ᔓ software ᴬᴳ

webMethods EntireX Adapter

EntireX Adapter Installation and User's Guide

Version 9.5 SP1

November 2013

webMethods EntireX Adapter

Table of Contents

webMethods EntireX Adapter Installation and User's Guide

This guide describes how to install, configure and use the webMethods EntireX Adapter. It contains information for administrators and application developers who want to call remote procedures on mainframes and other hosts, using the EntireX RPC technique. To use this guide effectively, you should be familiar with EntireX's Remote Procedure Call (RPC) technology. For further information see *Related Literature* below.

In this guide the following topics are covered:

Related Literature

- *Integration Server Administrator's Guide*. See
  https://empower.softwareag.com/Products/Documentation/default.asp under "webMethods" >
  "webMethods Product Suite" > "Integration Server" > "Documentation by Product"

- the documentation on *webMethods EntireX*, including the *webMethods EntireX Broker* and
  webMethods EntireX RPC Servers. See
  https://empower.softwareag.com/Products/Documentation/default.asp under "webMethods" >
  "webMethods Product Suite" > "Documentation by Product".

- the documentation on the Software AG Designer. See
  https://empower.softwareag.com/Products/Documentation/default.asp under "webMethods" >
  "webMethods Product Suite" > "Documentation by Product"

- The Software AG Designer Online Help.

- *Software AG Installer*. See *https://empower.softwareag.com/Products/Documentation/default.asp* under
  "Software AG Installer and Update Manager"

- *webMethods Adapter Development Kit User's Guide*. See
  https://empower.softwareag.com/Products/Documentation/default.asp under "webMethods" >
  "webMethods Adapters and eStandards Modules" > "Adapters"

- the documentation of the Eclipse Update Manager, located at
  http://help.eclipse.org/helios/topic/org.eclipse.platform.doc.user/tasks/tasks-129.htm

- Natural Product Documentation at
  https://empower.softwareag.com/Products/Documentation/default.asp.

This documentation is part of a documentation module including the following documents:

- *webMethods EntireX Adapter Release Notes*.

- For the EntireX Workbench (*webMethods Integration Server Wrapper* and *Software AG IDL Extractor
  for webMethods Integration Server*), refer to
  https://empower.softwareag.com/Products/Documentation/default.asp.

1 Installing the EntireX Adapter

This chapter describes how to install and configure the EntireX Adapter and is intended for system administrators. For prerequisites, see *Supported Versions of webMethods Integration Server and EntireX Workbench* in the *Release Notes*.

- To install the EntireX Adapter into an Integration Server, use the Software AG Installer. For further information, see *Software AG Installation Guide*.

- To use connections to IMS Connect, CICS ECI or Direct RPC, you need a license file for the EntireX Adapter.

  ▶ **To use connections to IMS Connect, CICS ECI, or Direct RPC**

  1 Copy the license file to a safe folder. The package WmEntireX is not a safe folder. The Integration Server deletes the license file on update.

  2 Change the location of the license file in the License Information Screen in the Adapter Administration GUI (see *License Information*).

  The EntireX Adapter does not assume that the EntireX Mini Runtime is installed. Installing the EntireX Adapter into the Integration Server is all that is required on the machine of the Integration Server.

- To configure the EntireX Adapter, see *Settings and Information* and *Direct RPC*.

2 About the EntireX Adapter

The EntireX Adapter is an add-on to the webMethods Integration Server that enables you to call remote procedures on mainframes and other hosts, using the EntireX RPC technique. The adapter provides seamless and real-time communication with EntireX RPC servers and clients, Natural RPC servers and clients, ACI Servers, ApplinX, IMS Connect, and CICS ECI. Using the EntireX Adapter, Integration Server applications can create and run services that can exchange messages with RPC servers and clients on mainframe and other hosts.

Architecture and Components

The EntireX Adapter enables you to configure the following components:

- **Adapter Connections**
  Enable the webMethods Integration Platform to connect at runtime to EntireX RPC servers, Natural RPC servers, ACI servers, ApplinX, IMS Connect, and CICS ECI.

- **Adapter Services**
  Perform operations on EntireX RPC servers and clients, using the built-in services provided with the EntireX Adapter.

- **Listener Notifications**
  Provides synchronous notification of RPC messages that are called by EntireX RPC clients.

The following diagram illustrates how the EntireX Adapter interfaces with EntireX RPC servers and clients. The various architectural pieces involved in the integration process are described below the diagram.

```
Integration Server
    EntireX Adapter

    Adapter          Adapter
    Service          Connection

                              EntireX
                              Broker

                              1    RPC Servers
                              2
                              3    RPC Clients
                              4
                              5    ApplinX Server
                              6    CICS ECI
                              7    IMS Connect
                              8    ACI Server
```

1 EntireX Direct RPC Connection
2 EntireX RPC Connection or EntireX Reliable RPC Connection
3 EntireX RPC Listener Connection or EntireX Reliable RPC Listener Connection
4 EntireX Direct RPC Listener Connection
5 ACI Server Connection
6 CICS ECI Connection
7 IMS Connect Connection
8 ACI Server Connection

- **webMethods Integration Server**
  The EntireX Adapter is installed and runs on the Integration Server.

- **WmART Package**
  The WmART package provides a common framework for webMethods Adapters to utilize the Integration Server functionality, making the Integration Server the runtime environment for the EntireX Adapter. The WmART package is installed with the Integration Server.

- **EntireX Adapter**
  The EntireX Adapter is delivered as an adapter with the Software AG Installer. The package is called WmEntireX. The EntireX Adapter provides IS Administration Console and Software AG Designer user interfaces that enable you to configure and manage adapter connections, adapter services, and adapter notifications. However, it is recommended to use the Eclipse-based EntireX Workbench with the plug-ins for the EntireX Adapter to generate adapter connections and all

other nodes for the Integration Server. The plug-in supports secure connections to the Integration Server and updating the nodes.

■ **Adapter Connection Templates**
Adapter connections enable the EntireX Adapter to connect to EntireX RPC servers and clients. You must configure an adapter connection before you can configure adapter services or adapter notifications. The EntireX Adapter's connections are generated from a Software AG IDL file at design time with the EntireX Workbench (Eclipse-based). The IDL (Interface Definition Language) defines the interface of the remote procedures. Use the IS Administration Console to configure adapter connections. For a detailed description of adapter connections, see *EntireX Adapter Connections* (main section).

■ **Adapter Notification Template**
Adapter notifications enable the EntireX Adapter to listen to requests from EntireX RPC clients. These are used only for Reliable RPC Listener connections. The EntireX Adapter provides an adapter notification template that enables you to configure adapter notifications. You automatically create adapter notifications in the Eclipse plug-in when you create a Reliable RPC Connection with the Listener. However, you can use the Designer using the EntireX Adapter's notification templates. For a detailed description of adapter notifications, see *Listeners and Adapter Notifications*.

■ **EntireX RPC Servers and Clients**
See the *webMethods EntireX Administration* (at *https://empower.softwareag.com/Products/Documentation/default.asp*) for details on RPC servers and clients. The following table gives an overview of the connection types discussed above:

Connection Type	Connects to	Integration Server outbound	Integration Server inbound	IDL	Additional Nodes	Settings to generate	IDL Parameters
RPC Connection	EntireX Broker	X	-	one per IDL library	one adapter service for each IDL program	RPC Connection & Adapter Services	IN, INOUT, OUT
Reliable RPC Connection	EntireX Broker	X	-	one per IDL library	one adapter service for each IDL program	Reliable RPC Connection & Adapter Services	IN
RPC Listener Connection	EntireX Broker	-	X	one per IDL library	one adapter listener per IDL library	RPC Connection & Listener	IN, INOUT, OUT
Reliable RPC Listener Connection	EntireX Broker	-	X	one per IDL library	one adapter Listener per IDL library and for each IDL program a notification, a document type, a	Reliable RPC Connection & Listener	IN

Connection Type	Connects to	Integration Server outbound	Integration Server inbound	IDL	Additional Nodes	Settings to generate	IDL Parameters
					trigger, and a wrapping service		
Connection to IMS Connect	IMS Connect	X	-	one per IDL library (with one IDL program)	one adapter service for the IDL program	IMS Connect	IN, INOUT, OUT
CICS ECI Connection	CICS ECI	X	-	one per IDL library	one adapter service for each IDL program	CICS ECI Connection & Adapter Services	IN, INOUT, OUT
ApplinX Connection	ApplinX Server	X	-	one per ApplinX procedure group	one adapter service for each ApplinX path procedure	ApplinX Connection & Adapter Services	(invisible)
Direct RPC Connection	RPC Server	X	-	one per IDL library	one adapter service for each IDL program	Direct RPC Connection & Adapter Services	IN, INOUT, OUT
Direct RPC Listener Connection	PRC Client	-	X	one per IDL library	one adapter per IDL library	Direct RPC Connection & Listener	IN, INOUT, OUT
ACI Server Connection	EntireX BrokerXx	X	-	one per IDL library	one adapter service for each IDL program	ACI Server Connection & Adapter Services	IN, INOUT, OUT

Package Management

The EntireX Adapter is provided as a package called WmEntireX. The adapter package is managed the same way as any package on the Integration Server. There are several considerations when you set up and manage your packages on the Integration Server:

- Create user-defined packages for your connections and adapter services. See *EntireX Adapter Package Management* for details.

- Understand how package dependencies work so you make the best decisions regarding how you manage your connections and adapter services. See *Package Dependency Requirements and Guidelines* for details.

- Control which groups have access to which adapter services. See *Group Access Control* for details.

■ Enable and disable packages. See *Importing and Exporting Packages* for details.

Adapter Connections

The EntireX Adapter uses adapter connections to define how its adapter services and adapter notifications connect to EntireX RPC servers and clients. Each EntireX Adapter connection contains a set of connection parameters, including the logon parameters that the adapter requires to connect to the EntireX Broker. You configure *extended properties* of a connection using the IS Administration Console. For instructions on configuring and managing EntireX Adapter connections, see *EntireX Adapter Connections* (main section). EntireX Adapter connections can be used only for adapter services or only for adapter notifications. A connection cannot be used simultaneously by services and notifications. If this is required, duplicate a connection and use one connection for services and the other for notifications.

This section covers the following topics:

- Connection Pooling
- Runtime Behavior of Connection Pools
- Built-in Services for Connections

Connection Pooling

The Integration Server includes a connection management service that dynamically manages connections and connection pools based on configuration settings that you specify for the connection. All adapter services use connection pooling. A connection pool is a collection of connections with the same set of attributes. The Integration Server maintains connection pools in memory. Connection pools improve performance by enabling adapter services to reuse open connections rather than opening new connections.

Runtime Behavior of Connection Pools

When you enable a connection, the Integration Server initializes the connection pool, creating the number of connection instances you specified in the connection's `Minimum Pool Size` field. Whenever an adapter service needs a connection, the Integration Server provides a connection from the pool. If no connection is available in the pool, and the `Maximum Pool Size` has not been reached, the server creates one or more new connections (according to the number specified in `Pool Increment Size`) and adds them to the connection pool. If the pool is full (as specified in `Maximum Pool Size`), the requesting service will wait for the Integration Server to obtain a connection, up to the length of time specified in the `Block Timeout` field, until a connection becomes available. Periodically, the Integration Server inspects the pool and removes inactive connections that have exceeded the expiration period specified in `Expire Timeout`. You can enable the system to retry the initialization any number of times, at specified intervals. For information on configuring connections, see *EntireX Adapter Connections* (main section).

Built-in Services for Connections

Integration Server provides built-in services that enable you to programmatically control connections. You can use them to enable and disable a connection, and to return usage statistics, the current state ("Enabled" or "Disabled") and error status for a connection. These services are located in the WmART package, in the *pub.art.connection* folder. For details, see the *webMethods Integration Server Built-in Services Reference*.

Adapter Services

Adapter services enable the EntireX Adapter to initiate operations that interact with EntireX providers. You configure adapter services using adapter service templates, which are provided with the EntireX Adapter. Configuring a new service from an adapter service template is straightforward. There are two alternatives: use the EntireX Workbench or use the Designer. Using the EntireX Workbench is recommended. You create an adapter connection together with a set of adapter services, one for each IDL program in the IDL library you are using to create the connection. Using the Designer, you assign a default adapter connection to the service, select the adapter service template, and select the EntireX subprogram to call. This has to be done for each service separately.

After you configure a service, you can incorporate it into a flow or Java service to interact with an EntireX broker. For example, you can create a flow service that calls an EntireX RPC server.

Using Adapter Notifications

Adapter notifications are only needed for Reliable RPC Listener connections, sending RPC messages with only IN parameters to the Integration Server. The EntireX Adapter provides a notification called *Asynchronous Listener Notification*, which monitors a specified Reliable RPC Listener connection and notifies the adapter when an RPC call arrives at that connection. You configure the notification to invoke a flow or Java service using the data from the notification. See *Configuring Adapter Notifications* for instructions.

Using the Eclipse plug-ins of the adapter will generate all the objects needed for a Reliable RPC Listener connection automatically, including the listener and adapter notifications.

Viewing the Adapter's Update Level

You can view the list of updates that have been applied to the adapter. The list of updates appears in the Updates field on the "About page" in the IS Administration Console.

3 Package Management

This chapter describes how to set up and manage your EntireX Adapter packages, set up access control lists (ACLs).

EntireX Adapter Package Management

The EntireX Adapter is provided as a package called WmEntireX. This section covers the following topics:

- Package Dependency Requirements and Guidelines
- Enabling and Disabling Packages
- Importing and Exporting Packages

Package Dependency Requirements and Guidelines

This section contains a list of dependency requirements and guidelines for user-defined packages. For instructions on setting package dependencies, see *webMethods Service Development Help*.

- A user-defined package must be dependent on its associated adapter package, WmEntireX. (The WmEntireX package is dependent on the WmART package.)

- Package dependencies ensure that at startup the Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the user-defined package(s) last. The WmART package is automatically installed when you install the Integration Server. You should not need to reload the WmART package manually.

  Tip: When you create connections, adapter services, and adapter notifications, define them in user-defined packages rather than in the WmEntireX package. This makes it easier to manage packages. As you create user-defined packages in which to store connections, adapter services, and adapter notifications, use the package management functionality provided in the Designer and set the user-defined packages to be dependent on the WmEntireX package. That way, whenever the WmEntireX package loads or reloads, the user-defined packages load automatically as well.

- If the connections and adapter services of an adapter are defined in different packages, then:

  - a package that contains the connection(s) must have a dependency on the adapter package.

  - packages that contain adapter services must have a dependency on their associated connection package.

- Keep connections for different adapters in separate packages so that you do not create interdependencies between adapters. If a package contains connections for two different adapters, and you reload one of the adapter packages, the connections for both adapters will reload automatically.

■ The Integration Server will not allow you to enable a package if it has a dependency on another package that is disabled. That is, before you can enable your package, you have to enable all packages on which your package depends. For information on enabling packages, see *Enabling and Disabling Packages*.

■ The Integration Server will allow you to disable a package even if another package that is enabled has a dependency on it. Therefore, you have to manually disable any user-defined packages that have a dependency on the adapter package before you disable the adapter package. For information on disabling packages, see *Enabling and Disabling Packages*.

■ You can give connections, adapter services, and notifications the same name provided that they are in different folders and packages.

Enabling and Disabling Packages

All packages are automatically enabled by default. To temporarily prohibit access to the elements in a package, disable the package. When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents the Integration Server from loading that package at startup.

▶ **To enable a package**

1 Open the IS Administration Console if it is not already open.

2 In the **Packages** menu of the navigation area, click **Management**.

3 Click **No** in the **Enabled** column.

As a result, the server displays a tick and "Yes" in the **Enabled** column.

▯ **Note:** Enabling an adapter package will not cause its associated user-defined package(s) to be reloaded. For information on reloading packages, see the *webMethods Service Development Help*.

⚠ **Important:** Before you manually enable a user-defined package, you first have to enable its associated adapter package (WmEntireX). Similarly, if your adapter has multiple user-defined packages, and you want to disable some of them, disable the adapter package first. Otherwise, errors will be issued when you try to access the remaining enabled user-defined packages. See *Managing Packages* in the *webMethods Service Development Help* doumentation or choose **Software AG Designer Guides** > **webMethods Service Development Help** > **Managing Packages** in the Software AG Designer online help.

▶ **To disable a package**

1 Open the IS Administration Console if it is not already open.

2 In the **Packages** menu of the navigation area, click **Management**.

3   Click **Yes** in the **Enabled** column for the package that you want to disable. The server issues a prompt to verify that you want to disable the package. Click **OK** to disable the package. When the package is disabled, the server displays "No" in the **Enabled** column.

A disabled package will:

■ remain disabled until you explicitly enable it using the IS Administration Console.

■ not listed in the Designer.

Importing and Exporting Packages

To export packages, use the Designer. You can export the package to a ZIP file and save it to your hard drive. The ZIP file can then be imported for use by another Integration Server.

⚠   **Important:** Do not rename packages you export. The rename function is comparable to moving a package: when you import the renamed package, you lose any triggers, connections, and notifications associated with this package. For details on managing packages, see *webMethods Service Development Help*.

Group Access Control

To control which development group has access to which adapter services, use access control lists (ACLs). You can use ACLs to prevent one development group from inadvertently updating the work of another group, or to allow or deny access to services that are restricted to one group but not to others. For general information on assigning and managing ACLs, see *webMethods Service Development Help*.

4   **EntireX Adapter Connections**

This chapter describes how to configure and manage EntireX Adapter connections.

💡 **Tip:** It is recommended to create connections with the webMethods Integration Server Wrapper of the EntireX Workbench. Connections which are used dynamically can be created with the IS Administration Console.

Before Configuring or Managing Adapter Connections

▶ **To prepare to configure or manage adapter connections**

1 Install the webMethods Integration Server and the EntireX Adapter on the same machine, using the Software AG Installer.

2 Make sure you have webMethods administrator privileges so that you can access the EntireX Adapter's administrative screens. See the *Administering webMethods Integration Server* for information on setting user privileges.

3 Start your Integration Server and the IS Administration Console, if they are not already running.

4 Using the IS Administration Console, make sure the WmEntireX package is enabled. See *Importing and Exporting Packages* for instructions.

5 If you have not already done so, use the Designer to create a user-defined package to contain the connection. For more information on managing packages for the adapter, see *EntireX Adapter Package Management*.

Configuring Adapter Connections

When you configure EntireX Adapter connections, you specify information that the Integration Server uses to connect to an EntireX RPC server or client, or to IMS Connect or CICS ECI. You create EntireX Adapter connections using the EntireX Workbench. This ensures that the information generated from the interface definition stored in the Software AG IDL file is present. A wizard inside the EntireX Workbench allows you to configure the most common parameters of the connection. Using the IS Administration Console, you have access to all parameters of a connection.

The EntireX Adapter supports the following connection types:

1. EntireX RPC Connection

2. EntireX Reliable RPC Connection

3. EntireX Direct RPC Connection

4. EntireX RPC Listener Connection

5. EntireX Reliable RPC Listener Connection

6. EntireX Direct RPC Listener Connection

7. Connection to IMS Connect

8. CICS ECI Connection

9. ApplinX Connection

10. ACI Server Connection

For connections of type "EntireX RPC", "EntireX Reliable RPC", "ApplinX", "EntireX Direct RPC", "IMS Connect", "CICS ECI", and "ACI Server", the EntireX Workbench wizard additionally creates an EntireX Adapter service for each program of the IDL file; see *EntireX Adapter Services* for further information.

For connections of type "EntireX RPC Listener", "EntireX Direct RPC Listener" and "EntireX Reliable RPC Listener", the EntireX Workbench wizard additionally creates an EntireX Adapter listener; see also *Listeners and Adapter Notifications*.

▶ **To create a connection**

1   Start the EntireX Workbench.

2   Navigate to the Software AG IDL file you want to use.

3   On the IDL file, right-click and choose **Generate webMethods IS Connection from Software AG IDL...**

4   Follow the wizard (Integration Server Wrapper), which guides you through the creation of the adapter connection and the adapter services or adapter listener.

    As a result, the connection you created appears on the adapter's connections screen and in the Designer **Package Navigator** view.

    Note: It is not possible to generate new connections using **Configure New Connection** on the **Connections screen** of the EntireX Adapter in the IS Administration Console because such connections would not have any metadata. Only connections generated by the Designer have the mandatory metadata. However, it is possible to create a new connection by copying an existing connection on the **Connections screen**. This is useful for dynamically selected connections where the metadata is not generated from the interface definition (Software AG IDL file). To use dynamically selected connections with $connectionName, see chapter 8 in *webMethods Adapter Development Kit User's Guide Version 6.5*, see *https://empower.softwareag.com/Products/Documentation/default.asp* under "webMethods" > "Adapters".

Viewing Adapter Connection Parameters

You can view a connection's parameters using the IS Administration Console or the Designer.

▶ **To view the parameters for a connection using the IS Administration Console**

1   In the **Adapters** menu in the IS Administration Console navigation area, click **EntireX Adapter**.

2   On the **Connections** screen, click the **View** icon for the connection you want to see. The **View Connection** screen displays the parameters for the connection. For descriptions of the connection parameters, see *Configuring Adapter Connections*.

3   Click **Return to EntireX Adapter Connections** to return to the main connections screen.

▶ **To view the parameters for a connection using the Designer**

1   Start the Designer if it is not already running.

2   From the Designer **Package Navigator** view, open the package and folder in which the connection is located.

3   Click the connection you want to view. The parameters for the connection appear on the **Connection Information** tab. For descriptions of the connection parameters, see *Configuring Adapter Connections*.

Editing Adapter Connections

If the login information for a server changes, or if you want to redefine parameters that a connection uses when connecting to a server, you can update the connection's parameters. This section describes how to edit a connection and lists parameter settings for the connection types covered. The following topics are covered:

- Editing a Connection
- Connection Parameters for RPC Connections
- Connection Parameters for Reliable RPC Connections
- Connection Parameters for RPC Listener Connections and Reliable RPC Listener Connections
- Connection Parameters for Direct RPC Connections
- Connection Parameters for Direct RPC Listener Connections
- Connection Parameters for ACI Server Connections
- Connection Parameters for Connections to IMS Connect
- Connection Parameters for CICS ECI Connections
- Connection Parameters for ApplinX Connections

■ Common Connection Management Properties (for all Connection Types)

Editing a Connection

You edit (and modify) adapter connections using the IS Administration Console.

▶ **To edit a connection**

1   In the **Adapters** menu in the IS Administration Console navigation area, click **EntireX Adapter**.

2   Make sure that the connection is disabled before editing it. See *Disabling Adapter Connections* for instructions.

3   On the **Connections** screen, click the **Edit** icon for the connection you want to edit. The **Edit Connection** screen displays the current parameters for the connection. Update the connection's parameters by typing or selecting the values you want to specify. The tables below give a complete listing of parameters for the following connection types:

   ■ *"RPC Connection"*
   ■ *"Reliable RPC Connection"*
   ■ *"RPC Listener Connections"* and *" Reliable RPC Listener Connections"*
   ■ *"Direct RPC Connection"*
   ■ *"Direct RPC Listener Connection"*
   ■ *"ACI Server Connection"*
   ■ *"IMS Connect"*
   ■ *"CICS ECI "*
   ■ *"ApplinX Connections"*

4   On the Connections screen, edit the connection management properties (connection pooling). The tables in *Common Connection Management Properties (for all Connection Types)* give a complete listing of parameters.

5   Click **Save Changes** to save the connection and return to the Connections screen.

Connection Parameters for RPC Connections

Parameter	Description	Default	Can be set from the EntireX Workbench	Can be set with service
Broker ID	The ID of the broker you want to connect to. This ID consists of a host and an optional port. Default for the port is 1971.	localhost	Yes	No
Server Address	The address of the RPC server registered to the broker above. The address is given in the format `<class>/<server>/<service>`.	RPC/SRV1/CALLNAT	Yes	Yes
Logon User	The name of the user to log on to the broker.		Yes	Yes
Logon Password	The password for the user above.		Yes	Yes
Retype Logon Password	Verification of the password.		Not applicable	Not applicable
Encryption	The level of broker encryption to be used for the RPC call.	No_Encryption	No	No
Encoding	The character encoding to be used for the RPC connection in the Integration Server. Default: the encoding of the Integration Server.		Yes	No
Compression Level	The level of broker data compression.	NO_COMPRESSION	No	No
Timeout	The number of seconds to wait for a response from the RPC server.	60S	No	No
RPC Library Name	The name of the RPC library. Default: the library name used to generate the connection. Mainly used to override the library name for Natural RPC servers.[1]		No	Yes
RPC User	The RPC user for Natural Security.[1]		No	Yes
RPC Password	The password for the user above.[1]		No	Yes
Retype RPC Password	Verification of the password above.		Not applicable	Not applicable
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated	No
Transaction Type	The transaction type, "local transactions" and "no transactions" are supported.	none	No	No

📄 **Notes:**

1. Depending on the server configuration, some Natural RPC servers need the RPC library name. See the documentation of the Natural RPC Server at *https://empower.softwareag.com/Products/Documentation/default.asp* for details.

The transaction type "local" of the Integration Server is mapped to conversational RPC for RPC connections and Reliable RPC with client-commit for Reliable RPC connections. See Appendix B in *webMethods Adapter Development Kit User's Guide Version 6.5* for transactions of the Integration Server. XA transactions are not supported by the EntireX Adapter.

Connection Parameters for Reliable RPC Connections

Parameter	Description	Default	Can be set from the EntireX Workbench	Can be set dynamically with service
Broker ID	The ID of the broker you want to connect to. This ID consists of a host and an optional port. Default for the port is "1971".	localhost	Yes	No
Server Address	The address of the RPC server registered to the broker above. The address is given in the format `<class>/<server>/<service>`.	`RPC/SRV1/CALLNAT`	Yes	Yes
Logon User	The name of the user to log on to the broker.		Yes	Yes
Logon Password	The password for the user above.		Yes	Yes
Retype Logon Password	Verification of the password.		Not applicable	Not applicable
Encryption	The level of broker encryption to be used for the RPC call	No_Encryption	No	No
Encoding	The character encoding to be used for the RPC connection in the Integration Server. Default: the encoding of the Integration Server.		Yes	No
Compression Level	The level of broker data compression.	NO_COMPRESSION	No	No
RPC Library Name	The name of the RPC library. Default: the library name used to generate the connection. Mainly used to override the library name for Natural RPC servers.		No	Yes
RPC User	The RPC user for Natural Security.		No	Yes
RPC Password	The password for the user above		No	Yes

Parameter	Description	Default	Can be set from the EntireX Workbench	Can be set dynamically with service
Retype RPC Password	Verification of the password above.		Not applicable	Not applicable
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated	No
Transaction Type	The transaction type. "local transactions "and "no transactions" are supported.	none	No	No

Connection Parameters for RPC Listener Connections and Reliable RPC Listener Connections

Parameter	Description	Default	Can be set from the EntireX Workbench
Broker ID	The ID of the broker you want to connect to. This ID consists of a host and an optional port. Default for the port is "1971".	localhost	Yes
Server Address	The address of the RPC server registered to the broker above. The address is given in the format `<class>/<server>/<service>`.	RPC/SRV1/CALLNAT	Yes
Logon User	The name of the user to log on to the broker.		Yes
Logon Password	The password for the user above.		Yes
Retype Logon Password	Verification of the password.		Not applicable
Encryption	The level of broker encryption to be used for the RPC call.	No_Encryption	No
Encoding	The character encoding to be used for the RPC connection in the Integration Server. Default: the encoding of the Integration Server.		Yes
Compression Level	The level of broker data compression.	NO_COMPRESSION	No
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated

Connection Parameters for Direct RPC Connections

Parameter	Description	Default	Can be set from the EntireX Workbench	Can be set dynamically with service
Server Address	The address of the RPC server. The address is given in the format `<class>/<server>/<service>`.	RPC/SRV1/CALLNAT	Yes	Yes
Timeout	The number of seconds to wait for a response from the RPC server.	60S	No	No
RPC Library Name	The name of the RPC library. Default: the library name used to generate the connection. Mainly used to override the library name for Natural RPC servers.		No	Yes
RPC User	The RPC user for Natural Security.		No	Yes
RPC Password	The password for the user above.		No	Yes
Retype RPC Password	Verification of the password above.		Not applicable	Not applicable
Metadata Information	Caution: Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated	No
Transaction Type	The transaction type "local transactions" and "no transactions" are supported.	none	No	No

The transaction type "local" of the EntireX Adapter connection is mapped to conversational RPC for RPC connections and Reliable RPC with client-commit for Reliable RPC connections. See Appendix B in *webMethods Adapter Development Kit User's Guide*, Version 6.5 for transactions of the Integration Server. XA transactions are not supported by the EntireX Adapter.

Connection Parameters for Direct RPC Listener Connections

Parameter	Description	Default	Can be set from the EntireX Workbench
Server Address	The address of the RPC server. The address is given in the format `<class>/<server>/<service>`.	RPC/SRV1/CALLNAT	Yes
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated

Connection Parameters for ACI Server Connections

Parameter	Description	Default	Can be set from the EntireX Workbench
Broker ID	The ID of the broker you want to connect to. This ID consists of a host and an optional port. Default for the port is 1971.	localhost	Yes
Server Address	The address of the RPC server registered to the broker above. The address is given in the format <class>/<server>/<service>. The address may contain an asterix ('*') as a wildcard which is substituted by the IDL program name at runtime. This helps you to use only one connection for multiple IDL programs or adapter services.	RPC/SRV1/CALLNAT	Yes
Logon User	The name of the user to log on to the broker.		Yes
Logon Password	The password for the user above.		Yes
Retype Logon Password	Verification of the password.		Not applicable
Encryption	The level of broker encryption to be used for the RPC call.	No_Encryption	No
Encoding	The character encoding to be used for the RPC connection in the Integration Server. Default: the encoding of the Integration Server.		No
Timeout	The number of seconds to wait for a response from the RPC server.	60S	No
Server Type	Defines how arrays of groups are marshalled. Allowed values are "COBOL" and "Natural". **Note:** See *Writing ACI Servers for the RPC-ACI Bridge in COBOL/Natural* in the *EntireX RPC-ACI Bridge* documentation for restrictions concerning ACI servers.	COBOL	No
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated

Connection Parameters for Connections to IMS Connect

Parameter	Description	Default	Can be set from EntireX Workbench	Can be set dynamically with service
Host	Hostname of IMS Connect.		Yes	No
Port	IMS Connect Port.		Yes	No
IMS Connect Datastore ID	Name of the data store, as defined in the IMS Connect configuration member.		Yes	Yes
Encoding	The character encoding used to call IMS Connect. It is recommended to specify the appropriate EBCDIC encoding.	cp037	Yes, default value is cp037	No
Socket Timeout [1]	Socket timeout (in milliseconds). The minimum value is 2.000 ms. If a lower value is specified, it will be changed to 2.000.	10000	No	No
Check for DFS Errors	Return an error and do not return the message if it might contain a DFS error message.	true	No	No
Logical Terminal ID	IMS Connect logical terminal ID, maximal 8 bytes (optional).		No	Yes
Use old Exit	Use old IMS Connect user message exit if set to "true". The old exit name is "*SAMPLE*", the new exit name is "*SAMPL1*".	true	No	No
Exit Name	Optional name of the IMS Connect user message exit.	Default name for old exit is "*SAMPLE*", for new exit "*SAMPL1*"	No	No
RACF User ID	The name of the user to log on to IMS Connect.		Yes	Yes
RACF Password / PassTicket	The password of the user above.		Yes	Yes
Retype RACF Password / PassTicket	Verification of the password above.		Not applicable	Not applicable
RACF Group Name	Security setting		No	Yes
RACF Application Name	Defined to RACF on the PKTDATA definition.		No	Yes
SSL Parameters	Truststore and optional certificate. Example: `trust_store=CACerts.jks`		No	No
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file		Generated	No

Parameter	Description	Default	Can be set from EntireX Workbench	Can be set dynamically with service
	name, library name, and version of the EntireX Workbench.			
Use IDL program as transaction name	Automatically use the IDL program name as transaction name. If set to "true" or "yes", 10 bytes are used for the transaction name. If set to a number, this number of bytes is used for the transaction name.	false	No	No

📄 **Notes:**

1. The socket timeout value is used to set the IMS Connect timer value. The timer value specifies the delay that IMS Connect will wait until IMS returns data to IMS Connect, which in turn will be sent to the adapter. If the socket timeout is less than one minute, the IMS timer value is one second less than the socket timeout. If the socket timeout is less than one hour, the IMS timer value is one minute less than the socket timeout. The maximum IMS timer value is 60 minutes.

Connection Parameters for CICS ECI Connections

Parameter	Description	Default	Can be set from EntireX Workbench	Can be set dynamically with service
Host	Hostname of CICS.		Yes	No
Port	CICS Port		Yes	No
CICS Mirror Transaction ID	Name of the CICS mirror transaction. Default is "CPMI", which is the default dispatching transaction for ECI.	CPMI	Yes	Yes
Encoding	The character encoding used to call CICS ECI. It is recommended to specify the appropriate EBCDIC encoding	cp037	Yes	No
Socket Timeout	Socket timeout (in milliseconds)	10000	No	No
RACF User ID	The name of the user to log on to CICS ECI.		Yes	Yes
RACF Password / PassTicket	The password for the user above.		Yes	Yes
Retype RACF Password / PassTicket	Verification of the password above		No	No
SSL Parameters	Truststore and optional certificate. Example: `trust_store=CACerts.jks`		No	No
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file		Generated	No

Parameter	Description	Default	Can be set from EntireX Workbench	Can be set dynamically with service
	name, library name, and version of the EntireX Workbench.			

Connection Parameters for ApplinX Connections

Parameter	Description	Default	Can be set from EntireX Workbench
Host	Hostname of ApplinX server.		Yes
Port	ApplinX server Port.		Yes
Secure connection	Set to "true" for a secure connection.	false	No
Metadata Information	**Caution:** Do not change this generated property. It contains timestamp, IDL file name, library name, and version of the EntireX Workbench.		Generated

Common Connection Management Properties (for all Connection Types)

Parameter	Description
Enable Connection Pooling	Enables the adapter to use connection pooling. Default: `true`. See *EntireX Adapter Connections* for more information on connection pooling.
Minimum Pool Size	If connection pooling is enabled, this field specifies the minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it creates this number of connections. Default: `1`.
Maximum Pool Size	The maximum number of connection objects that can exist in the connection pool. The adapter will reuse any inactive connections in the pool or, if all connections are active and the connection pool has reached its maximum size, the adapter will wait for a connection to become available. Default: `10`.
Pool Increment Size	If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. Default: `1`.
Block Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that the Integration Server will wait to obtain a connection before it times out and returns an error. Default: `1000`.
Expire Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. For example, to specify 10 seconds, specify 10000. Enter 0 to specify no timeout. Default: `1000`. **Note:** The adapter will never violate the `Minimum Pool Size` parameter. These connections remain in the pool regardless of how long they are inactive.

Parameter	Description
Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. Default: 0.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

Deleting Adapter Connections

If you no longer want to use a particular EntireX Adapter connection, you can delete it by following the instructions in this section. You delete adapter connections using the IS Administration Console. If you delete an EntireX Adapter connection, the adapter services that are defined to use the connection will no longer work. You can change the connection an adapter service uses. Therefore, if you delete an EntireX Adapter connection, you can assign a different connection to an adapter service and reuse the service.

▶ **To delete a connection**

1   In the **Adapters** menu in the IS Administration Console navigation area, click **EntireX Adapter**.

2   Make sure that the connection is disabled before deleting. To disable the connection, click **Yes** in the Enabled column and click **OK** to confirm. The Enabled column now shows "No" (Disabled) for the connection.

3   On the Connections screen, click the **Delete** icon for the connection you want to delete.

As a result, the Integration Server deletes the adapter connection.

Enabling Adapter Connections

An EntireX Adapter connection must be enabled before you can configure any adapter service using the connection, or before an adapter service can use the connection at runtime. You enable adapter connections using the IS Administration Console.

> **Note:** When you reload a package that contains enabled connections, the connections will automatically be enabled when the package reloads. If the package contains connections that are disabled, they will remain disabled when the package reloads.

▶ **To enable a connection**

1   In the **Adapters** menu in the IS Administration Console navigation area, click **EntireX Adapter**.

2   On the **Connections** screen, click **No** in the **Enabled** column for the connection you want to enable.

As a result, the IS Administration Console enables the adapter connection and displays a tick and "Yes" in the **Enabled** column.

> **Note:** The EntireX broker configured in the Connections does not need to be running to enable a connection. Only the parameters are checked for correctness while enabling a connection.

Disabling Adapter Connections

EntireX Adapter connections must be disabled before you can edit or delete them. You disable adapter connections using the IS Administration Console.

▶ **To disable a connection**

1   In the **Adapters** menu in the IS Administration Console navigation area, click **EntireX Adapter**.

2   On the **Connections** screen, click **Yes** in the **Enabled** column for the connection you want to disable.

As a result, the adapter connection becomes disabled and you see a "No" in the **Enabled** column.

5 EntireX Adapter Services

This chapter describes how to configure and manage EntireX Adapter services. EntireX Adapter services allow clients to interact with EntireX RPC servers, Natural RPC servers, ACI servers, ApplinX, IMS Connect, and CICS ECI using a configured adapter connection. EntireX Adapter services are available for synchronous and asynchronous RPC calls (RPC and reliable RPC).

Before Configuring or Managing Adapter Services

▶ **To prepare configuration and management of EntireX Adapter services**

1   Start your Integration Server and the IS Administration Console, if they are not already running.

2   Make sure you have webMethods administrator privileges so that you can access the EntireX Adapter's administrative screens. See the *Administering webMethods Integration Server* for information on setting user privileges.

3   Using the IS Administration Console, make sure the WmEntireX package is enabled. See *Enabling and Disabling Packages* for instructions.

4   Start the Designer if it is not already running.

5   Make sure you are viewing the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer help.

6   Using the Designer, create a user-defined package to contain the service, if you have not already done so. Add a dependency on WmEntireX package (version *.*) to your user-defined package. When you configure adapter services, you should always define them in user-defined packages rather than in the WmEntireX package. For more information on managing packages for the adapter, see *Package Management*.

7   Generate the adapter connection with the EntireX Workbench.

    Or:

    Using the IS Administration Console, configure an adapter connection to use with the adapter service. See *Configuring Adapter Connections* for instructions.

Configuring Adapter Services

The EntireX Adapter service enables you to send RPC calls to an EntireX RPC server or Natural RPC server. You configure EntireX Adapter services using a specific adapter connection. After you follow the steps in this section to configure the adapter service, you can invoke it from a flow or Java service. To use EntireX Adapter services, you provide values for the service's input signature in the pipeline. For more information on using EntireX Adapter services, see *Using Adapter Services*.

▶ **To create EntireX Adapter services**

1   In the EntireX Workbench, select the Software AG IDL file to use.

2   In the context menu, choose **Generate webMethods IS Connection from Software AG IDL....**

Using Adapter Services

▶ **To enable the EntireX Adapter service to send an RPC call to an EntireX RPC server**

■   Provide values for the parameters in the service's input signature when configuring the adapter service within a flow service in the Designer. You can either map in values from the pipeline, or you can set constant values using the **Pipeline View**.

This has the following effect on the input and output signature:

▪ The input signature of an adapter service wraps all input parameters of the EntireX subprogram in the `inRec`.

A parameter `throwException` is added to the signature. If this optional parameter is set to "true", errors are returned as exceptions and not by means of parameters in the output signature. A parameter `$connectionname` is added to the signature. If this optional parameter

A parameter `throwException` is added to the signature. If this optional parameter is set to "true", errors are returned as exceptions and not by means of parameters in the output signature.

A parameter `$connectionname` is added to the signature. If this optional parameter, is set to a connection name, it overrides the default connection name of the service.

Depending on the connection type, adapter services have optional parameters that can be used to overwrite the corresponding values of the connection. These optional parameters are:

- **CICS ECI Connection**
  - RACF user ID
  - RACF password/PassTicket
  - CICS Mirror Transaction ID
- **EntireX Direct RPC Connection**
  - RPC library name
  - RPC user
  - RPC password
  - Server address
- **EntireX RPC Connection / EntireX Reliable RPC Connection**
  - Logon user
  - Logon password
  - RPC library name
  - RPC user
  - RPC password
  - Server address
- **IMS Connect Connection**
  - Logical terminal ID
  - RACF user ID
  - RACF password/PassTicket
  - RACF group name
  - RACF application name
  - IMS datastore ID
  - Maximum segment length
- **EntireX ACI Connection**
  - Logon user
  - Logon password

  See **EntireX Adapter Connections** for details on the parameters.

- The output signature of an adapter service wraps all output parameter of the EntireX sub-program in the `outRec`. Three parameters `errorFlag`, `errorMessage` (optional), and `errorCode` are added to the signature. If `errorFlag` is true, `errorMessage` contains the error message. `errorCode` contains "00000000" for a successful call, otherwise the EntireX message class (4 digits) and message code (4 digits).

Viewing Adapter Services

Use the Designer to view adapter services. Make sure you are viewing the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help.

▶ **To view a service**

1   In the Designer **Package Navigator** view, expand the package and folder that contain the service you want to view.

2   Select the service you want to view. The Designer displays the service in the **Service Editor**.

As a result, the Adapter Service Editor for the EntireX Adapter service appears.

📄   **Notes:**

1. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

2. On the **RPC Call** tab, the EntireX Subprogram Name field holds the name of the IDL program called by this service. The tables **Input Parameter** and **Output Parameter** display names, the IDL types, and the parameter types in the pipeline.

3. The **Input/Output** tab shows the service's input and output signatures. To use the EntireX Adapter service, you provide values for the signature input within the service pipeline in the Designer. See *Using Adapter Services* for instructions on using the EntireX Adapter service.

4. For information on configuring the **Input/Output** and **Adapter Settings** tabs, see *webMethods Service Development Help*. These tabs apply to all services that you configure using the Designer. The information on **Audit** and **Permissions** appears in the **Properties** panel. For the results, see the **Results** panel.

Deleting Adapter Services

Use the Designer to delete adapter services. Make sure you are viewing the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help.

▶ **To delete a service**

1   In the Designer **Package Navigator** view, expand the package and folder that contain the service you want to delete.

2   Right-click the service and then click **Delete**.

Runtime Connection Allocation for Adapter Services

Adapter services of the EntireX Adapter are enabled for dynamically selecting a connection node. The connection name may be specified on the pipeline in the `$connectionName` field. This field has to be part of the services input signature. If `$connectionName` is not specified, the default connection name is used, usually the connection used to create the service.

See chapter 8 in *webMethods Adapter Development Kit User's Guide* , version 6.5 for more details on connection selection.

Connection selection is applicable to services that are generated with the EntireX Adapter (version 8.0 or higher) and helps to:

- **Switch between development and test environment**
  - Generate the connections and services as usual.
  - Design your services that call the adapter services to set `$connectionName` to one single connection.
  - Then you can switch to a different environment by changing this single connection to a different resource.
- **Reduce the number of connection pools**
  - Generate the connections and services as usual.
  - Then for each IDL library one connection is generated and one connection pool is used at runtime.
  - Design your services that call the adapter services to set `$connectionName` to one single connection.
  - This reduces the connection pools to a single pool associated with this connection.

Configure the Formatting of Decimal and Alphanumeric Parameters

Adapter services have two parameters to configure the formatting of decimal parameters (IDL data types N, NU, P and PU, and alphanumeric parameters of fixed length (IDL data types A, K, and U).

To set these options, use the Designer and open the adapter service for the EntireX Adapter. In **RPC Call**, select **Parameters for Decimal Numbers (N/P)** or **Parameters for fixed length Strings (A/K/U)**.

The following tables give an overview of parameters to be configured:

- **Parameters for Decimal Numbers (N/P)**
  (As an example, we use the value "3.14 " for a parameter defined as `N7.2` in the IDL.)

Parameter	Example
Remove leading zeros & keep decimal point (default)	3.14
Remove leading zeros & remove decimal point	314
Keep leading zeros & keep decimal point	0000003.14
Keep leading zeros & remove decimal point	000000314

- **Justify Decimal Numbers (N/P)**

  Note: This parameter is highly deprecated and should be only used for compatibility with webMethods Mainframe.

Parameter	Explanation
not justified (default)	The field formatting is unchanged.
left justified	The numeric field is left justified.
right justified	The numeric field is right justified.

- **Parameters for fixed length Strings (A/K/U)**

Parameter	Explanation
Remove whitespace characters (default)	Trim leading and trailing whitespace characters.
Keep whitespace characters	Format the string with the defined number of characters.

- **Justify fixed length Strings (A/K/U)**

  Note: This parameter is highly deprecated and should be only used for compatibility with webMethods Mainframe.

Parameter	Explanation
not justified (default)	The field formatting is unchanged.
left justified	The string field is left justified.
right justified	The string field is right justified.

Configuring Null Value Suppression

Null value suppression (NVS) allows you to suppress parameters with no specified value. Suppressed parameters will not appear in the result of the adapter service call.

Null value suppression is available for the following IDL data types:

- All string data types
- Groups and structures
- Arrays

To activate NVS in the context of adapter services, you need to modify each adapter service seperately using Designer. For each adapter service, select **Null Value Suppression** in **RPC Call**.

To fine tune NVS, the following suppression modes can be selected:

Suppression Mode for Parameters	Explanation
No suppression (default)	No suppression, all parameters are present in the output structure.
Suppress null elements	String parameters with empty strings are suppressed, empty structures and arrays are suppressed.
Suppress null elements and trim array cells at end	Same as above. In addition, all empty array elements at the end of the array will be removed from the array.
Suppress null elements and trim array cells at end containing null elements and zero numbers	Same as above. In addition, numbers of numeric/decimal type (I1, I2, I4, N, P) with the value zero are treated as an empty value.

A string parameter is suppressed if the assigned string is:

- of type variable length (AV/KV/UV) and the string is empty (same for variable length strings with a maximum length).
- of type fixed length (A/K/V) and the parameter **Remove whitespace characters** is enabled, and trimming the leading and the trailing whitespace characters would result in an empty string.

6 Listeners and Adapter Notifications

Listeners are used for inbound connections to the Integration Server. They are needed for RPC Listener connections, Reliable RPC Listener connections, and Direct RPC Listener connections.

Adapter notifications monitor an EntireX RPC service and notify the Integration Server when a message arrives on the EntireX RPC service for Reliable RPC.

⚠️ **Important:** Use the Integration Server Wrapper of the EntireX Workbench to create RPC Listener connections with listeners and Reliable RPC Listener connections with listeners, notifications, triggers, and document types. This chapter describes steps to manually configure these objects. This is needed for advanced configurations, for example triggers with filters.

Remarks on Reliable RPC Listener Connections

A generated trigger object needs a trigger service that it can call. The input signature for a trigger service needs to have a document reference to the generated publishable document type. While an Integration Server service normally does not have such an input signature, we have to use a workaround to enable the Integration Server service to retrieve the input data from the publishable document. Therefore you can use the service `com.softwareag.entirex.wmadapter.Deployment:putDocDataToPipeline`, which retrieves all input data from the incoming document and puts it in the pipeline where the Integration Server service is able to retrieve the data. The following section provides a detailed description on how to use the `putDocDataToPipeline` service.

■ Suppose you have a flow service *Service:Flow1* which simply dumps the pipeline to a file:
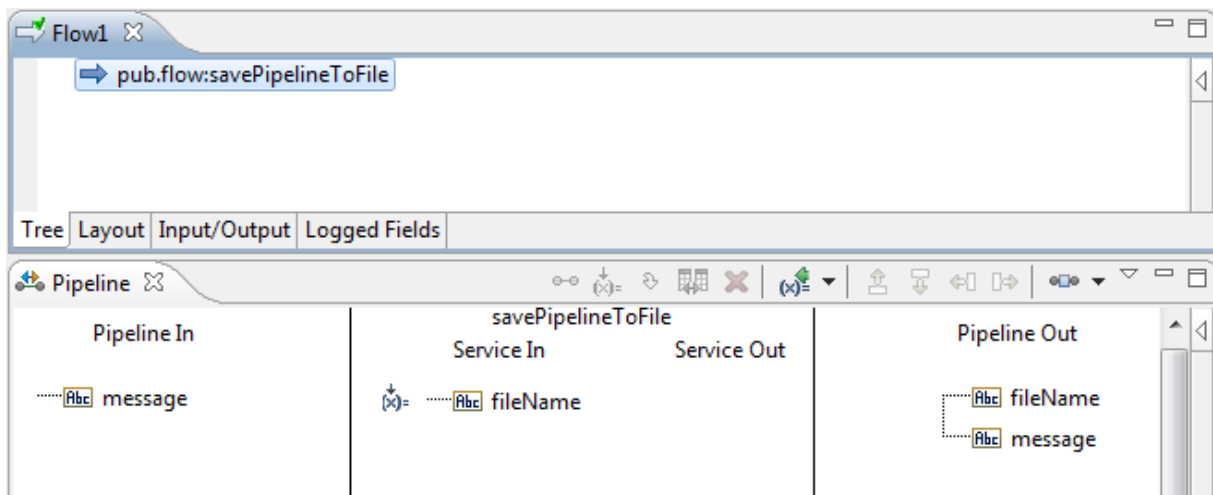
■ The Software AG IDL Extractor for webMethods Integration Server extracts the following IDL:

```
Library 'RelListener' Is

    Program 'Flow1' Is  /* Service:Flow1
        Define Data Parameter
        1 message (AV) in
        End-Define
```

■ Generating a Reliable RPC Listener connection, a listener, a notification, a trigger, and a publishable document results in:
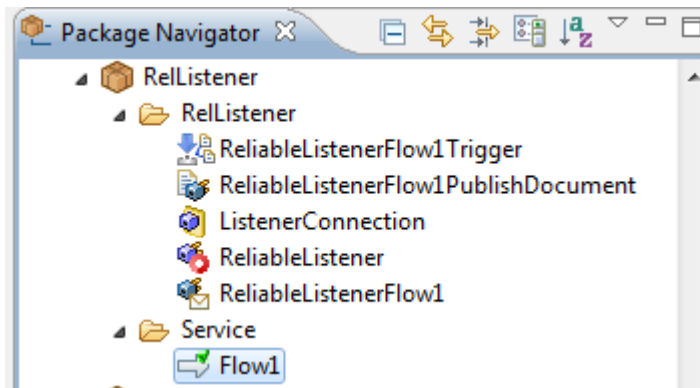
■ For the original Service:Flow1, the pipeline looks as follows:

```
-<IDataXMLCoder version="1.0">
  -<record javaclass="com.wm.data.ISMemDataImpl">
      <value name="fileName">Flow1-Pipeline.txt</value>
    -<record name="RelListener:ReliableListenerFlow1PublishDocument" javaclass="com.wm.data.ISMemDataImpl">
        <value name="message">abc1</value>
      +<record name="_env" javaclass="com.wm.data.ISMemDataImpl"></record>
      </record>
    </record>
  </IDataXMLCoder>
```

As a result, all parameters of the IDL program are inside the record for the document.

■ Now add the invoke node for
com.softwareag.entirex.wmadapter.Deployment:putDocDataToPipeline to Flow1:
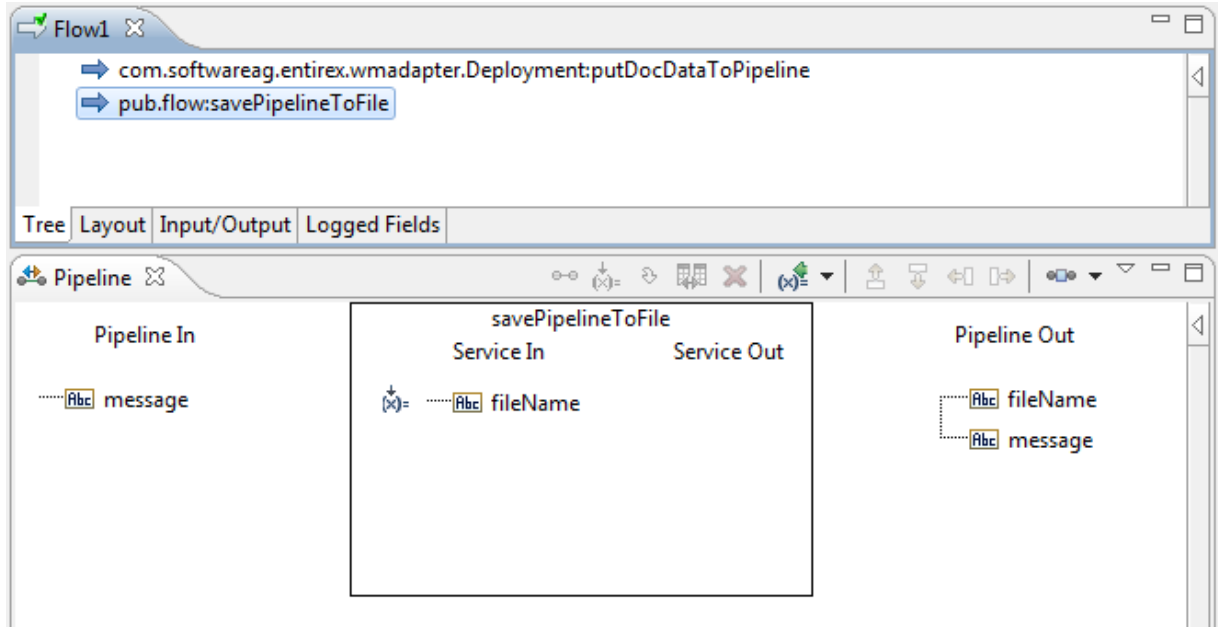
- With the `com.softwareag.entirex.wmadapter.Deployment:putDocDataToPipeline` service the pipeline looks as follows:

```
−<IDataXMLCoder version="1.0">
   −<record javaclass="com.wm.data.ISMemDataImpl">
      <value name="fileName">Flow1-Pipeline.txt</value>
     −<record name="RelListener:ReliableListenerFlow1PublishDocument" javaclass="com.wm.data.ISMemDataImpl">
         <value name="message">abc1</value>
       +<record name="_env" javaclass="com.wm.data.ISMemDataImpl"></record>
      </record>
      <value name="message">abc1</value>
     +<record name="_env" javaclass="com.wm.data.ISMemDataImpl"></record>
   </record>
</IDataXMLCoder>
```

The parameters of the IDL program (here `message`) are on the top level of the pipeline. Also the record `_env` is on the top level of the pipeline. This enables the service to use the parameters directly.

Before Configuring New Listeners

1. Start the Integration Server and the IS Administration Console if they are not already running.

2. Make sure that you have webMethods administrator privileges so that you can access the EntireX Adapter's administrative screens.

> **Note:** For information on setting user privileges, see *IS Administration Console's Guide*.

   a. Using the IS Administration Console, make sure that the WmEntireX package is enabled. To verify the status of the WmEntireX package, see *Enabling and Disabling Packages*.

   b. Using the IS Administration Console, configure a valid connection, for instance by copying a generated connection. For more information on configuring connections, see *EntireX Adapter Connections* (main section).

   c. Start the Designer if it is not already running.

   d. Make sure you are viewing the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help.

   e. Using the Designer, create a user-defined package to contain the listener, if you have not already done so. For more information on managing packages, see *Package Management*.

Configuring New Listeners

When you configure listeners of the EntireX Adapter, you specify information that the Integration Server uses to listen for, and get messages on, a specified EntireX broker. The broker is specified in the connection associated with the listener. You configure EntireX Adapter Listeners using the IS Administration Console.

▶ **To configure a new listener**

1   In the **Adapters** menu in the IS Administration Console's navigation area, click **EntireX Adapter**.

2   In the **EntireX Adapter** menu, click **Listeners**.

3   On the **Listeners** screen, click **Configure New Listener**.

4   On the **Listener Types** screen, select EntireX Listener.

5   On the **Configure Listener Type** screen, in the EntireX Adapter section, use the following fields:

Field	Description
Package	The package in which to create the listener.
Folder Name	The folder in which to create the listener.
Listener Name	The name of the listener.
Connection Name	The connection to associate with the listener. From the list of connections, select an appropriate connection that you created on the **Configure Connections Type** screen. Use separate connections for the adapter services and the adapter listeners.
Retry Limit	The number of times the adapter tries to reconnect if the adapter fails to connect, or loses connection with the EntireX broker. Default: 5.
Retry Backoff Timeout	The number of seconds that elapse between each of the retries specified in the retry limit. Default: 10.
EntireX Subprogram Names	A list of subprogram names, separated by blanks. This list and the following list of service names is usually generated by the EntireX Workbench.
Service Names	A list of Integration Server services, separated by blanks. Each service corresponds to a subprogram name. For the subprogram the service at the same position in the list is called. The lists must have the same number of names.
Minimum Listener Threads	The number of worker threads for this listener that should be started when the listener is enabled. If set to 0, worker threads are started on demand. Default: 1.
Maximum Listener Threads	The maximum number of listener threads to start. Default: 1.
Entry Handler Service	The name of an Integration Server service which will be called in addition before the listener calls a service listed in **Service Names**. The specification `pub.wmentirex.listener:entryExitHandlerSpecification` describes the interface of the handler service.
Exit Handler Service	The name of an Integration Serverservice which will be called in addition after the listener has called a service listed in **Service Names**. The specification `pub.wmentirex.listener:entryExitHandlerSpecification`describes the interface of the handler service.
Execute Service with Client Credentials	If enabled, the RPC client has to provide RPC user ID and RPC password. These credentials are then used to execute the Integration Server service. Note that currently the RPC user ID and password are limited to 8 characters. Default: `false`.
Null Value Suppression	See *Configuring Null Value Suppression*.

6   Save the listener.

Testing Listeners

You can test listeners to ensure that you have configured them correctly.

▶ **To test a listener**

1   Configure a listener using the IS Administration Console. For instructions on how to configure a listener, see *Configuring New Listeners*.

2   For Reliable RPC Listener Connections only (asynchronous calls): Configure a listener notification using the Designer. See *webMethods Service Development Help* for more information.

3   For Reliable RPC Listener Connections only (asynchronous calls): Enable the listener notification using the IS Administration Console. For instructions on how to enable a listener notification, see *Managing EntireX Adapter Notifications*.

4   Enable the listener using the IS Administration Console.

   ⚠   **Important:** The broker configured in the connection associated with the listener must be running while the listener is enabled.

5   Use an RPC client to call the RPC server specified in the connection associated with the listener.

   As a result, the listener receives the call, uses the parameters to call the flow service or the Java service, and returns the results to the RPC client.

   📄   **Notes:**

   1. On error a listener may go into the disabled state.

   2. If a listener is disabled, it will not start at next start of Integration Server.

Viewing Listener Parameters

You can view a listener's parameters using the IS Administration Console or the Designer. You can also view the notification order of a listener.

▶ **To view a listener's parameters using the IS Administration Console**

1   In the **Adapters** menu in the navigation area of the IS Administration Console, click **EntireX Adapter**.

2   In the **EntireX Adapter** menu, click **Listeners**.

3   On the listeners page, click the **View** icon for the listener that you want to see.

As a result, the View Listener screen displays the parameters for the listener. For descriptions of the listener parameters, see *Configuring New Listeners*.

4   Click **Return to EntireX Adapter Listeners** to return to the main listeners screen.

> **Note:** Alternatively, you can use the Designer. See *webMethods Service Development Help* for more information.

Before Configuring or Managing Notifications

▶ **To prepare to configure or manage EntireX Adapter notifications**

1   Start your Integration Server and the IS Administration Console, if they are not already running.

2   Make sure you have webMethods administrator privileges so that you can access the EntireX Adapter's administrative screens. See *Administering webMethods Integration Server* for information on setting user privileges.

3   Using the IS Administration Console, make sure the WmEntireX package is enabled. See *Enabling and Disabling Packages* for instructions.

4   Using the IS Administration Console, configure an adapter connection to use with the notification. See *Configuring Adapter Connections*.

5   Start the Designer if it is not already running.

6   Make sure you are viewing the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help.

7   Using the Designer, create a user-defined package to contain the notification, if you have not already done so.

> ⚠ **Important:** When you configure notifications, you should always define them in user-defined packages rather than in the WmEntireX package. For more information on managing packages for the adapter, see *Package Management*.

> ⚠ **Important:** You must enable a notification before you can use it. See *Managing EntireX Adapter Notifications* for instructions.

Configuring Adapter Notifications

When you configure EntireX Adapter notifications, you specify information that the Integration Server uses to read data from an EntireX broker and call a specified flow service or Java service. For more information on notifications, see *Listeners and Adapter Notifications* (separate sections). Be sure to review the section *Before Configuring or Managing Notifications* before you configure EntireX Adapter notifications.

Adapter notifications are only used for Reliable RPC Listener Connections.

▶ **To configure an EntireX Adapter notification**

1   From the Designer **File** menu, choose **New...**

2   Select **Adapter Notification** from the list of elements and click **Next**.

3   Select **EntireX Adapter** as the adapter type and click **Next**.

4   Select the appropriate **Adapter Connection Name** and click **Next**.

5   Select the **Asynchronous Listener Notification** template and click **Next**.

6   Select an **Adapter Notification Listener** and click **Next**.

7   Type a unique name for the notification and select the appropriate folder. Click **Next**.

8   Select a service to be called when an RPC call arrives at the listener. Click **Next**.

9   The names of the request and reply documents are displayed. Click **Finish**.

    As a result, the editor for the adapter notification appears.

    > **Note:** You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Notification Template**, as necessary.

10 Select the **Listener Notification** tab and choose the EntireX Subprogram Name. You may select **Map Data Types to String** to map all IDL parameters to `java.lang.String`. Otherwise, the IDL data types are mapped to Java primitive data types.

    As a result, the tables of output parameters and input parameters display the names, the IDL data types and the Java data types in the pipeline.

    > **Note:** For information on using the **Permissions** tab to assign an access control list (ACL) to an element, see

11 From the **File** menu, choose **Save** (or **Save All**).

📄 **Note:** You must schedule and enable the notification using the IS Administration Console before you can use it. See *Managing EntireX Adapter Notifications* for details.

Managing EntireX Adapter Notifications

You must enable an EntireX Adapter notification before you can use it to monitor EntireX brokers for messages. Use the IS Administration Console along with the following procedure to do so.

📄 **Note:** You must have webMethods administrator privileges to access the EntireX Adapter's administrative screens. See the *Administering webMethods Integration Server* for information on setting user privileges.

▶ **To enable (or disable) an EntireX Adapter notification**

1 Start the IS Administration Console.

2 From the **Adapters** menu in the navigation area of the IS Administration Console, choose **EntireX Adapter**.

3 From the navigation area, select **Listener Notifications**.

4 In the **Listener Notifications** table, you can enable and disable the notification.

⚠️ **Important:** You cannot enable a notification if the adapter connection that the notification is using is disabled.

As a result, the name of the package is displayed.

Viewing Notifications

Use the Designer to view a notification. Make sure you are using the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help.

▶ **To view a notification**

1 In the Designer **Package Navigator** view, expand the package and folder that contain the notification you want to view.

2 Select the notification you want to view.

The Designer displays the notification in the **Adapter Notification Editor**.

Editing Notifications

Use the Designer to edit notifications. Make sure you are using the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help. You can change which connection an adapter notification uses.

▶ **To edit a notification**

1   In the Designer **Package Navigator** view, expand the package and folder that contain the notification you want to view.

2   Select the notification you want to edit.

    The Designer displays the notification in the **Adapter Notification Editor**.

3   Modify the values for notification's parameters as needed. For detailed descriptions of the notification's parameters, see the section on configuring a notification.

Deleting Notifications

Use the Designer to delete adapter notifications. Make sure you are viewing the Designer in the **Service Development** perspective, as described in *Switching Perspectives* in the Designer online help.

▶ **To delete a notification**

1   In the Designer **Package Navigator** view, expand the package and folder that contain the notification you want to delete.

2   Right-click the notification and then click **Delete**.

7 Settings and Information

This chapter describes information and tracing screens in the IS Administration Console.

Adapter Settings

This screen shows the current level of EntireX Tracing. You can change this level with the **Change Settings** link. The trace levels are "0" (no tracing), "1" (trace calls), "2" (trace calls and buffers), "3" (full tracing).

The trace file is written to the standard log directory of the Integration Server or the directory defined in **Change Settings** and has the name *wmentirexYYYYMMDD.log*.

You can change the location (directory) for the trace files with the **Change Settings** link. This has immediate effect.

Further Setting Options for Tracing:

■ The adapter uses the logging facility of the Integration Server. The facility code is "0800" for the EntireX Adapter. The logging level specifies the amount of data logged. These log entries are written to the server log file.

Connections Information

This screen displays the following sections about connections and listeners.

■ **Broker Information**
The status of all brokers used in the connections. A green dot shows that a connection to the broker is successful. As status, the version and the operating system platform is displayed. A red dot shows that a connection to the broker is not possible. As status, the error message is displayed.

■ **Connections for Services**
The status of the external servers such as RPC Server, IMS Connect and CICS ECI used by the connections for adapter services. A green dot shows that connection to the RPC server is successful and the status information from the server is displayed. A red dot shows that a connection to the RPC server or to the Broker is not possible. As status, the error message is displayed. The current number of connection instances is shown in column **Count**.

■ **Connections for Listeners**
For each Listener the corresponding name of the connection, Broker ID, and server address is displayed.

Built-in Services for Connections Information

The information on connections and listeners shown on the IS Administration Console's page is available with built-in services. The following services are available in the WmEntireX package:

- pub.wmentirex.connectioninfo:brokerInfo
- pub.wmentirex.connectioninfo:brokerList
- pub.wmentirex.connectioninfo:listenerList
- pub.wmentirex.connectioninfo:serviceInfo
- pub.wmentirex.connectioninfo:serviceList

pub.wmentirex.connectioninfo:brokerInfo

The service retrieves the status of the broker given in the broker ID. Use `pub.wmentirex.connectioninfo:brokerList` before to get the list of broker IDs used in all connections.

Input Parameters

`brokerID String`	The ID of the broker to be `pinged`.
`refreshData String`	The number of milliseconds to refresh the data. If `refreshData` is null or not a number, the default of 10 seconds is used.

Output Parameters

`errorFlag String`	"true" if an error occurred, "false" if the call succeeded.
`message String`	The result of the ping call to the broker.

pub.wmentirex.connectioninfo:brokerList

The service retrieves the list of all broker IDs configured in the connections.

Input Parameters

`refreshData String`	The number of milliseconds to refresh the data. If `refreshData` is null or not a number, the default of 10 seconds is used.

Output Parameters

`brokerIDs String[]`	The array of broker IDs.

pub.wmentirex.connectioninfo:listenerList

The service retrieves name, broker ID and server address for each listener.

Input Parameters

`refreshData String`   The number of milliseconds to refresh the data. If `refreshData` is null or not a number, the default of 10 seconds is used.

Output Parameters

`listeners IData[]`   The array of IData objects containing three strings *name*, *brokerID*, *serverAddress* for each listener.

pub.wmentirex.connectioninfo:serviceInfo

The service retrieves the result of the ping call for the connection.

Input Parameters

`connection String`   The name of the connection to `ping`, e.g. `"folder:name"`.

`refreshData String`   The number of milliseconds to refresh the data. If `refreshData` is null or not a number, the default of 10 seconds is used.

Output Parameters

`errorFlag String`   "true" if an error occurred, "false" if the call succeeded.

`message String`   The result of the ping call to the connection.

pub.wmentirex.connectioninfo:serviceList

The service retrieves the connection names. One of these names can be used as input for `pub.wmentirex.connectioninfo:serviceInfo`.

Input Parameters

`refreshData String`   The number of milliseconds to refresh the data. If `refreshData` is null or not a number, the default of 10 seconds is used.

Output Parameters

`connections String[]`   The array of connection names in the format `folder:connection`.

Support Information

Connections, adapter services and adapter listeners contain metadata which is generated during deployment of connections and services. Use the service `pub.wmentirex.supportinfo:createInfo` to extract this metadata for support purposes and error diagnosis.

pub.wmentirex.supportinfo:createInfo

The service extracts the generated metadata from all connections and adapter services in a package and writes this to a file in the installation folder of the Integration Server.

Input Parameters

`packageName String` The name of a package. The generated metadata for all connections, adapter services and adapter listeners in this package is written to a file.

Output Parameters

`result String` File <*filename*> created on host <*hostname*>

Services Information

This screen shows statistics about the adapter services of the EntireX Adapter. Only services that have been executed at least once during the runtime of the Integration Server are shown.

The following information is shown for each service:

- name of the service
- corresponding program name
- number of successful calls
- number of erroneous calls
- average processing time in milliseconds (only successful calls)
- time of last access
- processing time in milliseconds for the last call
- status of last call (detailed error message if the last call failed)

> **Note:** All these items except the service name can be reset with the link "Reset Statistics".

License Information

This screen shows information on the license used to enable the usage of Direct RPC, or connections using IMS Connect or CICS ECI connectivity.

- License File Location: *name-of-licence-file*. If the file is not found, the location where the file is expected is shown.
- Valid until: values are a date specification or "Unlimited"
- License for connections to IMS Connect: "Yes" or "No"
- License for connections to CICS ECI: "Yes" or "No"
- License for Direct RPC: "Yes" or "No"
- Error message (only if the license is not granted)

Notes:

1. The folder *<webMethods installation>/IntegrationServer/config* is the default folder for the license file.

2. You can change the location of the license file (file name with folder) with the **Change Settings** link.

3. The folder *<webMethods installation>/IntegrationServer/packages/WmEntireX* is not a safe folder for the license file, because the Integration Server cleans this folder on update.

Application Monitoring

This functionality is only available with the product webMethods Optimize for Infrastructure. For details, see the *Optimize documentation Guide for Enterprise Transaction Systems*.

8 Direct RPC

Direct RPC is a component that enables RPC clients and RPC servers to connect directly to the EntireX Adapter without the EntireX Broker.

Configuring Direct RPC

To enable Direct RPC, choose **Direct RPC Administration** in the EntireX Adapter's administration menu and follow the wizard. The table below gives an overview of parameters to be specified:

Name	Values	Default	Action	Description
Status	"Stopped", "Running"	"Stopped"	Start or stop.	Current status of the Direct RPC component.
Logging Level	"Fatal", "Error", "Warning", "Info", "Debug", "Trace"	"Fatal"	Change logging level.	Logging level. See *Notes*.
TCP Port Number	Port number	"1971"	Change port number.	TCP listening port for Direct RPC.
SSL Port Number	Port number	-	Change port number.	SSL (or TLS) listening port for Direct RPC.
Keystore Alias (for SSL)	The alias name of a keystore defined in the Integration Server.	-	Change alias name.	Only needed for SSL. Defined in the IS web admin "Security > Keystore".
Auto Start	Enabled, Disabled	"Disabled"	Enable or disable.	Automatic start of Direct RPC during startup of the Adapter.

Notes:

▪ Possible logging levels:

Fatal	Fatal errors, Direct RPC cannot work correctly.
Error	Severe errors, Direct RPC continues to work correctly.
Warning	Errors with low severity.
Info	To follow the flow of requests between RPC clients, servers and Direct RPC.
Debug	All communication buffers will be written to the log file.
Trace	Internal use only (Software AG support).

▪ *wmentirex.directrpcYYYYMMDD.log* is located in folder *logs* in the Integration Server installation folder. To change the default location, see *Adapter Settings*.

▪ It is not recommended to use "Info" , "Debug" or "Trace" in a production.

Encoding for RPC Clients and Servers (Default and Available Codepages)

The default encoding of an RPC client or server is the same as the default encoding of the platform used. The table below gives an overview on default and available codepages:

Platform	Default Codepage	Available Codepages (single byte only)
Windows/UNIX/Linux	ISO 8859-1	ASCII codepages
IBM mainframe (z/OS, z/VSE)	CP037	EBCDIC codepages
Fujitsu mainframe (BS2000/OSD)	CP273	EBCDIC codepages

To specify the encoding of an RPC client or server, see *webMethods EntireX* documentation.

Monitoring

The following table gives an overview on monitored objects:

Item	Description
Servers	Number of registered servers.
RPC Requests	Number of RPC requests.
Socket Connections	Number of plain socket connections (TCP) from RPC clients or servers.
Secure Socket Connections	Number of secure socket connections (SSL or TLS) from RPC clients or servers.
Direct Connections	Number of direct connections from Adapter services or Adapter Listeners.
Calls	Number of calls (including calls such as `logon`).
Conversations	Number of conversational calls.

Notes:

1. For each item the following information is provided: Current (number of items currently active), maximum (high watermark) and total (number of activations so far).

2. If Direct RPC connections are enabled, the value of `Direct Connections` can be greater than zero even if Direct RPC is stopped.

Services

The following information is displayed for each service:

Name	The name of the service
Type	"External RPC Server" or "Direct RPC Server".
Instances	Number of registered replicates.
Wait for Server	Number of client requests that are waiting for a free server to be processed. The percentage (number of calls waiting compared to the total number of calls) is helpful in deciding if further server replicates are necessary.
Action	Shutdown (only for external RPC servers).

> **Note:** As a prerequisite, at least one server must be registered or listener enabled.

Servers

The following information is displayed for each service:

Service Name	The name of the service
Type	"Listener" or "RPC Server".
Name	Name of the Direct Listener or the external RPC server.
Host	Host where the external RPC Server is running.
Version	Version of the external RPC server.
Action	Shutdown (only for external RPC Servers).

> **Note:** `Host`, `Name` and `Version` can only be displayed if the RPC server provides this information.

Built-in Services for Direct RPC

The functionality of Direct RPC as shown on the IS Administration Console's page is available with built-in services in the WmEntireX package. They correspond to the parameters on the IS Administration Console's page. When a service is executed, the signature of the input and output parameters is shown. The following services are available:

- `pub.wmentirex.directrpc.admin:getAutoStart`

- `pub.wmentirex.directrpc.admin:getIbmMainframeDefaultCharset`

- `pub.wmentirex.directrpc.admin:getKeyStoreAlias`

- `pub.wmentirex.directrpc.admin:getLoggingLevel`

- `pub.wmentirex.directrpc.admin:getSSLPortNumber`

- `pub.wmentirex.directrpc.admin:getSiemensMainframeDefaultCharset`

- `pub.wmentirex.directrpc.admin:getStatus`

- `pub.wmentirex.directrpc.admin:getTCPPortNumber`

- `pub.wmentirex.directrpc.admin:getWindowsUnixDefaultCharset`

- `pub.wmentirex.directrpc.admin:setAutoStart`

- `pub.wmentirex.directrpc.admin:setIbmMainframeDefaultCharset`

- `pub.wmentirex.directrpc.admin:setKeyStoreAlias`

- `pub.wmentirex.directrpc.admin:setLoggingLevel`

- `pub.wmentirex.directrpc.admin:setSSLPortNumber`

- `pub.wmentirex.directrpc.admin:setSiemensMainframeDefaultCharset`

- `pub.wmentirex.directrpc.admin:setTCPPortNumber`

- `pub.wmentirex.directrpc.admin:setWindowsUnixDefaultCharset`

- `pub.wmentirex.directrpc.admin:start`

- `pub.wmentirex.directrpc.admin:stop`

In addition, there are 4 services for advanced configuration of the SSL transport:

- `pub.wmentirex.directrpc.admin:getSSLClientAuthentication`

- `pub.wmentirex.directrpc.admin:getTrustStoreAlias`

- `pub.wmentirex.directrpc.admin:setSSLClientAuthentication`

- `pub.wmentirex.directrpc.admin:setTrustStoreAlias`

> **Note:** Use the `setSSLClientAuthentication` service to switch on SSL client authentication. Client authentication requires that a truststore alias has been set with `pub.wmentirex.directrpc.admin:setTrustStoreAlias`. The truststore alias is defined in the IS web admin **Security** > **Keystore**. Note that the names of the keystore alias and the truststore alias must be different.

Limitations of Direct RPC

- **An external RPC client cannot connect to an external RPC server.**
  It is only possible that an adapter service connects to an external RPC server, or an external RPC client connects to an adapter listener.

- Exceptions to the rule above are some components of the EntireX Workbench: the IDL Tester, the IDL Extractor for COBOL, the IDL Extractor for Natural and the Natural Wrapper. This requires the latest version of the EntireX Workbench, and only single-byte encodings can be used.

- Reliable RPC is not supported.

- Command and Information Services (CIS) are not supported. Using this API (directly or through the System Management Hub) can result in errors, or the data returned is incomplete or missing.

- An adapter listener and an external RPC server cannot be registered for the same service.

- Replicates of external RPC servers must use the same encoding.

- EntireX Security is not supported.

- ACI applications are not supported. For Natural RPC clients or servers, Broker ACI is only supported for versions higher than 1. See the documentation on the `ACIVERS` parameter in *Parameter Reference* in the latest *Natural for Mainframe* documentation under *https://empower.softwareag.com/Products/Documentation/default.asp*.

9   Security Support

This chapter covers the following topics:

Security Support for Adapter Services

The EntireX Adapter uses adapter services to send synchronous or asynchronous requests to various backend systems. Access to these systems can be secured in the following ways:

■ **Provide security credentials in the adapter connection for security-protected backend systems**
  If access to a backend system is security protected, security credentials (user ID and password) have to be provided in the corresponding adapter connection. This can be done when the connection is configured or edited, see *EntireX Adapter Connections*. This security support is optional and not available for all connection types, see the overview of available connection parameters in *Editing Adapter Connections*.

  The credentials can also be provided dynamically when calling the adapter service, see *Using Adapter Services*.

■ **Set access control permissions for the adapter service**
  Using the Designer you can set access control permissions to restrict the right to execute an adapter service to a particular user group of the Integration Server:

  ▶ **To set access control permissions using the Designer**

  1   Choose **Service Development** > **Package Navigator** > **Permissions**.
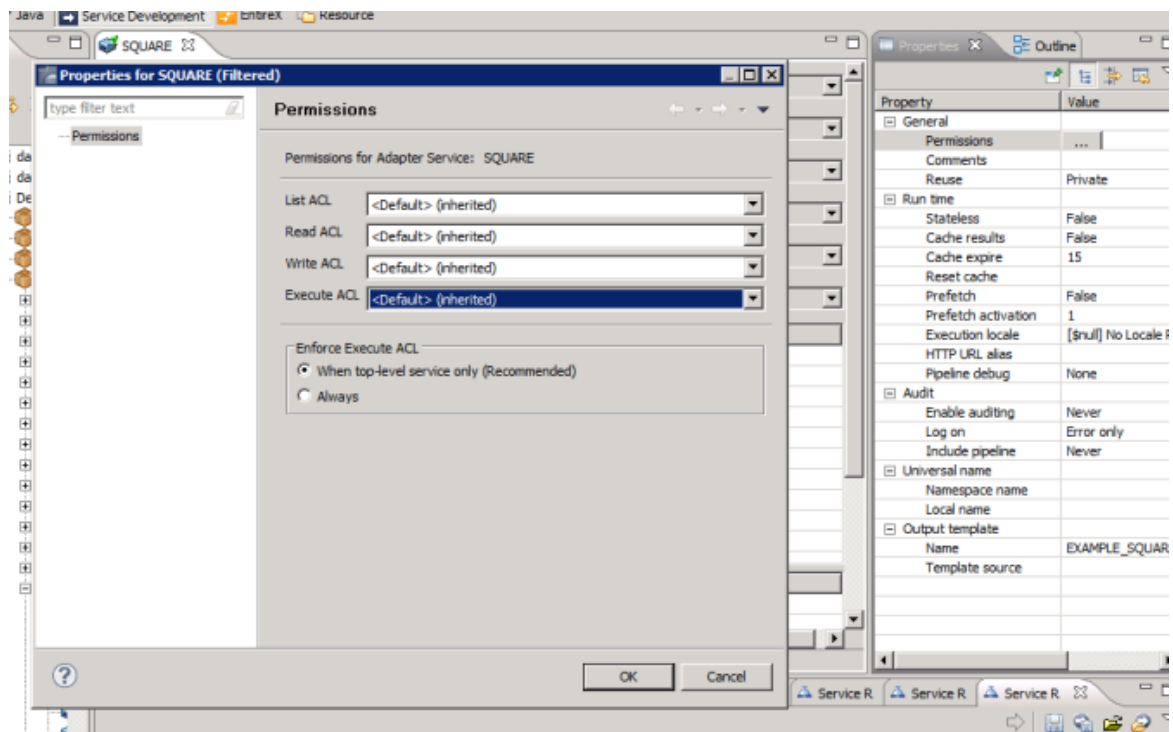
2   Set **Enforce Execute ACL** to **When top-level service only** to select the group which is spe-
     cified on the top-level service. This way, access control is already checked when executing
     the top-level service.

     Or:

     Set **Enforce Execute ACL** to **Always** and select a user group in **Execute ACL**.

Security Support for Adapter Listeners

Adapter listeners receive synchronous or asynchronous requests from external RPC clients. If the
RPC request is sent to the EntireX Adapter using an EntireX Broker which is configured to use
security then the credentials of the RPC clients are checked to see if the client is permitted to call
the adapter listener.

By enabling the listener property **Execute Service with Client Credentials**, an adapter listener
can be configured to use the credentials (user ID and password) provided by the RPC client to
execute the Integration Server service:

Listener Properties

EntireX Subprogram Names	SQUARE
Service Names	Default:squareFlow
Minimum Listener Threads	
Maximum Listener Threads	
Entry Handler Service	
Exit Handler Service	
Execute Service with Client Credentials	true

Then the RPC client has to provide credentials (RPC user ID and RPC password) which are used to execute the Integration Server service. Note that currently the RPC user ID and password are limited to 8 characters.

Support for SSL/TLS

Communication between the Adapter and the backend systems is done using TCP/IP or, for secure communication, SSL/TLS.

The configuration of SSL depends on the connection type:

Connection type	Description
Connection types using the EntireX Broker	Use the URL-style syntax by which Broker IDs are defined to select SSL as transport method and to define the SSL parameters. For example: `ssl://host:1958?definition_of_sslparameters`. The syntax for defining SSL parameters is described in the notes below.
IMS Connect and CICS ECI	Specify the SSL port in the **Port** property and the SSL parameters in the **SSL Parameters** property as described in *Editing a Connection*. The syntax for defining SSL parameters is described in the notes below.
Direct RPC	Specify the SSL port and SSL parameters in the configuration wizard for **Direct RPC**.

📄 **Notes:**

1. SSL parameters are defined in URL style as name value pairs separated by an ampersand ('&').

2. You always have to specify `trust_store=file`, where `file` is the path name of a Java keystore file which contains the list of trusted certificate authorities. Specify `verify_server=no`, if you do not want to check that the certificate of the SSL server is issued for the specified hostname. Example: `ssl://host:1958?trust_store=/temp/ExxCACert.jks&verify_server=no`

3. If the SSL server requests a client certificate, specify `key_store=file` and `key_passwd=pwd`. Again, `file` is the path name of a Java keystore file which contains the private key and `pwd` is the password which is needed to access the private key entry in the keystore. Example: `ssl://host:1958?trust_store=/temp/ExxCACert.jks&key_store=/temp/ExxJavaAppCert.jks&key_passwd=ExxJavaAppCert`

10 Message Class 0800 - webMethods EntireX Adapter

Messages are logged with message class 0800. In the Administrator, section **Settings > Logging** you find the logging settings in the tree for Adapters. See the *webMethods Integration Server Administrator's Guide* for changing the logging settings.

The messages have the format:

0800*nnnn*

where   0800   is the message class, and

        nnnn   is the message number in the range 0000 - 9999

08000212	**WmEntireX: exception during register: \<exception detail>**
Explanation	An error occurred during REGISTER call. This can happen when an adapter listener starts.
Action	Use exception detail to fix. This is typically a configuration issue.

08000213	**WmEntireX: exception during receive: \<exception detail>**
Explanation	An error occurred during RECEIVE call in an adapter listener.
Action	Use exception detail to fix.

08000219	**WmEntireX: connection.receive: unknown server command \<actual command>**
Explanation	\<actual command> was send to the adapter listener. This is unknown and cannot be handled.
Action	Change command. The only known command is PING.

08000220   **WmEntireX: connection.receive: conversational call <actual packet type> rejected**

Explanation   The adapter listener does not support conversational RPC.

Action   Use non-conversational calls.

08000221   **WmEntireX: connection.receive: illegal packet type <actual packet type>**

Explanation   The adapter listener received an illegal packet type.

Action   Change your RPC client to send valid packet types.

08000229   **WmEntireX: unsupported RPC protocol version: <actual protocol version>**

Explanation   The RPC client is sending an old protocol version that is not supported by the adapter
                     listener.

Action   Update your RPC client.

08000246   **WmEntireX: Some programs not found in COBOL metadata, expected library:
                     <library name>, actual: <actual library name>, programs not found: <expected
                     program names>**

Explanation   The COBOL metadata of the connection is not consistent.

Action   Re-generate your connection and the adapter services.

08000300   **WmEntireX: exception during processing notification result: <exception details>**

Explanation   The adapter listener trapped into an exception while processing the result of the
                     notification execution.

Action   Use exception details to fix.

08000310   **WmEntireX: listener.processNotificationResults error: <error details>**

Explanation   The Integration Server runtime reported an error during notification.

Action   Use error details to fix.

08000333   **WmEntireX: exception during syncListener.runNotification: <exception details>**

Explanation   The adapter listener trapped into an exception while processing a notification event.

Action   Use exception details to fix.

08000401	**WmEntireX: no program info for \<program name\> found in metadata info**
Explanation	The metadata for this connection does not contain necessary info for this program.
Action	Re-create this connection.

08000402	**WmEntireX: illegal program info for \<program name\> found in metadata info, length=\<number of info blocks for the program\>**
Explanation	The metadata for this connection does not contain consistent info for this program.
Action	Re-create this connection.

08000403	**WmEntireX: illegal program info for \<program name\> found in metadata info, length(headerInfos)=\<number of info blocks for the program header\>**
Explanation	The metadata for this connection does not contain consistent info for this program.
Action	Re-create this connection.

08000404	**WmEntireX: exception while processing metadata info: \<exception details\>**
Explanation	The metadata for this connection does not contain consistent info for this program.
Action	Re-create this connection.

08000420	**WmEntireX: exception while reading metadata info: \<exception details\>**
Explanation	The metadata for this connection is not readable.
Action	Re-create this connection. Use exception details to fix.

08000421	**WmEntireX: metadata info cannot be empty**
Explanation	The metadata for this connection is empty.
Action	Re-create this connection.

08000422	**WmEntireX: no library name found in metadata info**
Explanation	The metadata for this connection does not contain an IDL library name.
Action	Re-create this connection.

08000423	**WmEntireX: no programs found in metadata info**
Explanation	The metadata for this connection does not contain an IDL program name.
Action	Re-create this connection.

08000424	**WmEntireX: no information found for program <program name>**
Explanation	The metadata for this connection does not contain the signature information for `<program name>`.
Action	Re-create this connection.

08000441	**Illegal value of type <actual type>, <expected type> expected**
Explanation	The actual parameter type provided by the Integration Server differs from the expected parameter type. The types are incompatible.
Action	Supply parameters of expected type.

08000442	**Illegal value <actual value> for type <actual IDL data type>**
Explanation	The actual parameter value provided by the Integration Server does not match the expected parameter type.
Action	Use exception details to fix this.

08000443	**Illegal type <actual type> for <parameter name>**
Explanation	The parameter has an unknown IDL type.
Action	Contact Software AG support.

11 Message Class 2011 - Connections to IMS Connect

This section describes messages that can occur with connections from the webMethods EntireX Adapter to IMS Connect. The messages have the format:

2011*nnnn*

where   2011   is the message class, and

      nnnn   is the message number in the range 0000 - 9999

20110001	**Process request and reply: illegal action <number indicating action>**
Explanation	During request/reply processing the next action is illegal in the current state.
Action	Contact Software AG support.

20110009	**<Exception with detailed message >**
Explanation	An exception occurred while formatting the message. In most cases, the reason is an inconsistency in metadata.
Action	See detailed message or contact Software AG support.

20110010	**Compression=off not possible**
Explanation	An RPC with compression "0" or "off" is not supported.
Action	Change compression to 1 or 2.

20110011	**Type of marshalling not allowed.** **or** **Unsupported call type <number of call type>**
Explanation	There are two reasons for this error code:

1. For the `marshalling` property the values "cobol" and "natural" are allowed. Reason: The value of the property may be invalid. Or: The RPC from the client is in the wrong marshalling.

2. The RPC call type is wrong.

Action	Check configuration and client.

20110012	**<Exception with detailed message>**
Explanation	This exception occurs in message processing.
Action	See detailed message and contact Software AG support.

20110013	**Preparing message failed, see log for details** **or** **Message received from server does not fit buffer**
Explanation	There are two reasons for this error code. First, marshalling of message failed before sending to IMS Connect. Second, the received message has not the expected layout and the values cannot be unmarshalled from it. Marshalling of message failed before sending to IMS Connect.
Action	Check metadata (IDL, SVM, adapter connections) for consistency or contact Software AG support.

20110014	**Transformation error on reply: <Exception with detailed message>**
Explanation	The metadata is inconsistent or this data type is not supported.
Action	See detailed message or contact Software AG support.

20110015	**Transformation error on request: <Exception with detailed message>**
Explanation	This data type is not supported.
Action	See detailed message or contact Software AG support.

20110022	Unsupported command <command>. or Exception: <Exception with detailed message>
Explanation	There are two reasons for this error code.
	1. An unknown internal command is used.
	2. An exception occurred during action processing. The log files contain a stack trace.
Action	See detailed message and log files or contact Software AG support.

20110023	Preparing message failed, see log for details
Explanation	Marshalling of message failed before sending to IMS Connect.
Action	Check metadata (IDL, SVM, adapter connections) for consistency or contact Software AG support. Details can be found in `wmentirex.log` with trace level 2.

20110301	Receive only not supported
Explanation	This type of call is not supported.
Action	Modify your client.

20110302	Send only not supported
Explanation	This type of call is not supported.
Action	Modify your client.

20110303	<detailed error message from IMS Connect>
Explanation	IMS Connect reported an error, or the configuration is wrong.
	If IMS Connect returns code 16 and reason code 64, one cause is that the IMS exit is not installed or is not running. The default exit name is HWSSMPL0 (old) or HWSSMPL1 (new). Other configurations may also result in this return code and reason code.
	If the IMS Connect connection in the EntireX Adapter uses a port which is not related to IMS but to some other application, a call may result in the following message: `Unable to invoke adapter service <service name> [ADA.800.500] WmEntireX: exception while executing service: 2011 0303 java.io.EOFException`
	If the IMS Connect RPC server uses a port which is not related to IMS but to some other application, a call may result in the following message: `exception while executing service: 2011 0303 java.io.EOFException`
Action	Use the error code from IMS Connect to correct this or correct the configuration.

20110304	**Conversations not supported**
Explanation	IMS Connect does not support conversational RPC calls.
Action	Use non-conversational calls.

12 Message Class 2012 - Connections to CICS ECI

This message class contains messages that can occur with connections to CICS ECI. The messages have the format:

2012*nnnn*

where   2012   is the message class, and

      nnnn   is the message number in the range 0000 - 9999

20120001	**Process request and reply: illegal action <number indicating action>.**
Explanation	During request/reply processing the next action is illegal in the current state.
Action	Contact Software AG support.

20120009	**<Exception with detailed message >**
Explanation	An exception occurred while formatting the message. In most cases, the reason is an inconsistency in metadata.
Action	See detailed message or contact Software AG support.

20120010	**Compression=off not possible.**
Explanation	An RPC with compression "0" ("off") is not supported.
Action	Change compression to 1 or 2.

20120011	**Type of marshalling not allowed.** **or** **Unsupported call type <number of call type>**
Explanation	There are two reasons for this error code:

1. For the `marshalling` property the values "cobol" and "natural" are allowed. Reason: The value of the property may be invalid. Or: The RPC from the client is in the wrong marshalling.

2. The RPC call type is wrong.

Action	Check configuration and client.

20120012	**<Exception with detailed message>**
Explanation	This exception occurs in message processing.
Action	See detailed message and contact Software AG support.

20120013	**Preparing message failed, see log for details.** **or** **Message received from server does not fit buffer**
Explanation	There are two reasons for this error code. First, marshalling of message failed before sending to CICS ECI. Second, the received message does not have the expected layout and the values cannot be unmarshalled from it.
Action	Check metadata (IDL, SVM, adapter connections) for consistency or contact Software AG support.

20120014	**Transformation error on reply: <Exception with detailed message>**
Explanation	The metadata is inconsistent or this data type is not supported.
Action	See detailed message or contact Software AG support.

20120015	**Transformation error on request: <Exception with detailed message>.**
Explanation	This data type is not supported.
Action	See detailed message or Contact Software AG support.

20120022	**Unsupported command <command>.** **or** **Exception: <Exception with detailed message>**
Explanation	There are two reasons for this error code. First, an unknown internal command is used. Second, an exception occurred during action processing. The log files contain a stack trace.
Action	See detailed message and log files or contact Software AG support.

20120023	**Preparing message failed, see log for details**
Explanation	Marshalling of message failed before sending to CICS ECI.
Action	Check metadata (IDL, SVM, adapter connections) for consistency or contact Software AG support. Details can be found in `wmentirex.log` with trace level 2.

20120351	**Receive only not supported**
Explanation	This type of call is not supported.
Action	Modify your client.

20120352	**Send only not supported**
Explanation	This type of call is not supported.
Action	Modify your client.

20120353	**<detailed error message from CICS ECI>**
Explanation	CICS ECI reported an error, or the configuration is wrong. This error number is also issued if the connection to CICS ECI is lost.
Action	Use the error code from CICS to correct this. The CICS error codes are described in the documentation of CICS from IBM.

Possible error codes are (others may occur):

- **AEIO: DUPKEY**
  possible reasons: Callee not found - the called program does not exist or is not started.
- **ASRA**
  possible reasons: Division by zero.
- **4038: Index overwrite**
- **TCII (CICS Abend)**

A call to CICS ECI can result in an EOF exception because of

- Connection to the wrong port. There is some service listening on the port, but not CICS ECI.
- CICS ECI is not correctly configured.

20120354   **Conversations not supported**

Explanation   CICS ECI does not support conversational RPC calls.

Action   Use non-conversational calls.

13 Message Class 2023 - Connections to ApplinX

This message class contains messages that can occur with ApplinX Connections in the EntireX Adapter. The messages have the format:

2023*nnnn*

where 2023 is the message class, and

        nnnn is the message number in the range 0000 - 9999

20230001	**Process request and reply: illegal action <number indicating action>.**
Explanation	During request/reply processing the next action is illegal in the current state.
Action	Contact Software AG support.

20230009	**<Exception with detailed message >**
Explanation	An exception occurred while formatting the message. In most cases, the reason is an inconsistency in metadata.
Action	See detailed message or contact Software AG support.

20230010	**Compression=off not possible**
Explanation	An RPC with compression "0" ("off") is not supported.
Action	Change compression to 1 or 2.

20230011	**Type of marshalling not allowed**
Explanation	For the `marshalling` property the values `cobol` and `natural` are allowed. First cause: The value of the property may be invalid. Second cause: The RPC from the client is in the wrong marshalling.
Action	Check configuration and client.

20230012	**<Exception with detailed message>**
Explanation	This exception occurs in message processing.
Action	See detailed message and contact Software AG support.

20230013	**Preparing message failed, see log for details** **or** **Message received from server does not fit buffer**
Explanation	There are two possible reasons for this error code: 1. Marshalling of message failed before sending to CICS ECI. 2. The received message has not the expected layout and the values cannot be unmarshalled from it.
Action	Check metadata (IDL, SVM, adapter connections) for consistency or contact Software AG support.

20230015	**Transformation error on request: <Exception with detailed message>.**
Explanation	This data type is not supported.
Action	See detailed message or contact Software AG support.

20230022	**Unsupported command <command>.** **or** **Exception: <Exception with detailed message>**
Explanation	There are two possible reasons for this error code: 1. An unknown internal command is used. 2. An exception occurred during action processing. The log files contain a stack trace.
Action	See detailed message and log files or contact Software AG support.

14 Message Class 0400 - Direct RPC

This message class contains messages that can occur with Direct RPC. The messages have the format:

0400*nnnn*

where 0400 is the message class, and

nnnn is the message number in the range 0000 - 9999

04000001	**Exception during processing call: <exception text>**
Explanation	An unexpected error occurred during processing of a call from an RPC client or server.
Action	Check the exception text for details.

4000002	**Transport session not active**
Explanation	A call happens on an adapter connection although this connection instance is disabled.
Action	Contact Software AG support.

04000003	**Invalid usage of Direct RPC (<class/server/service>)**
Explanation	An RPC client sends a request to an RPC server using Direct RPC. This is not supported.
Action	Change your application setup so that

1. an RPC client sends a request to a Direct RPC listener, or

2. a Direct RPC adapter service sends a request to an RPC server.

04000004	**Invalid register: replicate from different transport (<class/server/service>)**
Explanation	Both a listener and an RPC server try to register using the same `class/server/service` name. This is not supported.
Action	Change your application so that only listeners or RPC servers register for the same `class/server/service` name.

04000005	**Send with wait not possible for servers (<class/server/service>)**
Explanation	An RPC server issues a synchronous send command, which is not supported.
Action	Check your configuration: for example, an ACI server might be using Direct RPC.

04000006	**Send with no wait not possible for clients (<class/server/service>)**
Explanation	An RPC client issues an asynchronous send command, which is not supported.
Action	Check your configuration: for example, an ACI client might be using Direct RPC.

04000007	**Receive not possible for clients (<class/server/service>)**
Explanation	An RPC client issues a receive command, which is not supported.
Action	Check your configuration: for example, an ACI client might be using Direct RPC.

04000008	**Direct RPC not started**
Explanation	An adapter service or a listener issues a call using Direct RPC, but the Direct RPC component is not started.
Action	Start the Direct RPC component, or ignore error if stopping of Direct RPC is intended.

04000010	**Command Service error: <detailed message>**
Explanation	An error occurred while processing a Command Service request.
Action	See error message for details.

04000011	**Compression not supported**
Explanation	The compression feature is not supported.
Action	Disable compression (use value `"NO_COMPRESSION"`) on the RPC client or RPC server.

04000012	**Publish and Subscribe not supported**
Explanation	The Publish and Subscribe functionality is not supported.
Action	Change your application.

04000013	**Character set for <character set type> not available**
Explanation	No character set is available for an RPC call.
Action	Contact Software AG support.

04000014	**Encoding LS=<char set name> not supported**
Explanation	The specified character set is not available. Only character sets supported by the Java JVM can be used.
Action	Change the configuration of the RPC client or server.

04000015	**<error message>**
Explanation	An error during character set conversion.
Action	See error message for details.

04000017	**Information Service error: <detail message>**
Explanation	An error occurred while processing an Information Service request.
Action	See error message for details.

15 Message Class 1018 - EntireX RPC-ACI Bridge

Messages have the format:

1018*nnnn*

where 1018 is the message class, and

nnnn is the message number in the range 0000 - 9999.

10180000

Explanation	This error can occur if the ACI server sends an invalid message. The ACI server is not able to return an error code. It can only send a valid message or quit the Broker communication.
Action	Correct the ACI server.

10180001   **Process request and reply: illegal action <number indicating action>**

Explanation	During request/reply processing the next action is illegal in the current state.
Action	Contact Software AG support.

10180002   **<Broker Exception with detailed message>**

Explanation	An exception with the Broker for ACI occurred during SEND.
Action	See message class and message code of the Broker exception and follow detailed message.

10180003	**\<Broker Exception with detailed message>**
Explanation	An exception with the Broker for ACI occurred during `RECEIVE`.
Action	See message class and message code of the Broker exception and follow detailed message.

10180004	**\<Broker Exception with detailed message>**
Explanation	An exception with the Broker for ACI occurred during shutdown and logoff.
Action	See message class and message code of the Broker exception and follow detailed message.

10180009	**\<Exception with detailed message>**
Explanation	An exception occurred while formatting the message. In most cases, the reason is an inconsistency in the metadata.
Action	See detailed message or contact Software AG support.

10180010	**Compression=off not possible**
Explanation	An RPC with compression 0 is not supported.
Action	Change compression to 1 or 2.

10180011	**Type of marshalling not allowed** **or** **Unsupported call type \<number of call type>**
Explanation	There are two reasons for this error code. 1. for the `marshalling` property the values 'cobol', 'natural', and '' are allowed. Reason: The value of the property may be invalid. or: The RPC from the client is in the wrong marshalling. 2. The RPC call type is wrong.
Action	Check configuration and client.

10180012	**\<Exception with detailed message>**
Explanation	This exception occurs in message processing.
Action	See detailed message and contact Software AG support.

10180013	**Preparing message failed, see log for details** **or:** **Message received from server does not fit buffer**
Explanation	There are two reasons for this error code:

1. Marshalling of message failed before sending to CICS ECI.

2. The received message has not the expected layout and the values cannot be un-marshalled from it.

Action	Check metadata (IDL, SVM, adapter connections) for consistency or contact Software AG support.

10180015	**Transformation error on request: <Exception with detailed message>**
Explanation	This exception occurs in message processing.
Action	See detailed message or contact Software AG support.

10180022	**Unsupported command <command>** **or** **Exception: <Exception with detailed message>**
Explanation	There are two reasons for this error code:

1. An unknown internal command is used.

2. An exception occurred during action processing.

The log files contain a stack trace.

Action	See detailed message and log files or contact Software AG support.

16　Preparing IBM CICS® for ECI

This chapter describes how to set up the External Call Interface (ECI) within CICS.

⚠ **Important:** If the terms and concepts in this chapter are unfamiliar to you, please work with the appropriate CICS system programmer. Only authorized personnel should make changes to mainframe computer systems.

Defining an ECI Service

The `DFH$SOT` group contains three TCP/IP services. For our purposes, the pertinent service is ECI. The ECI service has, defined in it, the TCP/IP port number through which the CICS region listens to the ECI. By default, IBM predefines the TCP/IP port number as "1435". If this port is already reserved for another CICS region, you may have to define a different port number in the ECI service. For more information, see your IBM documentation.

You must specify `SOCKETCLOSE(NO)` on the `TCPIPSERVICE`. If you need to specify a timeout for a task initiated using ECI over TCP/IP, specify an `RTIMOUT` value on the mirror transaction. Note that the standard mirror, `CPMI`, is defined with profile *DFHCICSA*, which specifies `RTIMOUT(NO)`. This means that long running mirrors will wait indefinitely for data, unless you specify a different `RTIMOUT` for the mirror transaction.

▶ **To define an ECI service without security**

1   Use `CEDA ALTER` to supply the `ECI TCPIPService` with a unique TCP/IP port number.

2   Use `CEDA ALTER` to set the `ECI ATtachsec` to "LOCAL"

▶ **To define an ECI service with security**

1   Copy the ECI `TCPIPService` to an ECIS `TCPIPService`.

2   Use `CEDA ALTER` to supply the ECIS `TCPIPService` with a unique TCP/IP port number.

3   Use `CEDA ALTER` to set the `ECIS ATtachsec` to "VERIFY".

Installation Verification

A successful installation should pass the following verification tests:

1. `CEMT INQUIRE TCPIPSERVICE(*)` should now display your services with status OPE. A status of CLO might indicate an already used (not unique) TCP/IP port number.

2. `CEMT I TCPIPSERVICE(ECI)` to verify TCPIPSERCICE in CICS.

3. `CEMT I TRA(CIEP)` to verify Transaction CIEP in CICS.

4. `CEMT I PROGRAM(DFHIEP)` to verify Program DFHIEP in CICS.

5. `CEMT I TRAN(CPMI)` to verify Transaction CPMI in CICS.

6. `CEMT I PROGRAM(DFHMIRS)` to verify Program DFHMIRS in CICS.

7. `CEMT I TD(CIEO)` to verify Program TQ Queue CIEO.

8. Verify that the groups `DFHISC DFHDCTG` and `DFHIPECI` are added to the active autoinstall

> 💡 **Tip:** You can check the mentioned programs and transactions and TD queues by using `CEMT VERIFY TCP`.

Error Handling

This section describes the handling of errors in the CICS ECI connection or the CICS ECI RPC server.

Problem	Handling
A CICS program sends abend code in response.	The CICS session is closed and the next call opens a different session.
The TCP/IP connection is lost with a `SocketTimeoutException`.	The CICS session is closed and the next call opens a different session.
The TCP/IP connection is lost with an `EOFException`.	■ The TCP/IP socket is closed and the next call opens a different session. ■ There are no further attempts to send bytes on the TCP/IP connection which received the `EOFException`.

17 Preparing for IMS

This chapter describes how to prepare for connections to IMS.

> ⚠️ **Important:** If the terms and concepts in this chapter are unfamiliar to you, please work with the appropriate IMS system programmer. Only authorized personnel should make changes to mainframe computer systems.

Extracting from COBOL Source

Use the IDL Extractor for COBOL. See the *EntireX Workbench* documentation for more information.

Extracting from Message Format Service (MFS)

To extract interface definitions from Message Format Service (MFS) with MID and MOD definitions, use a command line extractor. This extractor is run with the following command line:

```
java -classpath <suite installation ↵
folder>\IntegrationServer\packages\WmEntireX\code\jars\entirex.jar ↵
com.softwareag.entirex.ims.extractor.MFSExtractor <inputfile>
```

The input file has to be an MFS source with MID and MOD definitions. The output is an IDL file with the same name as the input file, except for the suffix ".idl" which is appended. The IDL file can be used to generate connections and adapter services with the webMethods Integration Server Wrapper (see the *EntireX Workbench* documentation) .

Troubleshooting

This section describes how to fix errors in the IMS Connect connection or the IMS Connect RPC server.

Problem	Solution
If the IMS Connect connection uses a port which is not related to IMS but to some other application, a call may result in the following message: `Unable to invoke adapter service <service name>. [ADA.800.500] WmEntireX: exception while executing service: 2011 0303 java.io.EOFException`	Correct the configuration of the port.
If the IMS Connect RPC server uses a port which is not related to IMS but to some other application, a call may result in the following message: `exception while executing service: 2011 0303 java.io.EOFException`	Correct the configuration of the port.