

webMethods EntireX

EntireX Adapter

Version 9.12

October 2016

This document applies to webMethods EntireX Version 9.12.

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

Copyright © 1997-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: EXX-EEXXADAPTER-912-20161012

Table of Contents

webMethods EntireX Adapter for Integration Server	v
1 Introduction to the EntireX Adapter	1
Prerequisites	2
Architecture and Components	2
Package Management	6
Adapter Connections	6
Adapter Services	7
Using Adapter Notifications	8
Viewing the Adapter's Update Level	8
What's New	9
2 Package Management	11
Package Dependency Requirements and Guidelines	12
Enabling and Disabling Packages	13
Importing and Exporting Packages	14
Group Access Control	14
3 EntireX Adapter Connections	15
Before Configuring or Managing Adapter Connections	16
Configuring Adapter Connections	16
Viewing Adapter Connection Parameters	18
Editing Adapter Connections	19
Deleting Adapter Connections	29
Enabling Adapter Connections	29
Disabling Adapter Connections	30
4 EntireX Adapter Services	31
Before Configuring or Managing Adapter Services	32
Configuring Adapter Services	32
Using Adapter Services	33
Viewing Adapter Services	36
Deleting Adapter Services	36
Runtime Connection Allocation for Adapter Services	37
Configure the Formatting of Decimal and Alphanumeric Parameters	37
Configuring Null Value Suppression	39
5 Listeners and Adapter Notifications	41
Remarks on Reliable RPC Listener Connections	42
Before Configuring New Listeners	45
Configuring New Listeners	45
Testing Listeners	47
Viewing Listener Parameters	47
Before Configuring or Managing Notifications	48
Configuring Adapter Notifications	49
Managing EntireX Adapter Notifications	50
Viewing Notifications	50
Editing Notifications	51

Deleting Notifications	51
6 Settings and Information	53
Adapter Settings	55
Connections Information	56
Built-in Services for Connections Information	56
Support Information	58
Services Information	58
Listeners Information	59
License Information	59
Application Monitoring	60
7 Built-in Services for Creating Document Types, Flows and IDL Files	61
8 Direct RPC	63
Configuring Direct RPC	64
Encoding for RPC Clients and Servers (Default and Available Codepages)	65
Monitoring	65
Services	66
Servers	66
Built-in Services for Direct RPC	66
Limitations of Direct RPC	68
9 IDL Extraction from Integration Server	69
Using the Service <code>pub.wmentirex.listener.generateIDLfromService</code>	70
Integration Server Data Types to IDL Mapping	71
10 Security Support	75
Security Support for Adapter Services	76
Security Support for Adapter Listeners	77
Support for SSL/TLS	78
11 Preparing IBM CICS for ECI	81
Defining an ECI Service	82
Installation Verification	82
Error Handling	83
12 Preparing for IMS Connect	85
Extracting from Message Format Service (MFS)	86
13 Converting IS Data Structures with the COBOL Converter	87

webMethods EntireX Adapter for Integration Server

The webMethods EntireX Adapter for Integration Server enables webMethods Integration Server to consume (IS outbound) as well as provide (IS inbound) EntireX RPC services. EntireX supports bidirectional integration with Natural, COBOL, PL/I, C and other applications on z/OS, z/VSE, BS2000 and other platforms. On Linux, UNIX and Windows also Micro Focus COBOL can be integrated with EntireX, bidirectionally.

This document describes how to install, configure and use the webMethods EntireX Adapter for Integration Server. 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. See *EntireX RPC Programming*.

This document covers the following topics:

<i>Introduction</i>	EntireX Adapters functionality and features such as calling remote procedures on mainframes and other hosts.
<i>Package Management</i>	How to set up and manage your EntireX Adapter packages and set up access control lists (ACL).
<i>Adapter Connections</i>	How to configure and manage EntireX Adapter connections.
<i>Adapter Services</i>	How to configure and manage EntireX Adapter services. EntireX Adapter services allow clients to interact with EntireX RPC servers, using a configured adapter connection.
<i>Listeners and Adapter Notifications</i>	How to configure and manage listeners and adapter notifications.
<i>Settings and Information</i>	Monitoring and tracing in the IS Administration Console.
<i>Built-in Services for Creating Document Types, Flows and IDL Files</i>	Built-in services are available in the WmEntireX package for creating document types, flows and IDL files from EntireX Adapter objects and/or IDL files.
<i>Direct RPC</i>	Direct communication from an RPC client or server to the EntireX Adapter.
<i>IDL Extraction from Integration Server</i>	Creating an IDL file can be automated using IDL Extractor for Integration Server.
<i>Security Support</i>	Security Support in the EntireX Adapter.
<i>Preparing IBM CICS for ECI</i>	The EntireX Adapter supports connections to IBM CICS® ECI to call programs in CICS. Here we describe how to set up the External Call Interface (ECI) within CICS.
<i>Preparing for IMS Connect</i>	The EntireX Adapter supports connections to IMS Connect. Here we describe how to extract from Message Format Service (MFS).
<i>Converting IS Data Structures with the COBOL Converter</i>	Adapter services for connection type COBOL Converter can convert Integration Server data structures to/from a byte array representing the COBOL binary data.

Related Literature

■ EntireX Workbench

- Integration Server Wrapper
- IDL Extractor for Integration Server

■ Error Messages

- Message Class 0800 - webMethods EntireX Adapter for Integration Server
- Message Class 2011 - Connections to IMS Connect
- Message Class 2012 - Connections to CICS ECI
- Message Class 2014 - COBOL Converter
- Message Class 2023 - Connections to ApplinX
- Message Class 0400 - Direct RPC
- Message Class 1018 - EntireX RPC-ACI Bridge

■ webmethods Product Suite

- *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 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"

■ Other

- 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>.

1 Introduction to the EntireX Adapter

■ Prerequisites	2
■ Architecture and Components	2
■ Package Management	6
■ Adapter Connections	6
■ Adapter Services	7
■ Using Adapter Notifications	8
■ Viewing the Adapter's Update Level	8
■ What's New	9

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.

Prerequisites

Supported Versions of Integration Server and EntireX Workbench

The EntireX Adapter supports Integration Server version 9.9, 9.10 or 9.12. For operating systems supported by the Integration Server, see <https://empower.softwareag.com/Products/Documentation/default.asp> under "webMethods Product Suite" > "Integration Server" > "System Requirements" (pdf). For design time, Software AG Designer version 9.10 or 9.12 must be installed. This includes the latest version of the EntireX Workbench.

Supported Platforms

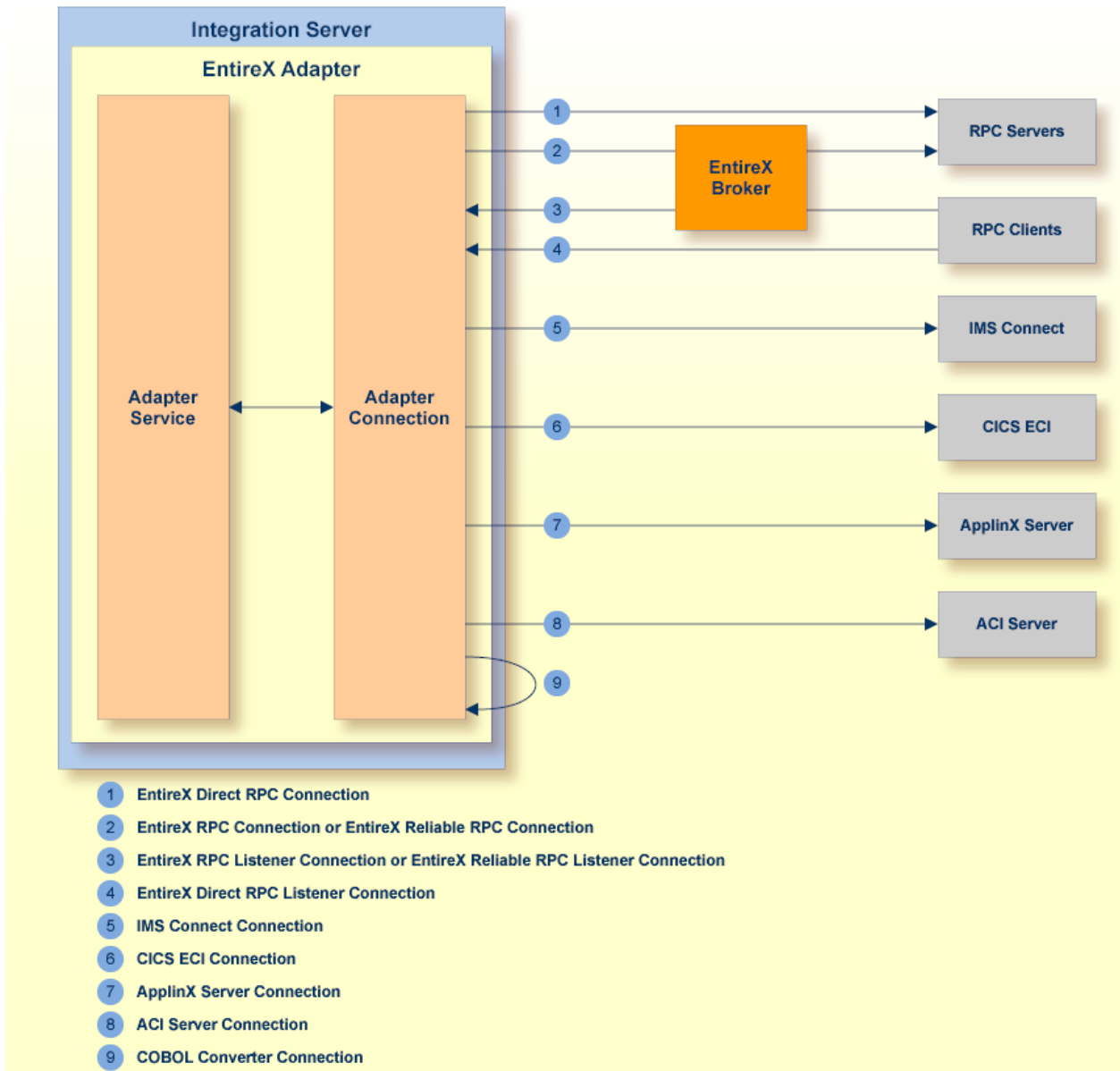
EntireX Adapter supports all platforms supported by the respective version of the Integration Server with the exception of the Mac OS platform. See <https://empower.softwareag.com/Products/Documentation/default.asp> under "webMethods Product Suite" > "Integration Server" > "System Requirements" (pdf).

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 following architectural elements are involved in the integration process:

- **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](#).

■ 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 *RPC-based Components* for details on EntireX 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
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
Reliable RPC Connection	EntireX Broker	x	-	one per IDL library	one adapter service for each IDL program	Reliable RPC Connection & Adapter Services	In

Connection Type	Connects to	Integration Server outbound	Integration Server inbound	IDL	Additional Nodes	Settings to generate	IDL Parameters
RPC Listener Connection	EntireX Broker	-	x	one per IDL library	one adapter listener per IDL library	RPC Connection & Listener	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
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 trigger, and a wrapping service	Reliable RPC Connection & Listener	In
IMS Connect Connection	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)
ACI Server Connection	EntireX Broker	x	-	one per IDL library	one adapter service for each IDL program	ACI Server Connection & Adapter Services	In, InOut, Out
COBOL Converter Connection	Integration Server internal	-	-	one per IDL library	one adapter service for each IDL program	COBOL Converter 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 [Package Management](#).
- 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](#).
- Control which groups have access to which adapter services. See [Group Access Control](#).
- Enable and disable packages. See [Importing and Exporting Packages](#).

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](#). 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](#).

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.

What's New

What's New in Version 9.12

- **User-defined Date and Time Formats**

The EntireX Adapter now supports user-defined formats for mapping the IDL data types Date and Time. See *Software AG IDL to Integration Server Mapping*.

- **Mixed-case Passwords for CICS ECI Connections**

You can now use mixed-case passwords for CICS ECI connections in EntireX Adapter. See [Connection Parameters for CICS ECI Connections](#).

- **COBOL MPO Support**

The EntireX Adapter - together with the IDL Extractor for COBOL - provides enhanced support of the COBOL MPO feature (multiple possible output). For details see *Workbench Enhancements* under *What's New in the EntireX Release Notes*.

- **Significant Change to Mapping of MPO Groups to IS Structures**

The mapping of MPO groups to Integration Server data structures and to the signature has been changed with this version to simplify the usage of the MPO construct in the Integration Server.

Previously the alternatives of an MPO group (groups in the IDL file with the attribute `CHOICE`) were mapped to arrays with either zero array elements (unselected alternative) or one array element (selected alternative). This has been changed such that arrays are no longer used. Unselected alternatives are suppressed and thus not visible in the output pipeline. The selected alternative is directly represented in the output pipeline without the enclosing array.

However, this enhancement means the new output signature is not compatible with existing Adapter services. This also affects the runtime behavior. If you use MPO groups, regenerate these Adapter services from the Designer and adapt the services that are using the new Adapter service to the changed output signature.



Note: This change does not apply to the RPC Connection types if the server mapping data is created as an SVM file.

[Provided with Fix 3 for EntireX version 9.9 and Fix 2 of EntireX version 9.10]

- **Alternative Mapping of IDL Data Types to Integration Server Data Types**

IDL data types can now be mapped to non-String Integration Server data types. The main use case is to support JSON document types in Integration Server. See *Integration Server Data Types to IDL Mapping* in the EntireX Adapter documentation and *Mapping IDL Data Types to IS Data Types* in the Integration Server Wrapper documentation.

[Provided with Fix 1 for EntireX Adapter version 9.12, requires Fix 1 of EntireX Workbench 9.12]

What was New in Version 9.10

A new connection type “COBOL Converter” has been introduced. Adapter services for this connection type can be used to convert Integration Server data structures from/to a byte array representing the COBOL binary data. See [Converting IS Data Structures with the COBOL Converter](#) for more information.

What was New in Version 9.9

■ Listeners Information

A new information screen provides statistics about the listeners of the EntireX Adapter. See [Listeners Information](#) under [Settings and Information](#).

■ License Handling

License handling for EntireX Broker connections now better reflects the EntireX pricing model:

- A license status line has been introduced for EntireX Broker connections (ACI connections or normal RPC connections).
- A warning message is given in the case of a possible license coverage mismatch.

See [License Information](#).

■ COBOL MPO Support

The EntireX Adapter - together with the IDL Extractor for COBOL - supports the concept of multiple possible output (MPO). See [Workbench Enhancements](#).

■ COBOL Server Using Fixed-length Tables in Variable Manner

The EntireX Adapter with IMS Connect connections - together with the IDL Extractor for COBOL - supports the concept of a COBOL server using data length to process a variable number of array elements. See [Workbench Enhancements](#).

■ Services for Creating Document Types, Flows and IDL Files

Built-in services are available in the WmEntireX package that can be used to create document types, flows and IDL files from EntireX Adapter objects and/or IDL files. See [Built-in Services for Creating Document Types, Flows and IDL Files](#)

■ New Namespace of Service `putDocDataToPipeline`

The namespace of this service changed from `com.softwareag.entirex.wmadapter.Deployment` to `pub.wmentirex.listener`. The service is still available using the old namespace. For a description of its usage see [Remarks on Reliable RPC Listener Connections](#).

■ New Advanced SSL/TLS Services for Direct RPC

New services are available to change the enabled TLS protocols and cipher suites. See [Built-in Services for Direct RPC](#).

■ EntireX Adapter Service `pub.wmentirex.listener:generateIDLfromService`

The meaning of the parameter `fileName` has changed, see [Using the Service `pub.wmentirex.listener:generateIDLfromService`](#).

2 Package Management

- Package Dependency Requirements and Guidelines 12
- Enabling and Disabling Packages 13
- Importing and Exporting Packages 14
- Group Access Control 14

The EntireX Adapter is provided as a package called WmEntireX. This chapter describes how to set up and manage your EntireX Adapter packages, set up access control lists (ACLs).

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* documentation 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.



Note: A disabled package will remain disabled until you explicitly enable it using the IS Administration Console. It will not be 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*.

3 EntireX Adapter Connections

- Before Configuring or Managing Adapter Connections 16
- Configuring Adapter Connections 16
- Viewing Adapter Connection Parameters 18
- Editing Adapter Connections 19
- Deleting Adapter Connections 29
- Enabling Adapter Connections 29
- Disabling Adapter Connections 30

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 that 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. See *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:

Connection Type	Note
EntireX RPC	1
EntireX Direct RPC	1
EntireX Reliable RPC	1
EntireX RPC Listener	2
EntireX Direct RPC Listener	2
EntireX Reliable RPC Listener	2
IMS Connect	1
CICS ECI	1
ApplinX	3
ACI Server	1
COBOL Converter	1



Notes:

1. The EntireX Workbench wizard additionally creates an EntireX Adapter service for each program of the IDL file; see [EntireX Adapter Services](#).
2. The EntireX Workbench wizard additionally creates an EntireX Adapter listener; see also [Listeners and Adapter Notifications](#).
3. The ApplinX Designer additionally creates an EntireX Adapter service for each ApplinX path procedure.

» 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 RPC Listener Connections and Reliable RPC Listener Connections](#)
- [Connection Parameters for Direct RPC Connections](#)
- [Connection Parameters for Reliable RPC Connections](#)
- [Connection Parameters for Direct RPC Listener Connections](#)
- [Connection Parameters for Connections to IMS Connect](#)
- [Connection Parameters for CICS ECI Connections](#)
- [Connection Parameters for ACI Server Connections](#)
- [Connection Parameters for ApplinX Connections](#)
- [Connection Parameters for COBOL Converter 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](#)
- [Direct RPC Connection](#)
- [Reliable RPC Connection](#)
- [RPC Listener Connection and Reliable RPC Listener Connection](#)
- [Direct RPC Listener Connection](#)
- [IMS Connect Connection](#)

- *CICS ECI Connection*
- *ApplinX Connection*
- *ACI Server Connection*
- *COBOL Converter Connection*

- 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	Deprecated. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See <i>SSL/TLS and Certificates with EntireX</i> .			
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

Parameter	Description	Default	Can be set from the EntireX Workbench	Can be set with service
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>.

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 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

Parameter	Description	Default	Can be set from the EntireX Workbench
Retype Logon Password	Verification of the password.		Not applicable
Encryption	Deprecated. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See <i>SSL/TLS and Certificates with EntireX</i> .	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 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	Deprecated. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See <i>SSL/TLS and Certificates with EntireX</i> .	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
Retype RPC Password	Verification of the password above.		Not applicable	Not applicable

Parameter	Description	Default	Can be set from the EntireX Workbench	Can be set dynamically with service
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 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 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

Parameter	Description	Default	Can be set from EntireX Workbench	Can be set dynamically with service
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 See also Support for SSL/TLS .		No	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	No
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. See Note .		Yes	Yes
Retype RACF Password / PassTicket	Verification of the password above		No	No
SSL Parameters	Truststore and optional certificate. Example: <code>trust_store=CACerts.jks</code> See also Support for SSL/TLS .		No	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	No



Note: By default the password is translated to uppercase. To use mixed-case passwords, set `watt.com.softwareag.entirex.wmadapter.cics.eci.mixedcase.password=true` under **Settings > Extended** on the Integration Server administration page.

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 <code><class>/<server>/<service></code> . The address may contain	RPC/SRV1/CALLNAT	Yes

Parameter	Description	Default	Can be set from the EntireX Workbench
	an asterix ('*') as a wildcard that is substituted by the IDL program name at runtime. This helps you to use only one connection for multiple IDL programs or adapter services.		
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	Deprecated. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See <i>SSL/TLS and Certificates with EntireX</i> .	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 <i>Writing ACI Servers for the RPC-ACI Bridge in COBOL Natural</i> 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 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

Connection Parameters for COBOL Converter Connections

Parameter	Description	Default	Can be set from EntireX Workbench	Can be set dynamically with service
Encoding	The character encoding of the COBOL binary data.	Cp037	Yes	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	No

Common Connection Management Properties (for all Connection Types)

Parameter	Description
Enable Connection Pooling	Enables the adapter to use connection pooling. Default: <code>true</code> . See Common Connection Management Properties (for all Connection Types) under <i>EntireX Adapter Connections</i> 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 <code>Minimum Pool Size</code> parameter. These connections remain in the pool regardless of how long they are inactive.
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.

4 EntireX Adapter Services

- Before Configuring or Managing Adapter Services 32
- Configuring Adapter Services 32
- Using Adapter Services 33
- Viewing Adapter Services 36
- Deleting Adapter Services 36
- Runtime Connection Allocation for Adapter Services 37
- Configure the Formatting of Decimal and Alphanumeric Parameters 37
- Configuring Null Value Suppression 39

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. 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 [EntireX 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. 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:

- **EntireX RPC Connection / EntireX Reliable RPC Connection**
 - Logon user
 - Logon password
 - RPC library name
 - RPC user
 - RPC password
 - Server address

See [Connection Parameters for RPC Connections](#) | [Connection Parameters for Reliable RPC Connections](#).

■ **EntireX Direct RPC Connection**

- RPC library name
- RPC user
- RPC password
- Server address

See [Connection Parameters for Direct RPC Connections](#).

■ **IMS Connect Connection**

- Logical terminal ID
- RACF user ID
- RACF password/PassTicket
- RACF group name
- RACF application name
- IMS datastore ID
- Maximum segment length

See [Connection Parameters for Connections to IMS Connect](#).

■ **CICS ECI Connection**

- RACF user ID
- RACF password/PassTicket. See [Note](#)
- CICS Mirror Transaction ID

See [Connection Parameters for CICS ECI Connections](#).

■ **EntireX ACI Connection**

- Logon user
- Logon password

See [Connection Parameters for ACI Server Connections](#).

■ **EntireX COBOL Converter Connection**

- cobolInput
- cobolOutput

See [Connection Parameters for COBOL Converter Connections](#) and [Converting IS Data Structures with the COBOL Converter](#) 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, modify each adapter service separately, using Designer. For each adapter service, select **Null Value Suppression** in **RPC Call**. You can fine-tune NVS with the following suppression modes:

Suppression Mode	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.

5

Listeners and Adapter Notifications

▪ Remarks on Reliable RPC Listener Connections	42
▪ Before Configuring New Listeners	45
▪ Configuring New Listeners	45
▪ Testing Listeners	47
▪ Viewing Listener Parameters	47
▪ Before Configuring or Managing Notifications	48
▪ Configuring Adapter Notifications	49
▪ Managing EntireX Adapter Notifications	50
▪ Viewing Notifications	50
▪ Editing Notifications	51
▪ Deleting Notifications	51

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.

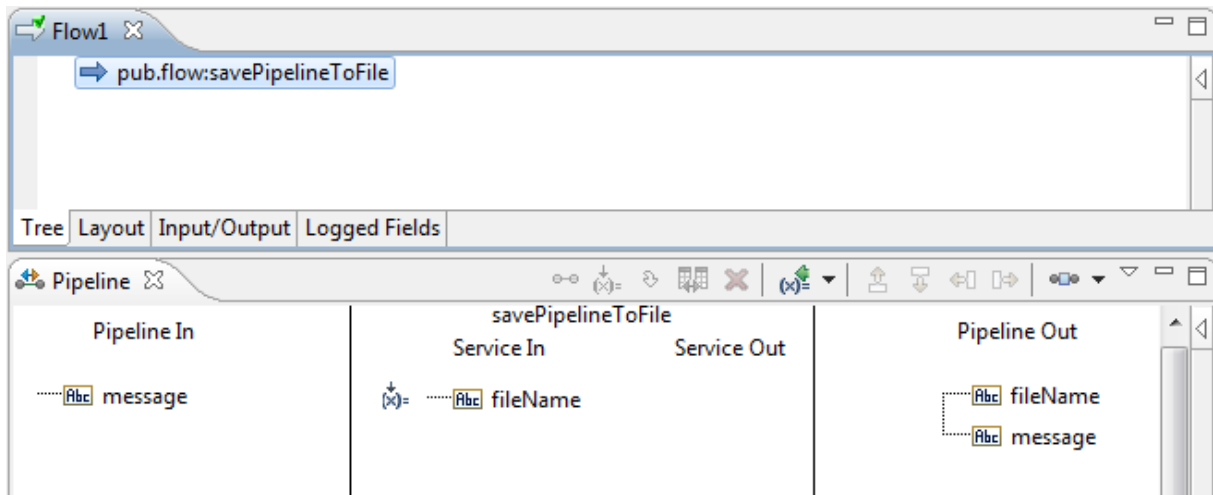
Adapter notifications monitor an EntireX RPC service and notify the Integration Server when a message arrives on the EntireX RPC service for Reliable RPC.

Important: Use the Integration Server Wrapper 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 `pub.wmentirex.listener.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` that simply dumps the pipeline to a file:



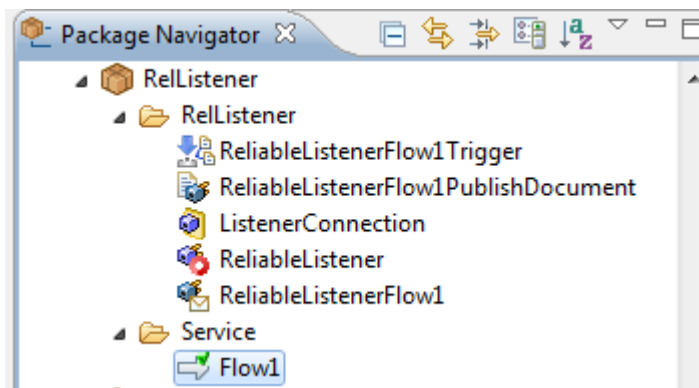
- The Software AG IDL Extractor for webMethods Integration Server extracts the following IDL:


```

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:

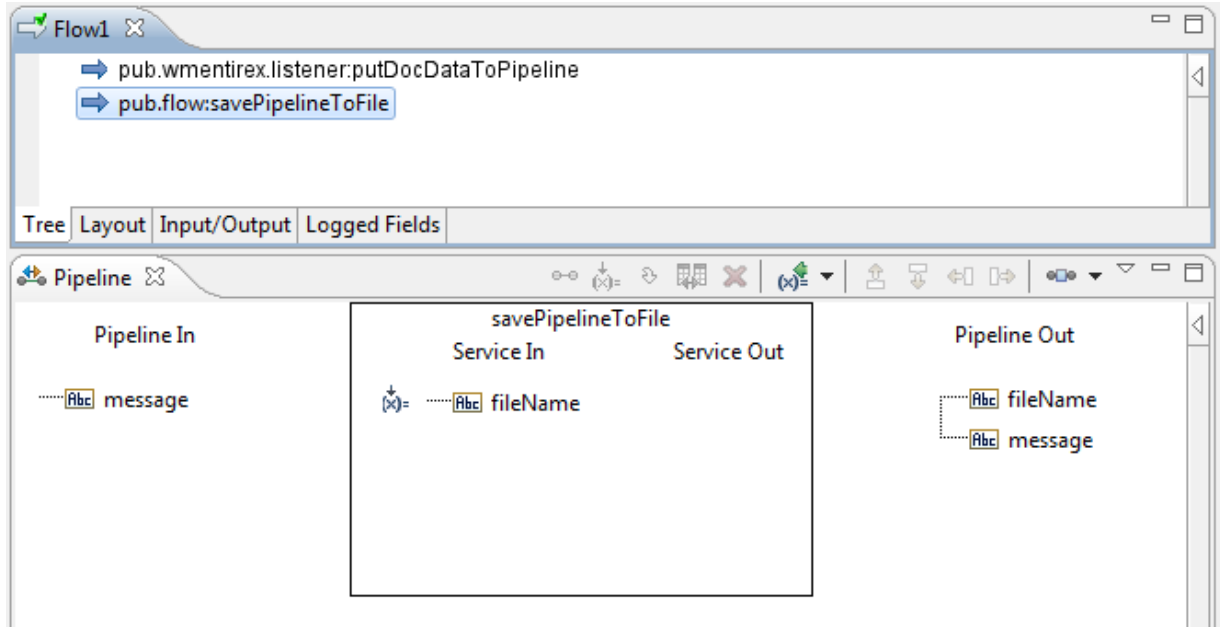
```

- <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 `pub.wmentirex.listener:putDocDataToPipeline` to `Flow1`:



- With the `pub.wmentirex.listener.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. See [Configuring Adapter Connections](#).
- 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. 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 that will be called in addition before the listener calls a service listed in Service Names . The specification <code>pub.wmentirex.listener:entryExitHandlerSpecification</code> describes the interface of the handler service.
Exit Handler Service	The name of an Integration Serverservice that will be called in addition after the listener has called a service listed in Service Names . The specification <code>pub.wmentirex.listener:entryExitHandlerSpecification</code> 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: <code>false</code> .
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. 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. 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. 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. 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](#).

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**.

6 Settings and Information

- Adapter Settings 55
- Connections Information 56
- Built-in Services for Connections Information 56
- Support Information 58
- Services Information 58
- Listeners Information 59
- License Information 59
- Application Monitoring 60

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

Adapter Settings

Changing Settings

EntireX Tracing	
Level of EntireX Tracing	1 (trace calls)
Tracefile Location (Folder)	C:\SoftwareAG910\IntegrationServer\instances\default\logs

Date and Time Patterns	
Date Pattern	MMM d, yyyy
Time Pattern	MMM d, yyyy h:mm:ss a
Locale for Date/Time Pattern	en

The first part of this screen shows the current level of EntireX Tracing.

The second part shows the setting for the Date and Time format patterns. You can change these values with the **Change Settings** link. This has immediate effect. For the Date and Time format patterns, the syntax of the Java class `java.text.SimpleDateFormat` is used. Localized date and time pattern strings are also supported. In these strings, the pattern letters described above may be replaced with other, locale-dependent, pattern letters using the specified locale. The default value appears when a field is left empty.

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. Parameters are either input or output (I/O).

Service	Parameter	I/O	Description
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.	brokerID String	I	ID of the broker to be pinged.
	refreshData String	I	The number of milliseconds to refresh the data. If refreshData is null or not a number, the default of 10 seconds is used.
	errorFlag String	O	"true" if an error occurred, "false" if the call succeeded.
	message String	O	The result of the ping call to the broker.
pub.wmentirex.connectioninfo:brokerList	refreshData String	I	The number of milliseconds to refresh the data. If refreshData is null or not a

Service	Parameter	I/O	Description
The service retrieves the list of all broker IDs configured in the connections.			number, the default of 10 seconds is used.
	brokerID String[]	O	The array of broker IDs.
pub.wmentirex.connectioninfo:listenerList The service retrieves name, broker ID and server address for each listener.	refreshData String	I	The number of milliseconds to refresh the data. If refreshData is null or not a number, the default of 10 seconds is used.
	listeners IData[]	O	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.	connection String	I	The name of the connection to ping, e.g. "folder:name".
	refreshData String	I	The number of milliseconds to refresh the data. If refreshData is null or not a number, the default of 10 seconds is used.
	errorFlag String	O	"true" if an error occurred, "false" if the call succeeded.
	message String	O	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	refreshData String	I	The number of milliseconds to refresh the data. If refreshData is null or not a number, the default of 10 seconds is used.
	connections String[]	O	The array of connection names in the format folder:connection.

Support Information

Connections, adapter services and adapter listeners contain metadata that 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.

Service	Parameter	I/O	Description
<code>pub.wmentirex.supportinfo:createInfo</code> The service extracts the generated metadata from all connections and adapter services in a package and writes this to a file in the folder <code><IntegrationServer_instance>/packages/WmEntireX/resources</code> .	<code>packageName</code> String	I	The name of a package. The generated metadata for all connections, adapter services and adapter listeners in this package is written to a file.
	<code>result</code> String	O	File <code><filename></code> created on host <code><hostname></code> .

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".

Listeners Information

This screen shows statistics about the listeners of the EntireX Adapter. For each listener, all services that have been invoked at least once during the runtime of the Integration Server are shown.

The following information is shown for each service invoked by a listener:

- name of the listener
- subprogram name and service name separated by '/'
- 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 listener and subprogram/service name can be reset with the link **Reset Statistics**.

License Information

This screen shows information on the license used by the EntireX Adapter.

- 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 connections: "Yes" or "No".
- License for EntireX Broker connections: "Yes" or a message that indicates a possible license coverage mismatch.
- Error message (only if the license is not granted).



Notes:

1. You must specify the location of the license file (file name with folder) with the **Change Settings** link.
2. The folder `<IntegrationServer_instance>/packages/WmEntireX` is not a safe folder for the license file, because the Integration Server cleans this folder on update.

Application Monitoring

Application Monitoring is an EntireX feature that enables you to monitor the response times in your distributed applications, and it also enables you to monitor certain error situations. See *Application Monitoring*.

➤ To enable application monitoring for Direct RPC, IMS Connect and CICS ECI

- 1 In the administration menu of the EntireX Adapter, choose **Application Monitoring**.
- 2 In the **Configuration** section, enable **Application Monitoring**.
- 3 In the configuration entry **Application Monitoring Data Collector ID**, enter the address of the Application Monitoring Data Collector.

The address has the format *host-name:port-number*, where *host-name* is the host on which the Application Monitoring Data Collector is running and *port-number* is the port number of the Application Monitoring Data Collector. The default port number is 57900.



Caution: If you have a Direct RPC scenario and you change the application monitoring configuration, all involved client applications and/or RPC servers have to be restarted (because they are caching information about the Adapter's application monitoring settings).

7 Built-in Services for Creating Document Types, Flows and IDL Files

The following built-in services are available in the WmEntireX package. They can be used to create document types, flows and IDL files from EntireX Adapter objects and/or IDL files. Parameters are either input or output (I/O).

Service	Parameter	I/O	Description
<p><code>pub.listener:createDocumentTypes</code></p> <p>This service creates document type objects, using the metadata stored in the listener, connection or adapter service object. Only objects created by the EntireX Adapter are supported. You can also use a Software AG IDL file as input.</p> <p>A document type named <code><programname>Request</code> is created for the input parameters of each program.</p> <p>A document type named <code><programname>Response</code> is created for the output parameters of each program.</p>	<code>input</code>	I	Name of a listener, adapter service, connection, or pathname of an IDL file.
	<code>packageName</code>	I	Name of the package where the created document types will be stored.
	<code>namespace</code>	I	Namespace of the folder where the created document types will be stored.
	<code>mapToString</code>	I	Defines how IDL data types are mapped to Integration Server data types. See <i>Mapping IDL Data Types to IS Data Types</i> in the Integration Server Wrapper documentation. Only used when a connection or IDL file is used as input.
	<code>result</code>	O	Either a success message if the creation was successful (which includes the number of created document types), or an error message.
<p><code>pub.listener:createFlow</code></p>	<code>listener</code>	I	Name of the listener.
	<code>packageName</code>	I	Name of the package where the created flow(s) will be stored. If this parameter is

Service	Parameter	I/O	Description
<p>This service creates flow objects for the specified EntireX Adapter listener object.</p> <p>For each service name specified in the definition of an EntireX Listener, an empty flow service with the appropriate input and output signature is created if this service does not exist. If the service exists it will remain unchanged.</p>			not specified, the flow(s) will be stored in the package of the listener.
	result	O	Either a success message if the creation was successful (which includes the number of created flow objects), or an error message.
<p>pub.listener:createIDL</p> <p>This service creates an IDL file, using the metadata which is stored in the listener, connection or adapter service object. Only objects created by the EntireX Adapter are supported.</p>	input	I	Name of a listener, connection or adapter service.
	filename	I	The name of the IDL file (with extension ".idl"). You cannot specify a folder name. The file is created in the folder <i><IntegrationServer_instance>/packages/WmEntireX/resources</i> .
	result	O	Either a success message if the creation was successful (which includes the full pathname of the IDL file), or an error message.

8 Direct RPC

- Configuring Direct RPC 64
- Encoding for RPC Clients and Servers (Default and Available Codepages) 65
- Monitoring 65
- Services 66
- Servers 66
- Built-in Services for Direct RPC 66
- Limitations of Direct RPC 68

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	<p>Fatal Fatal errors, Direct RPC cannot work correctly.</p> <p>Error Severe errors, Direct RPC continues to work correctly.</p> <p>Warning Errors with low severity.</p> <p>Info To follow the flow of requests between RPC clients, servers and Direct RPC.</p> <p>Debug All communication buffers will be written to the log file.</p> <p>Trace Internal use only (Software AG support).</p>	"Fatal"	Change logging level.	<p>Logging level.</p> <p>Log file <i>wmentirex.directrpcYYYYMMDD.log</i> is located in folder <i>logs</i> in the Integration Server installation folder. To change the default location, see Adapter Settings.</p> <p>It is not recommended to use "Info", "Debug" or "Trace" in a production environment.</p>
TCP Port Number	Port number	"1971"	Change port number.	TCP listening port for Direct RPC.
SSL Port Number	Port number	-	Change port number.	SSL/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.

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)	CP273	EBCDIC codepages

To specify the encoding of an RPC client or server, see the respective section of the 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/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 <code>logon</code>).
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 8 services for advanced configuration of the SSL transport:

- `pub.wmentirex.directrpc.admin:getSSLClientAuthentication`
- `pub.wmentirex.directrpc.admin:getTrustStoreAlias`
- `pub.wmentirex.directrpc.admin:getTLSProtocols`
- `pub.wmentirex.directrpc.admin:getCipherSuites`
- `pub.wmentirex.directrpc.admin:setSSLClientAuthentication`
- `pub.wmentirex.directrpc.admin:setTrustStoreAlias`
- `pub.wmentirex.directrpc.admin:setTLSProtocols`
- `pub.wmentirex.directrpc.admin:setCipherSuites`



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.

Use the `setTLSProtocols` service to change the set of enabled TLS protocols. The default is defined by the Java Runtime. Only protocols that are supported by the Java Runtime can be used. The set

of enabled and supported protocols are shown in the log file if the logging level is set to DEBUG or TRACE. Note that the `getTLSProtocols` service returns the value previously set by the `setTLSProtocols` service.

Use the `setCipherSuites` service to change the set of enabled cipher suites. The default is defined by the Java Runtime. Only cipher suites that are supported by the Java Runtime can be used. The set of enabled and supported cipher suites are shown in the log file if the logging level is set to DEBUG or TRACE. Note that the `getCipherSuites` service returns the value previously set by the `setCipherSuites` service.

Limitations of Direct RPC

- Reliable RPC is not supported.
- Command and Information Services (CIS) are not supported. Using this API 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 IDL Extraction from Integration Server

- Using the Service `pub.wmentirex.listener.generateIDLfromService` 70
- Integration Server Data Types to IDL Mapping 71

Integration Server services (e.g. flow services) can be called from an RPC client, using an adapter listener. To create a listener you need a Software AG IDL file, which describes the call interface (signature) of the Integration Server service. This IDL file can be created automatically using the IDL Extractor for Integration Server, a component of the EntireX Workbench that reads a package from the Integration Server and generates an IDL file from all services contained in the package.

As an alternative, you can create an IDL file from a specified subset of Integration Server services, using `service pub.wmentirex.listener.generateIDLfromService` from the `WmEntireX` package. This alternative is described below, along with an overview of data type mapping relevant for both approaches.

Using the Service `pub.wmentirex.listener.generateIDLfromService`

Service `pub.wmentirex.listener.generateIDLfromService` generates an IDL file for a given list of Integration Server services. For each service, a program in the IDL file is created. If the IDL file already exists, it is extended with the newly created program definitions. Duplicate definitions for the same program are possible. In this case, the IDL file has to be corrected manually. Note that this service creates/updates only the IDL file. Creating or updating a listener with the definitions of the IDL file has to be done with `webMethods Integration Server Wrapper` of the EntireX Workbench. The following parameters are relevant. Parameter direction can be In or Out:

Parameter	I/O	Description
<code>serviceName String</code>	I	The full name of the Integration Server service (e.g. <code>folder1.folder2:service</code>). Multiple services can be specified using a semicolon (;) as delimiter.
<code>fileName String</code>	I	The name of the IDL file (with extension ".idl"). You cannot specify a folder name. The file is created in the folder <code><IntegrationServer_instance>/packages/WmEntireX/resources</code> .
<code>libraryName String</code>	I	The library name to be used in the IDL file. If the IDL file already exists, this parameter is ignored.
<code>stringType String</code>	I	The IDL data type which is used for string data types. Possible values are <code>AV</code> , <code>AVn</code> , or <code>An</code> . This parameter is optional, default is <code>AV</code> .
<code>language String</code>	I	The target language for the RPC clients. Possible values are <code>COBOL</code> , <code>Natural</code> , <code>PL/I</code> , or <code>Other</code> .
<code>result String</code>	O	Either a success message if the generation was successful (which includes the full pathname of the IDL file) or an error message.

Integration Server Data Types to IDL Mapping

The signature of an Integration Server service specifies the data types for the parameters of the service. There are three data types: `String`, `Record` and `Object`. Parameters of type `String` are mapped to an IDL alphanumeric data type, parameters of type `Record` to IDL groups. Parameters of type `Object` cannot be mapped to an IDL data type and thus result in an error. If such an error occurs, the IDL program which contains the invalid data type mapping is written to the IDL file, however, all lines are comments. Starting with version 9.12 Fix 1 of the EntireX Adapter a subset of the Java Wrapper types are mapped to IDL types. This subset has been chosen so it is compatible to the JSON data type mapping in Integration Server.

Integration Server Object Type Java wrapper types	Software AG IDL Data Type	
	Adapter Version 9.12 Fix 1 or higher	Adapter Version 9.12 or lower
<code>java.lang.String</code>	See table below	See table below
<code>java.lang.Long</code>	N19	error
<code>java.lang.Integer</code>	I4	error
<code>java.lang.Float</code>	F4	error
<code>java.lang.Double</code>	F8	error
<code>java.lang.Boolean</code>	L	error
<code>byte[]</code>	BV, BV256 (depending on target language)	error
No wrapper type	Error; or BV, BV256 if extracting from EntireX Adapter Service ⁽¹⁾	Error; or BV, BV256 if extracting from EntireX Adapter Service ⁽¹⁾

⁽¹⁾ Starting with version 9.7, a parameter of an EntireX Adapter service of type `Object` is mapped to an IDL binary data type. Depending on the target language, it is either mapped to BV (for `Natural` and `Other`) or to BV256 (for `COBOL` and `PL/I`).

All parameters of type `String` are mapped to an IDL alphanumeric data type, available as variable (`AV`, `AVn`) or fixed (`An`) length. Which alphanumeric type is used is specified in the wizard page of the IDL Extractor for Integration Server or as a parameter of the generation service.

Parameters of type `String` may have an associated context type which is specified in the constraints of the parameter's properties. This content type influences the data type mapping. The content type was already considered in previous versions of the EntireX Adapter (version 9.6 or lower). However, its usage was not documented and the mapping was not yet complete. The handling of the content type was reworked with version 9.7. The following table shows both the new and the old mapping. The new mapping is different from the old one. If you want to use the old mapping with version 9.7 or higher, set `watt.com.softwareag.entirex.wmadapter.extractor.contenttype.handling.pre97=true` under **Settings > Extended** on the Integration Server administration page.

Integration Server Content Type	Software AG IDL Data Type			
	Adapter Version 9.7 or higher		Adapter Version 9.6 or lower	
boolean	L	Logical	L	Logical
decimal	Nx , $Nx.y$ or An , AV , AVn	Unpacked decimal or Alphanumeric ⁽¹⁾	$N10.2$	Unpacked decimal
float	F4	Floating point (small)	F4	Floating point (small)
double	F8	Floating point (large)	F8	Floating point (large)
Date	An , AV , AVn	Alphanumeric ⁽²⁾	D	Date
dateTime	An , AV , AVn	Alphanumeric ⁽²⁾	T	Time
base64Binary	An , AV , AVn	Alphanumeric ⁽²⁾	BV	Binary variable length
byte	I1	Integer (small)	I1	Integer (small)
unsignedByte	NU3	Unpacked decimal unsigned	I1	Integer (small)
short	I2	Integer (medium)	I2	Integer (medium)
unsignedShort	NU5	Unpacked decimal unsigned	I2	Integer (medium)
int	I4	Integer (large)	I4	Integer (large)
unsignedInt	NU10	Unpacked decimal unsigned	I4	Integer (large)
long	N19	Unpacked decimal	An , AV , AVn	Alphanumeric ⁽²⁾
unsignedLong	NU20	Unpacked decimal unsigned	An , AV , AVn	Alphanumeric ⁽²⁾
integer	N29	Unpacked decimal	An , AV , AVn	Alphanumeric ⁽²⁾
positiveInteger	NU29	Unpacked decimal unsigned	An , AV , AVn	Alphanumeric ⁽²⁾
nonPositiveInteger	N29	Unpacked decimal	An , AV , AVn	Alphanumeric ⁽²⁾
negativeInteger	N29	Unpacked decimal	An , AV , AVn	Alphanumeric ⁽²⁾
nonNegativeInteger	NU29	Unpacked decimal unsigned	An , AV , AVn	Alphanumeric ⁽²⁾
string	An , AV , AVn	Alphanumeric ⁽³⁾	An , AV , AVn	Alphanumeric ⁽²⁾
all others	An , AV , AVn	Alphanumeric ⁽²⁾	An , AV , AVn	Alphanumeric ⁽²⁾

**Notes:**

1. If the content type specifies a total number of digits and a number of fraction digits, then $Nx.y$ is used. If only a total number of digits is specified, then Nx is used. If no total number of digits is specified, or the total number exceeds the maximum allowed value, or the number of fraction digits exceeds the maximum allowed value, then a string according to ⁽²⁾ is used. The maximum allowed value for the total number of digits is 29, for COBOL it is 31. The maximum allowed value for the number of fraction digits is 7, for COBOL it is 31 and for Natural it is 29.

2. Depending on what is specified in the wizard page of the IDL Extractor for Integration Server or as a parameter of the generation service, the IDL data type AV , AV_n or A_n is used.
3. If the content type specifies a length len , then A_{len} is used. If the content type specifies a maximum length max , then AV_{max} or A_{max} is used, see ⁽²⁾ for the choice between A versus AV . If neither length nor maximum length is specified, the choice is the same as in ⁽²⁾.

10 Security Support

- Security Support for Adapter Services 76
- Security Support for Adapter Listeners 77
- Support for SSL/TLS 78

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 back-end systems. Access to these systems can be secured in the following ways:

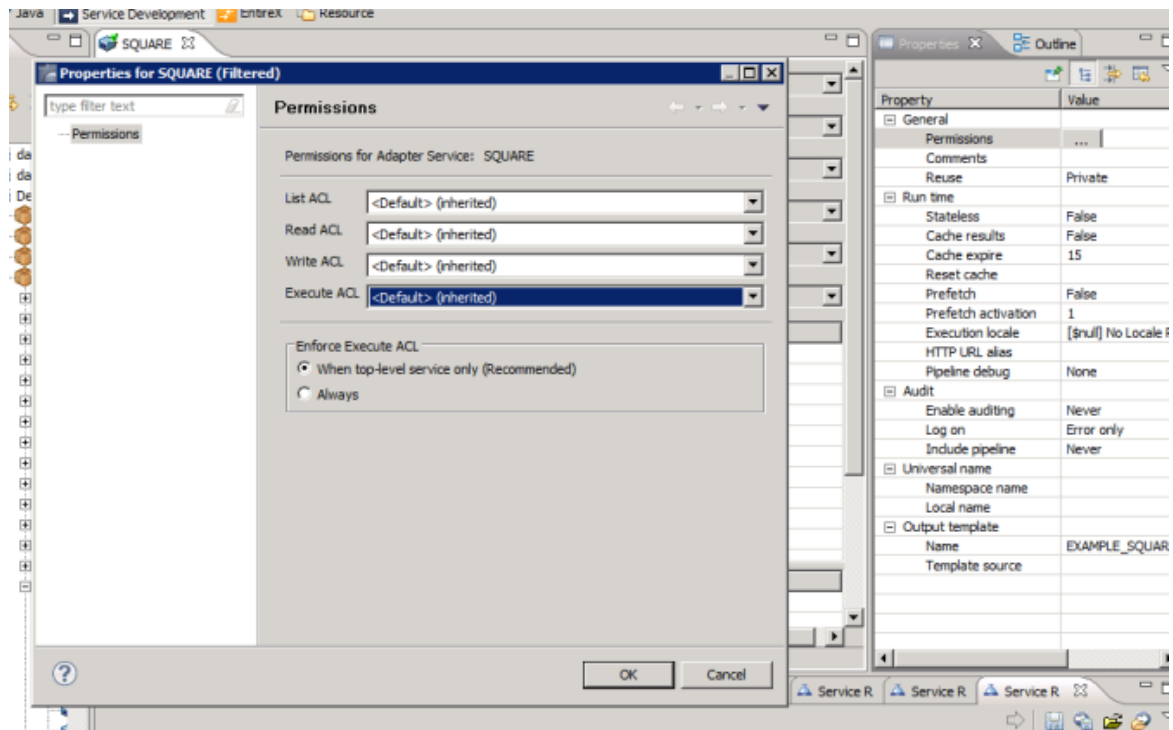
- **Provide security credentials in the adapter connection for security-protected back-end systems**
If access to a back-end system is security protected, security credentials (user ID and password) have to be provided in the corresponding adapter connection. This can be done when the connection is configured 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 [EntireX 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 that is specified 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 a broker that is configured to use security, 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 eight characters.

Support for SSL/TLS

Communication between the Adapter and the back-end systems is done using TCP/IP or, for secure communication, SSL/TLS. The configuration of SSL depends on the connection type:

Connection Type	Description	Notes
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: <code>ssl://host:1958?definition_of_sslparameters</code> . The syntax for defining SSL parameters is described in the notes below.	1,2,3,4,5
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.	1,3
Direct RPC	Specify the SSL port and SSL parameters in the configuration wizard for Direct RPC .	1,5

Notes

1. To operate with SSL, certificates need to be provided and maintained. Depending on the platform, Software AG provides default certificates, but we strongly recommend that you create your own. See *SSL/TLS Sample Certificates Delivered with EntireX* in the EntireX Security documentation.
2. Specify Broker ID and SSL parameters.

SSL transport will be chosen if the Broker ID starts with the string `ssl://`. Example of a typical URL-style Broker ID:

```
ssl://host:1958?definition_of_sslparameters
```

If no port number is specified, port 1958 is used as default.

3. If the SSL client checks the validity of the SSL server only, this is known as *one-way SSL*. The mandatory `trust_store` parameter specifies the file name of a keystore that must contain the list of trusted certificate authorities for the certificate of the SSL server. By default a check is made that the certificate of the SSL server is issued for the hostname specified in the Broker ID. The common name of the subject entry in the server's certificate is checked against the hostname. If they do not match, the connection will be refused. You can disable this check with SSL parameter `verify_server=no`.

If the SSL server additionally checks the identity of the SSL client, this is known as *two-way SSL*. In this case the SSL server requests a client certificate (the parameter `verify_client=yes` is defined in the configuration of the SSL server). Two additional SSL parameters must be specified on the SSL client side: `key_store` and `key_passwd`. This keystore must contain the private key of the SSL client. The password that protects the private key is specified with `key_passwd`.

The ampersand (&) character cannot appear in the password.

SSL parameters are separated by ampersand (&). See also *SSL/TLS Parameters for SSL Clients*.

Example of one-way SSL:

```
ssl://host:1958?trust_store=/temp/ExxCACert.jks&verify_server=no
```

Example of two-way SSL:

```
ssl://host:1958?trust_store=/temp/ExxCACert.jks&key_store=/temp/ExxJavaAppCert.jks&key_passwd=ExxJavaAppCert
```


4. Make sure the SSL server to which the EntireX Adapter (service or listener) connects is prepared for SSL connections as well. The SSL server can be EntireX Broker or Broker SSL Agent. See:
 - *Running Broker with SSL/TLS Transport* in the platform-specific Administration documentation
 - *Setting up and Administering the EntireX Broker SSL Agent* in the UNIX and Windows Administration documentation
5. For information on how to configure other EntireX components using SSL/TLS, see *Using SSL/TLS with EntireX Components*.

11

Preparing IBM CICS for ECI

▪ Defining an ECI Service	82
▪ Installation Verification	82
▪ Error Handling	83

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 `TCPIPService` 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 <code>SocketTimeoutException</code> .	The CICS session is closed and the next call opens a different session.
The TCP/IP connection is lost with an <code>EOFException</code> .	<ul style="list-style-type: none"> ■ 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 that received the <code>EOFException</code>.

12

Preparing for IMS Connect

- Extracting from Message Format Service (MFS) 86

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 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 and suffix ".idl". Use the IDL file to generate connections and adapter services with the Integration Server Wrapper.

Related Literature

- Extracting from a COBOL source is described under *IMS MPP Message Interface (IMS Connect)* in the IDL Extractor for COBOL documentation.
- For troubleshooting, see *Message Class 2011 - Connections to IMS Connect* in the Error Messages and Codes documentation.

13

Converting IS Data Structures with the COBOL Converter

Adapter services for the connection type "COBOL Converter" can be used to convert Integration Server data structures from/to a byte array representing the COBOL binary data. The IS data structure is described by a *Software AG IDL File* in the IDL Editor documentation (with an optional server mapping file, see *Server Mapping Files for COBOL*), which is usually created using the IDL Extractor for COBOL. The following COBOL interface types are supported:

- *CICS with DFHCOMMAREA Calling Convention*
- *CICS with DFHCOMMAREA Large Buffer Interface*
- *Batch with Standard Linkage Calling Convention*

The COBOL Converter functionality can be used in scenarios where COBOL binary data is either already available in IS, or required by an IS service. One example is working with files that contain COBOL data. Another typical example is COBOL binary data with a complex structure consumed or produced by components such as the AS/400 Adapter or the WebSphere MQ Adapter.

The conversion depends on the signature of the generated adapter service:

- **Input Only**

If the adapter service has only input parameters (structure `inRec`) they are converted to a COBOL binary data structure (byte array) and stored in the output parameter `cobolOutput`.

- **Output Only**

If the adapter service has only output parameters (structure `outRec`) the COBOL binary data structure (byte array) contained in the input parameter `cobolInput` is converted to the output parameters.

- **Input and Output**

If the adapter service has both input and output parameters (structures `inRec` and `outRec`), input parameter `cobolInput` is checked:

- If `cobolInput` is *empty*, the structure `inRec` is converted to a COBOL binary data structure (byte array) and stored in the output parameter `cobolOutput`.

- If `cobolInput` is *not empty*, the contained COBOL binary data structure (byte array) is converted to the output parameters.