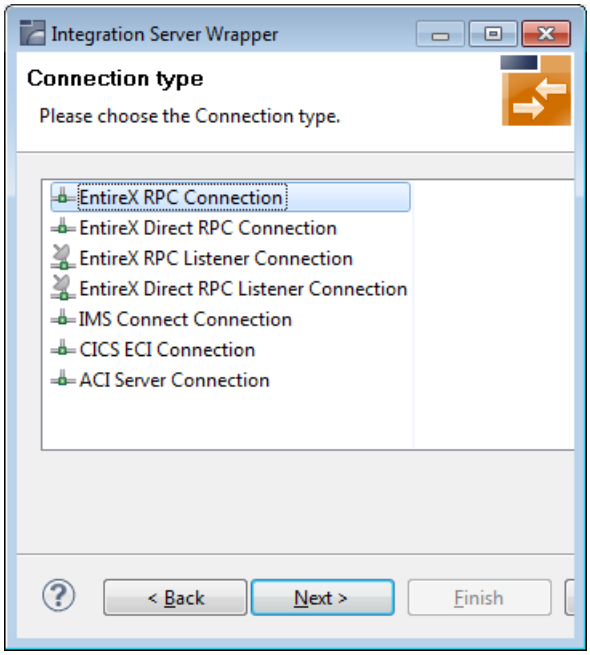


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

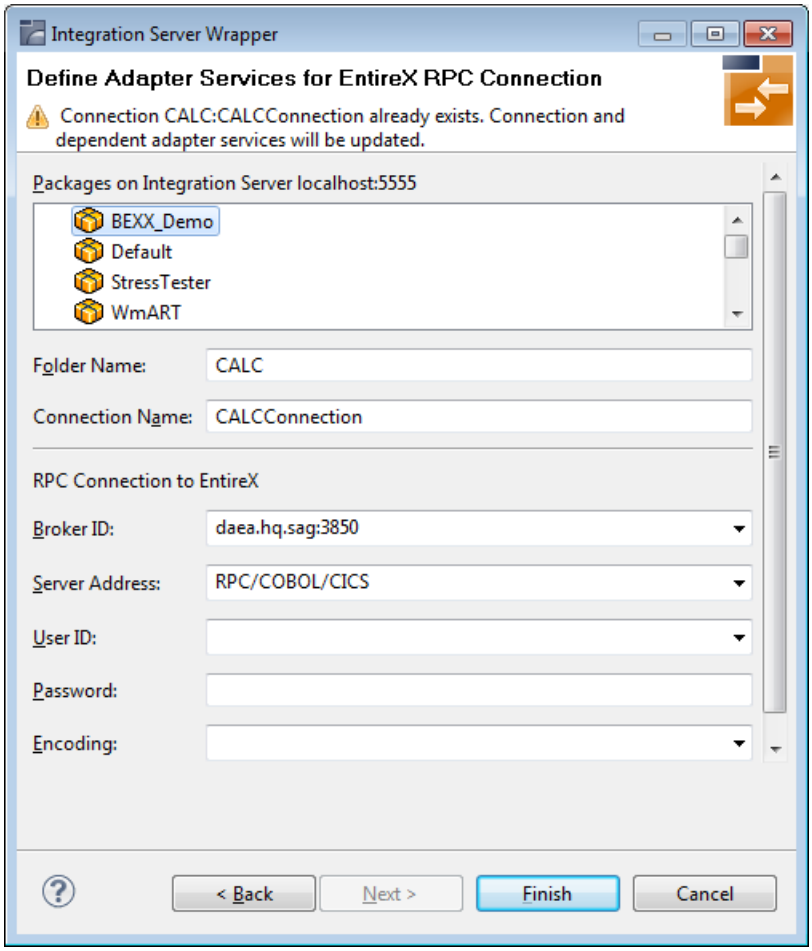
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



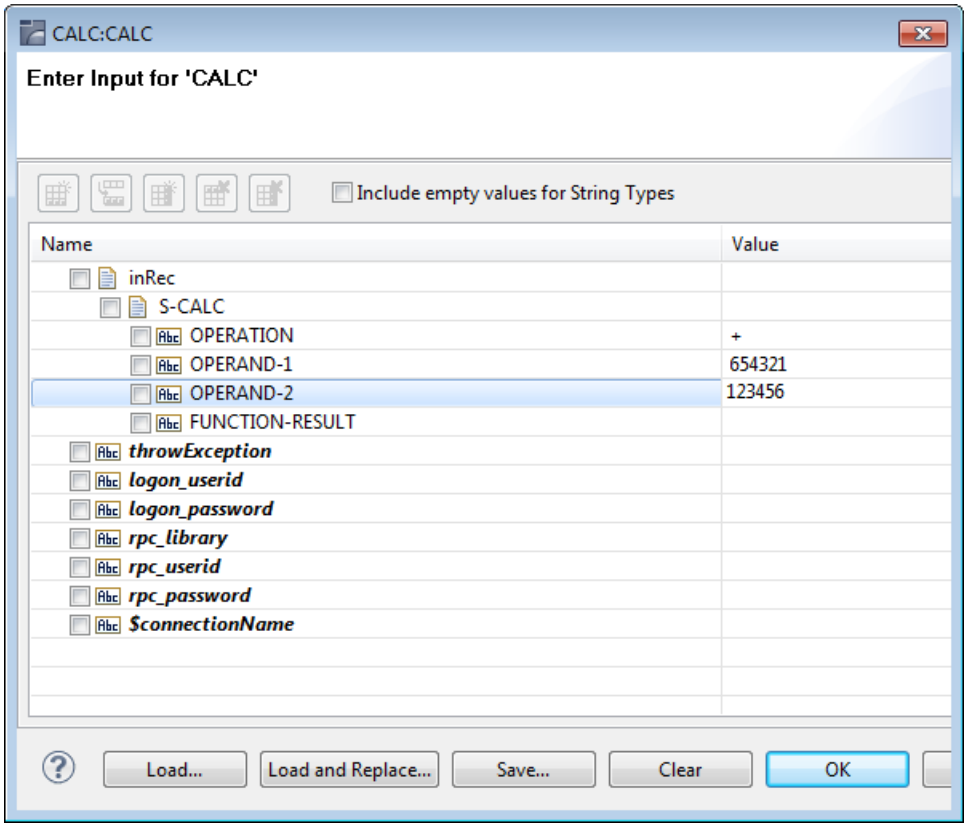
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

24 Calling COBOL on IBM i from Integration Server

Introduction

Under IBM i, a COBOL server with a standard call interface can be called.

1. Extract the interface of the COBOL server program.
2. Generate connection and application services in Integration Server.
3. Test the call from Integration Server to the COBOL server program.

This scenario makes the following important assumptions:

- You have a working COBOL batch server. For illustration and examples on such a server, see *Batch with Standard Linkage Calling Convention*.
- You have access to the related COBOL sources and copybooks as files on your PC. The minimum requirement is the `DATA DIVISION` of the interface.
- You have installed webMethods Integration Server and have a working IS instance and working webMethods EntireX Adapter.
- You can call the COBOL server program at runtime using different methods:
  - For the *EntireX RPC* connection method you need
    - an EntireX Broker on one of the supported platforms: z/OS | UNIX | Windows | BS2000/OSD | z/VSE (see separate documentation)
    - the EntireX IBM i RPC Server see *Administering the EntireX RPC Server* in the IBM i administration documentation
  - For the *EntireX Direct RPC* connection method you need the EntireX IBM i RPC Server see *Administering the EntireX RPC Server* in the IBM i administration documentation

1: Extract the Interface of a COBOL Server

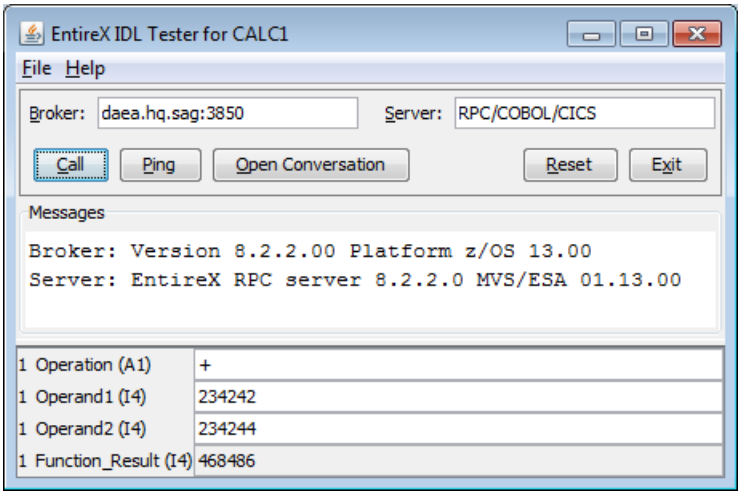
Follow the instructions for extracting COBOL, see *Using the IDL Extractor for COBOL - Overview* and choose *Scenario I: Create New IDL and SVM* if this is your first extraction. This process creates the following EntireX metafiles:

- IDL file. A Software AG IDL file contains definitions of the interface between client and server. See *Software AG IDL File* in the IDL Editor documentation.
- SVM file (optional). The server-side mapping file (SVM) contains COBOL-specific mapping information. See *Handling SVM Files*.

1a: (Optional) Test the Extraction Results

Optionally, you can test the results of the extraction operation, using the EntireX IDL Tester.

1. For the EntireX RPC Connection and the EntireX Direct RPC Connection method (not possible for other connection methods), test the COBOL Server backend using **Test Software AG IDL** from the Workbench:

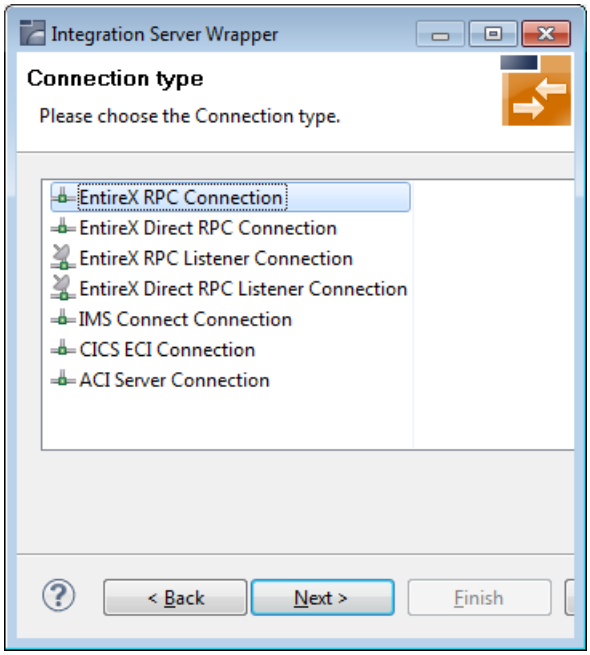


   Note that the Broker and Server parameters contain the explicit route to call the server program, and you can optionally ping the connection from this client. See *EntireX IDL Tester* in the EntireX Workbench documentation.

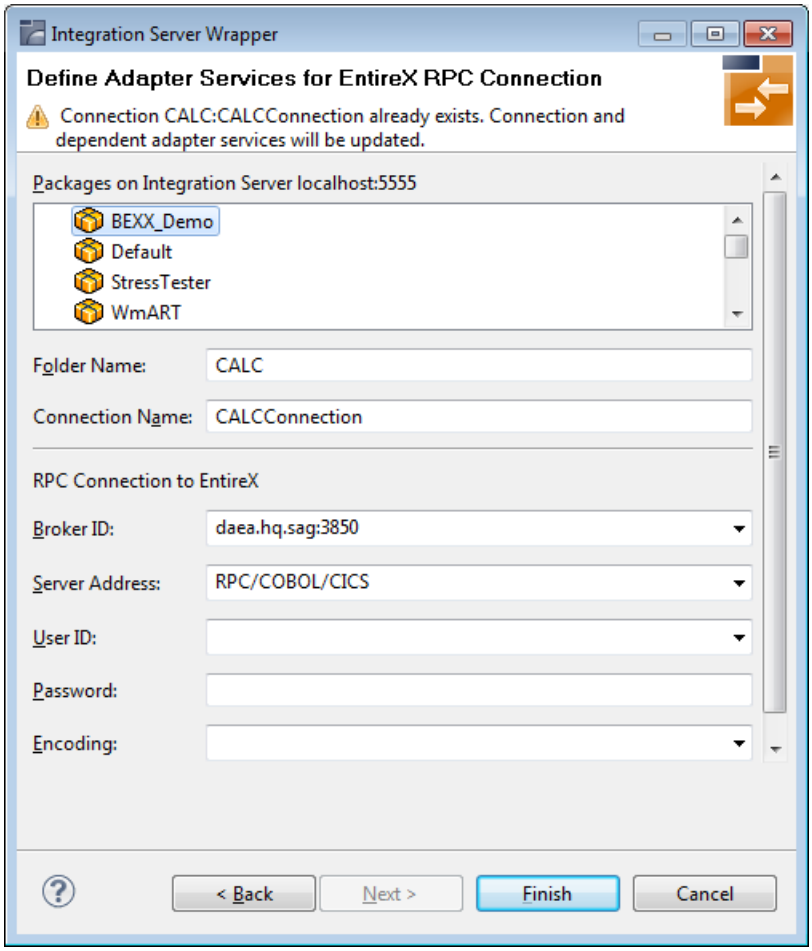
2. Check the IS log, the EntireX Adapter log, or the RPC logs. Applies to all connection methods.

2: Generate the Connection and Application Services in Integration Server

Select the IDL file, and from the context menu choose **Generate webMethods Connection from IDL file...**. From the wizard, select an Integration Server instance and select the connection type.



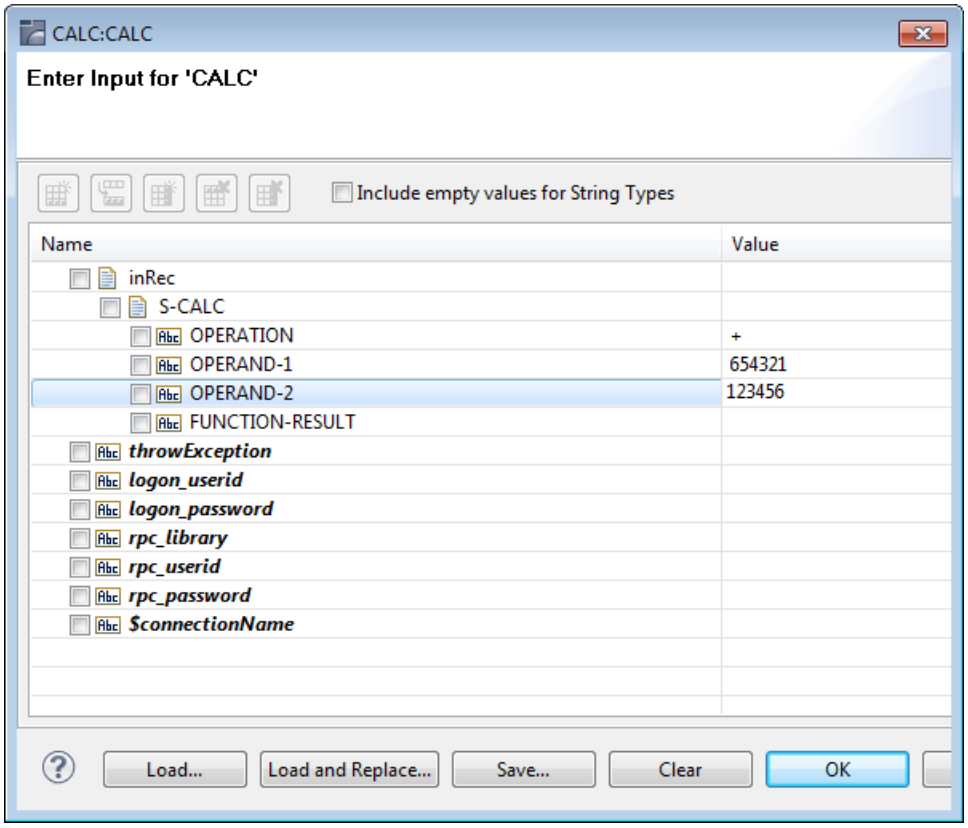
Then select the namespace where you want to write the services to, and specify the connection properties.



Choose **Finish**. The connection service will be automatically enabled in the Integration Server.

3: Test the Call from Integration Server to COBOL

From the **Service Development** perspective, refresh the package where the connection service was written, select the Adapter service, and use the service test to Run Service:



This invokes the adapter service through the connection service.

In case of error or unexpected results, use the IDL Tester as described under *Step 1a* above.

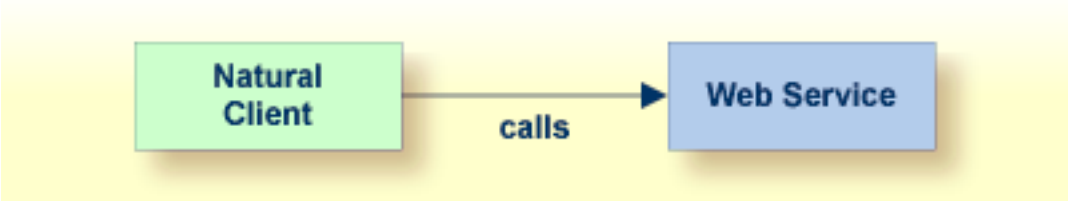
IV Connecting Web Services

- *Calling a Web Service from Natural*
- *Calling a Web Service from COBOL*

See also *Calling Natural from a Web Service* | *Calling COBOL from a Web Service*.

25 Calling a Web Service from Natural

Scenario: “I want to call a Web service from a Natural application.”

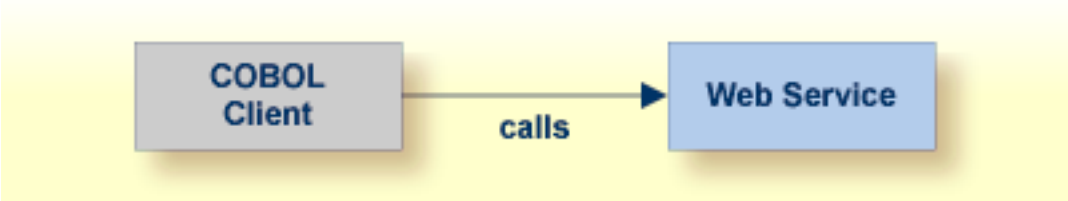


Solution: Select an existing Web service (1) and generate the integration logic (2) to call it from a Natural application (3). See also the steps below.

1. Extract WSDL from Web service. See *Using the Software AG IDL Extractor for WSDL*.
2. Generate objects for Natural client application. See *Natural Wrapper*.
3. Test call from Natural client to Web service. See *Sample Generation Result for the Client Side* under *Using the Natural Wrapper*.

26 Calling a Web Service from COBOL

Scenario: “I want to call a Web service from a COBOL application.”



Solution: Select an existing Web service (1) and generate the integration logic (2) to call it from a COBOL application (3). See also the steps below.

1. Extract WSDL from Web service. See *Using the Software AG IDL Extractor for WSDL*.
2. Generate objects for COBOL client application. See *COBOL Wrapper*.
3. Test call from COBOL client to Web service.