

# webMethods EntireX

## EntireX Glossary

Version 10.9

April 2023

This document applies to webMethods EntireX Version 10.9 and all subsequent releases.

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

Copyright © 1997-2023 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-GLOSSARY-109-20230403**

## Table of Contents

|                        |    |
|------------------------|----|
| EntireX Glossary ..... | 1  |
| A .....                | 1  |
| B .....                | 3  |
| C .....                | 5  |
| D .....                | 7  |
| E .....                | 8  |
| F .....                | 12 |
| H .....                | 12 |
| I .....                | 12 |
| J .....                | 15 |
| L .....                | 16 |
| M .....                | 16 |
| N .....                | 18 |
| P .....                | 19 |
| Q .....                | 20 |
| R .....                | 20 |
| S .....                | 23 |
| T .....                | 26 |
| U .....                | 27 |
| V .....                | 28 |
| W .....                | 28 |
| X .....                | 30 |
| Z .....                | 31 |
| Numbers .....          | 31 |



## EntireX Glossary

---

### A

---

|                   |  |
|-------------------|--|
| Accounting        | Accounting in EntireX refers to the handling of accounting records for EntireX Broker that can be used for several purposes, including application chargeback (for apportioning EntireX resource consumption); performance measurement and trend analysis.           |
| ACI               | ACI stands for Advanced Communication Interface. ACI-based programming is the base technology of EntireX. It uses a traditional Application Programming Interface (API) approach for conducting client/server and peer-to-peer dialog between distributed processes. |
| ACI Binary Field  | An ACI binary field is a field that is transmitted in the ACI binary data section of an ACI request or reply. It has an associated 2-byte identifier and is transmitted as binary data.  |
| ACI Control Block | The fields contained in the control block of the EntireX Advanced Communication Interface (ACI) define the Broker functions to be performed.   |
| ACI Reply         | An ACI reply is a protocol data unit (PDU) that is transmitted between the broker kernel and a stub. It consists of an optional transport header, an SPFX (stub prefix) and various other sections.  |
| ACI Request       | An ACI request is a protocol data unit (PDU) that is transmitted between a stub and the broker kernel. It consists of an optional transport header, an SPFX (stub prefix) and various other sections.  |
| ACI Server        | An ACI server is a customer-written server program that uses ACI commands to call ACI services.  |
| ACI Service       | An ACI service is a customer-written service contained in or called by an ACI server.  |

|                                  |   |
|----------------------------------|---|
| ACI String Field                 | An ACI string field is a field of the form <code>keyword=value</code> that is transmitted in the ACI string data section of an ACI request or reply. It is transmitted in the single-byte character set of the stub.  |
| ADALNK                           | ADALNK is an Adabas module used by Entire Net-Work, which is an alternative transport method to TCP/IP.   |
| Advanced Communication Interface | See <a href="#">ACI</a> .   |
| 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. The heart of Application Monitoring is the EntireX Application Monitoring Data Collector, which collects the response time data of each involved software component of selected synchronous EntireX RPC services. The Application Monitoring Data Collector stores the KPI values in CSV (comma-separated values) files. The files can be processed by any third-party tool that supports CSV files, for example Microsoft Excel. Alternatively, you can hook in your own monitoring back end, using the callback user exit of the Data Collector. |
| Attach Server                    | An attach server is a server that registers with the EntireX Broker, can offer any of a range of services but never actually receives a request for a client application. When client requests cannot be satisfied due to busy or unavailable servers, the Broker notifies the attach server which starts another instance of the required service.   |
| Attribute File                   | The Broker attribute file contains a series of parameters (attributes) that control the availability and characteristics of clients and servers, as well as of the Broker itself. You can customize the Broker environment by modifying the attribute settings.   |
| Authentication                   | Authentication is the process by which the security system verifies that a client is actually who they claim to be. Authentication is performed at the user point of entry, which avoids the need for multiple sign-ons to the different security systems. The result of the authentication is propagated for use in the authorization process on the server side without the need to pass the user's password at execution time, even in an encrypted form.  |
| Authorization                    | Authorization is the process whereby a client's access rights to a specific service are verified. If you are using the System Authorization Facility (SAF) option, the authorization rules are stored in the server's native security system.   |

**Authorization Rule** An authorization rule is used to perform access checks for authenticated user IDs against lists of services defined within the rule. This feature is available on z/OS, Linux and Windows using EntireX Security on these platforms. Authorization rules can be stored in the Broker attribute file or in an LDAP repository.

## B

**Batch RPC Server** See [RPC Server for Batch](#) (z/OS, BS2000).

**Broker** EntireX Broker controls communication among the distributed application components within your IT architecture. The Broker supports many types of communication models: synchronous and asynchronous, client/server, peer-to-peer. The Broker also handles RPC communication.

**Broker ACI** See [ACI](#).

**Broker ActiveX Control** Broker ActiveX Control allows GUI application developers to use an ActiveX-based interface to access EntireX Broker. It can be used within ActiveX containers, such as Visual Basic, PowerBuilder, Delphi, Microsoft Excel, Microsoft Word.

**Broker Administration Tool** With the Broker Administration Tool you can manage Broker instances on your local Windows machine.

**Broker Agent** A broker agent is a “dummy broker” that acts as a transport protocol converter or relay node for broker calls.

See also [Broker HTTP\(S\) Agent](#), [Broker SSL Agent](#), [Broker TCP Agent](#).

**Broker Attribute File** The Broker attribute file contains a series of parameters (attributes) that control the availability and characteristics of clients and servers, as well as of the Broker itself. You can customize the Broker environment by modifying the attribute settings.

**Broker Command-line Utility** Software AG provides two command-line utility programs for use with EntireX Broker. Both utility programs use command-line parameters that specify various options and information to be built into a request. These utility programs are ETBINFO and ETBCMD.

- The command-line utility ETBINFO queries the Broker for different types of information, generating an output text string with basic formatting. This text output can be further processed by script languages. ETBINFO uses data descriptions called profiles to control the type of data that is returned for a request. ETBINFO is useful

for monitoring and administering EntireX Broker efficiently - for example, how many users are to run concurrently and whether the number of specified message containers is large enough.

- The command-line utility `ETBCMD` allows you to take actions - for example, purge a unit of work, stop a server, shut down a Broker - against EntireX Broker.

#### Broker HTTP(S) Agent

The EntireX Broker HTTP(S) Agent is a Java-based component that implements a Java servlet for servlet-enabled Web servers. It builds the bridge between a Web server and EntireX Broker in the intranet.

#### Broker Kernel

The role of the Broker kernel is to control communication among distributed application components. It supports many types of communication models (asynchronous and synchronous, conversational, request/reply, message-oriented) to support a wide variety of application integration requirements. The Broker kernel shields the communicating programs from platform and language-specific issues by mapping requests and replies to its interface programs, the ACI stubs.

#### Broker SSL Agent

The EntireX Broker SSL Agent is a gateway to the broker whenever direct SSL/TLS communication with the broker is not possible.

#### Broker Stub

A Broker stub is a module delivered with EntireX to communicate with a Broker. It is platform-dependent, but application-independent. A Broker stub implements an API (application programming interface) that allows programs written in various programming languages (Assembler | C | COBOL | PL/I) to access the EntireX Broker. *EntireX RPC Programming* also makes use of stubs to communicate with the EntireX Broker. The preferred transport method for communication is TCP/IP. Depending on the platform, SSL/TLS or Entire Net-Work can be used instead.

For Java programmers, the Java class library EntireX Java ACI provides similar functionality and access to the EntireX Broker.

In earlier versions of EntireX, the term "stub" was also used to refer to application-dependent, Designer-generated pieces of code for issuing and receiving remote procedure calls. These objects are now referred to as *interface objects*.

#### Broker TCP Agent

The EntireX Broker TCP Agent is a gateway to the broker whenever direct TCP/IP communication with the broker is not possible.

#### Broker Trace Utility

See [EntireX Trace Utility](#).



---

## C

---

|                                  |  |
|----------------------------------|--|
| C RPC Server                     | See <a href="#">RPC Server for C</a> .   |
| C Wrapper                        | EntireX C Wrapper provides access to RPC-based components from C applications. It enables you to develop both client and server applications.  |
| CCB                              | Conversation control block. In the context of persistence in EntireX, part of the UOW (unit of work).  |
| CentraSite                       | CentraSite is a jointly developed SOA repository from Fujitsu and Software AG. CentraSite acts as a web services and SOA asset management platform, holding all metadata assets, and offering reports on usage.  |
| CICS ECI RPC Server              | See <a href="#">RPC Server for CICS ECI</a> .  |
| CICS RPC Server                  | See <a href="#">RPC Server for CICS</a> (z/OS and z/VSE).  |
| CICS Socket Listener             | See <a href="#">EntireX CICS® Socket Listener</a> .  |
| Client                           | Within EntireX Broker nomenclature, a client is an application issuing Broker requests to get a service processed by a server.   |
| Client Program                   | A client program is a piece of software that issues a request and thus initiates communication. A client program uses a Broker stub.   |
| COBOL Mapping                    | See <a href="#">Server Mapping File for COBOL</a> .  |
| COBOL Wrapper                    | EntireX COBOL Wrapper provides access to RPC-based components from COBOL applications. It enables you to develop both client and server applications.  |
| Command and Information Services | EntireX Broker provides two internal services: Command Service and Information Services that can be used to administer and monitor the EntireX Broker. The command service allows you to issue a set of Broker commands; the information services provide you with various statistics to better administer and tune your Broker. Because these services are implemented internally, nothing has to be started or configured. You can use these services immediately after starting EntireX Broker. |
| Command Central                  | Software AG Command Central is a tool that enables you to manage your Software AG products remotely from one location. Command Central offers a browser-based user interface, but you can also automate tasks by using commands to remotely execute actions from   |

a terminal or custom script (for example CI servers such as Jenkins, or generic configuration management tools such as Puppet or Chef).

Command Central can assist with the following configuration, management, and monitoring tasks:

- Infrastructure engineers can see at a glance which products and fixes are installed, where they are installed, and compare installations to find discrepancies.
- System administrators can configure environments by using a single web user interface or command-line tool. Maintenance involves minimum effort and risk.
- Release managers can prepare and deploy changes to multiple servers using command-line scripting for simpler, safer lifecycle management.
- Operators can monitor server status and health, as well as start and stop servers from a single location. They can also configure alerts to be sent to them in case of unplanned outages.

#### Command-line Utility

In order to configure and administer the Broker properly and efficiently, you need certain types of information, such as how many users are to run concurrently, whether the number of specified message containers is large enough. EntireX Broker has a command-line utility that allows you to query the Broker for different types of information. Basic formatting of the output is available, but the real purpose of this tool is for further processing of the information in script languages or elsewhere.

#### Command Logging

Command logging is a feature to assist in debugging Broker ACI applications. A command in this context represents one user request sent to the Broker and the related response of Broker.

#### Conversation

Within EntireX Broker nomenclature, a conversation is one message or a set of messages exchanged between one client and one server in EntireX Broker. All requests must specify the conversation ID generated when the client issues the first send command.

#### Container Orchestration

Container Orchestration allows automated deployment, scaling and management of EntireX Brokers and EntireX RPC Servers running in Docker containers. Multiple instances of containers can be deployed on multiple nodes running on multiple hosts. Container orchestration allows load balancing and health monitoring. It supports takeover scenarios if, for example, one node or one container dies. The management of the worker nodes is done by the master node.

---

|                 |   |
|-----------------|---|
| Correlation ID  | Correlation ID is contained within the message descriptor. It is used in conjunction with the message ID to correlate a <code>REQUEST</code> and a <code>REPLY</code> . The correlation ID is used internally and is not directly exposed through the ACI. It is exposed indirectly; that is, the ACI field <code>CONV - ID</code> is used as the correlation ID when sending a <code>REQUEST</code> or a <code>DATAGRAM</code> . |
| Custom Wrappers | The EntireX Custom Wrappers are user-configurable, template-based wrappers and need a Software AG IDL file, a template (e.g., client or server) and the Software AG IDL Compiler.   |

## D

---

|                     |   |
|---------------------|---|
| Data Collector      | The Application Monitoring Data Collector collects the response time data of each involved software component of selected synchronous EntireX RPC services and stores the KPI (key performance indicator) values in CSV (comma-separated values) files. The files can be processed by any third-party tool that supports CSV files. For web-Methods Integration Server scenarios, the Data Collector is available inside the EntireX Adapter. |
| Data Compression    | Data compression within EntireX Broker allows you to exchange smaller packet sizes between clients and servers. This helps to reduce response time during transmissions as well as improve the overall network throughput, especially with low-bandwidth connections.   |
| DCOM Wrapper        | The EntireX DCOM Wrapper generates DCOM-enabled components using RPC technology. This so-called “wrapping” makes it possible to treat existing applications as ActiveX components.  |
| Default Broker View | The EntireX Default Broker View is part of the Designer. It displays the status of the EntireX Default Broker and the active RPC Services registered to it.   |
| Deferred Delivery   | A UOW (unit of work) can be created by the sender and accepted by EntireX Broker even if the intended receiver is not registered or active. The Broker holds the UOW until the receiver registers or becomes active. The UOW can then be delivered. This process is referred to as deferred delivery.   |
| Designer            | The EntireX design-time user interface is implemented as part of Software AG Designer and delivered as an Eclipse feature. It provides various wrappers, IDL extractors, editors and testers for generating and testing RPC communication between RPC servers and RPC client applications.  |

- .NET RPC Server      See [RPC Server for .NET](#).
- .NET Wrapper      The EntireX .NET Wrapper provides access to RPC servers for .NET client applications and access to .NET servers for any RPC client. The .NET Wrapper generation tools of the Designer take as input a Software AG IDL file, which describes the interface of the RPC, and generates C# classes that implement the methods and data types of the interface.
- DVIPA      Dynamic virtual IP address (DVIPA) is a z/OS Communications Server feature used to describe the automatic failover of TCP/IP stack instances within a Sysplex cluster.

## E

---

- Eclipse Plug-in      A plug-in is a component that extends functionality of the platform or other plug-ins. Eclipse employs plug-ins to provide all of its additional functionality on top of the Rich Client Platform.
- Eclipse Workbench      The Eclipse Workbench is a platform for development tools and provides the user interface structure for Eclipse. The Workbench is responsible for the presentation and coordination of the user interface. Each workbench window contains one or more perspectives. A perspective contains views and editors. It controls what appears in certain menus and toolbars. Multiple workbench windows can exist on the desktop concurrently.
- Encryption      Encryption is the process by which the information or data being sent back and forth between two computers (including the password submitted when logging on) is “encoded”, shielding it from view by unauthorized persons.
- For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See *SSL/TLS, HTTP(S), and Certificates with EntireX* in the platform-independent Administration documentation.
- EntireX      webMethods EntireX is a secure, robust, high-performance communication infrastructure that uniquely combines message queuing capabilities with built-in support for synchronous request/reply and conversational communication. It manages interactions for tightly coupled XML-enabled and non-XML enabled systems in time-critical user applications. EntireX's powerful and easy-to-use wrapping technology and flexible programming interfaces turn existing application functions into business services or web services. These features allow an organization to define an integration architecture for enhan-

---

|                                     |  |
|-------------------------------------|--|
|                                     | cing existing applications, linking up legacy applications with new standard software or adding new custom applications to the mix.  |
| EntireX Adapter                     | The webMethods EntireX Adapter for Integration Server enables webMethods Integration Server to consume (IS outbound) as well as provide (IS inbound) EntireX services. EntireX supports bidirectional integration with Natural, COBOL and PL/I on z/OS, BS2000, z/VSE and other platforms. This includes zero footprint integration with CICS, IMS and IBM i (AS/400).   |
| EntireX Broker                      | EntireX Broker controls communication among the distributed application components within your IT architecture. The Broker supports many types of communication models: synchronous and asynchronous, client/server, peer-to-peer. The Broker also handles RPC communication.  |
| EntireX Broker ACI                  | See <a href="#">ACI</a> .  |
| EntireX CICS® Socket Listener       | The EntireX CICS® Socket Listener is a remote connector on z/OS or z/VSE to call CICS mainframe programs.  |
| EntireX Design Time                 | See <a href="#">Designer</a> .   |
| EntireX Java ACI                    | EntireX Java ACI is a Java class library that provides access to the EntireX Broker ACI for Java programmers. It covers the whole EntireX Broker ACI which enables you to write both client and server applications in Java. Any of these can then interact with each other and with other applications written in other languages on the same network using EntireX Broker. The EntireX Java ACI also contains the framework necessary for Java RPC requests. |
| EntireX Java Runtime                | EntireX Java Runtime is a protocol that enables RPC-based communication between clients and servers in a Java environment.   |
| EntireX Mainframe Broker Monitoring | EntireX Mainframe Broker Monitoring is a package with which you can monitor EntireX Broker on mainframe platforms z/OS and BS2000. Define an instance of your mainframe broker, using Command Central under Linux or Windows. This instance - a so-called proxy - holds connection information to the remote broker.   |
| EntireX Mini Runtime                | The EntireX Mini Runtime is a set of components that can be used for the deployment of applications using the Broker ACI, the Broker RPC, and - under Windows only - DCOM Wrapper objects and the Broker ActiveX Control. Unlike the full EntireX Runtime, the Mini Runtime does not include Java capabilities. The EntireX Mini Runtime is provided as a separate package in the Software AG Installer.   |

|                        |  |
|------------------------|--|
| EntireX RPC            | EntireX RPC enables programmers to develop client applications that access RPC servers (for example CICS, Batch or IMS RPC Server) providing RPC services (for example COBOL batch programs, CICS transaction, Java methods etc.) on mainframe, Linux and Windows platforms. EntireX RPC opens the door for various environments such as web services/SOA, webMethods IS, Java, .NET etc. to access EntireX-enabled RPC server environments.   |
| EntireX RPC-ACI Bridge | The EntireX RPC-ACI Bridge allows standard RPC clients to communicate with an ACI server. The RPC-ACI Bridge transforms RPC requests from clients into ACI messages.   |
| EntireX RPC Servers    | The EntireX RPC Servers (available under CICS, z/OS Batch, Linux, Windows and Java) execute servers on RPC client requests. Using EntireX RPC Wrapper technology of the Software AG Designer, you can generate these servers together with integrated wrappers (server interface objects). The Designer tool you choose to generate these servers depends on your environment, for example Java Wrapper, COBOL Wrapper, C Wrapper, XML/SOAP Wrapper, etc.  |
| EntireX Runtime        | The EntireX Runtime consists of all the parts required to run applications that were created or were enabled for distributed computing using EntireX technology (for example, ACI and RPC components, ActiveX Control etc.).   |
| EntireX Security       | EntireX Security provides protection for applications utilizing EntireX Broker. Functionality includes authentication for distributed application components as well as authorization to start and execute services. All user and resource definitions are managed through RACF, CA Top Secret and CA ACF2. EntireX Security comprises two separate components, one for the Broker kernel and one for each Broker stub. See <i>Introduction to EntireX Security</i> . For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See <i>SSL/TLS, HTTP(S), and Certificates with EntireX</i> in the platform-independent Administration documentation. |
| EntireX Support Tool   | The EntireX Support Tool provides information about the internal versions of the EntireX modules (executables, shared libraries and JAR files).  |
| EntireX Trace Utility  | Formerly referred to as “EntireX Broker Trace Utility”. The EntireX Trace Utility processes and analyzes a trace (or log) file of the EntireX broker kernel or broker stubs. Trace files from any broker version since 2.1 on all platforms can be used. The tool extracts from the trace file all logged broker calls and generates a CSV file (comma-separated values), where each line represents one broker call and   |

---

|                               |  |
|-------------------------------|--|
|                               | <p>the columns represent the most important parameters of the call. The CSV file can be used with other tools, such as Microsoft Excel.</p>  |
| EntireX Workbench             | See <i>Designer</i> . Designer was referred to in earlier EntireX versions as EntireX Workbench  |
| EntireX XML Adapters          | EntireX XML Adapters provide access to a wide range of enterprise resources (ERP and CRM systems, database management systems, etc.) by exposing services from those systems as web services. Using the EntireX XML/SOAP Wrapper, the web services defined with EntireX XML Adapters can be made available to EntireX or Natural clients.  |
| EntireX XML Mapping Editor    | The EntireX XML Mapping Editor allows you to map XML document structures to IDL libraries, programs and parameters. The mappings can be defined for the request and response to the server application, or from the server to the client. The input for the XML Mapping Editor can be a Software AG IDL file and/or an IDL-XML mapping file (perhaps produced by a previous XML Mapping Editor session or by importing a WSDL file, XML Document or XML Schema). The output is an IDL-XML mapping file, other XML structure definitions (such as sample XML files), and perhaps a created or changed IDL file. |
| EntireX XML to IDL Generators | XML to IDL Generators generate Software AG IDL files and related mapping files from DTD files, XML Schema files, XML files and WSDL files.   |
| EntireX XML/SOAP Runtime      | The EntireX XML/SOAP Runtime enables XML-based clients to communicate with an EntireX/Natural RPC Server; and it enables EntireX RPC clients to communicate with applications that have an XML or SOAP interface (via XML/SOAP Runtime).   |
| ETBCMD                        | The command-line utility ETBCMD allows you to take actions - for example, purge a unit of work, stop a server, shut down a Broker - against EntireX Broker.  |
| ETBINFO                       | The command-line utility ETBINFO queries the Broker for different types of information, generating an output text string with basic formatting. This text output can be further processed by script languages. ETBINFO uses data descriptions called profiles to control the type of data that is returned for a request. ETBINFO is useful for monitoring and administering EntireX Broker efficiently - for example, how many users are to run concurrently and whether the number of specified message containers is large enough.  |

**Explicit Logon** Using the LOGON function as the first function is called explicit logon. See also *Implicit Logon*.

**Extractor** An extractor is a tool contained in the Designer to extract a Software AG IDL file.

---

## F

**Failover** Feature used to describe the control switch between redundant resources in the event of a resource failure.

See also *High Availability in EntireX*.

---

## H

**High Availability** Under High Availability we understand an environment with engineered redundancy which, if any one component fails, guarantees the integrity of the system as a whole. To achieve high availability, EntireX uses existing third-party clustering technology.

**HiperSockets** HiperSockets is an IBM technology for offering TCP/IP communications between partitions at in-memory speed.

---

## I

**IDL** Software AG IDL (Interface Definition Language) is a language that lets a program or object (the client) written in one language communicate with another program written in another language (the server). An interface definition language works by requiring a program's interfaces to be described in an interface object or slight extension of the program that is compiled into it.

Software AG IDL is used to define an interface between the client and the server. These definitions are contained in a Software AG IDL file. IDL compilers read IDL files and generate interface objects or descriptions from the definitions they contain.

**IDL Compiler** The Software AG IDL Compiler generates interface objects, skeletons and wrappers. It uses a Software AG IDL file and a template file that controls the generated output.

**IDL Editor** The Software AG IDL Editor is a syntax-aware editor for Software AG IDL files. A content outline view enables operations on the IDL tree. Other features include syntax highlighting, content assist and a Problems view for resolving IDL syntax errors.



---

|                                      |   |
|--------------------------------------|---|
| IDL Extractor for COBOL              | The Software AG IDL Extractor for COBOL enables you to extract the interface of a COBOL server and transforms it into a Software AG IDL and a Software AG server mapping file. Both files are required to provide access for any RPC client to the COBOL server.  |
| IDL Extractor for Integration Server | The Software AG IDL Extractor for webMethods Integration Server is a wizard that reads a package from the Integration Server and generates a Software AG IDL file from existing services and nodes. Each service results in a program in the IDL file. All parameters of the services are mapped to an IDL alphanumeric data type, available as variable (AV) or fixed (A <i>n</i> ) length. From EntireX Adapter version 10.5, you can select individual services and nodes; with earlier versions all services and nodes are extracted. |
| IDL Extractor for Natural            | The Software AG IDL Extractor for Natural extracts a Software AG IDL definition from a Natural source in a Natural project in Eclipse, or from an object within a Natural RPC environment.  |
| IDL Extractor for PL/I               | The Software AG IDL Extractor for PL/I extracts a Software AG IDL file from a PL/I source. The PL/I source can be located in the file system or accessed remotely within a PL/I RPC environment definition.   |
| IDL Extractor for WSDL               | The Software AG IDL Extractor for WSDL is a wizard that generates web service client artifacts from a WSDL file. With these artifacts, EntireX RPC client applications can access external web services.  |
| IDL Extractor for XML Document       | The Software AG IDL Extractor for XML Document generates an IDL File and a related XML mapping file (XMM) from a given XML document.  |
| IDL Extractor for XML Schema         | The Software AG IDL Extractor for XML Schema generates an IDL File and a related XML mapping file (XMM) from given XML schema files.  |
| IDL File                             | A Software AG IDL file contains definitions of the interface between client and server. The IDL file is used by Software AG wrappers to generate RPC clients, RPC servers and tester etc. on the basis of these definitions. The IDL file can be edited by the IDL Editor provided by plug-ins for Eclipse.   |
| IDL Template                         | An IDL template file contains the rules that the Software AG IDL Compiler uses - together with the IDL file - to generate interface objects, skeletons and wrappers for a programming language. EntireX provides several templates for various programming languages.   |
| Impersonation                        | EntireX Security allows a server to be started under  |

- the user ID of the calling client or
- a defined user ID (obtained from the registry, for example).

Authorisation is carried out appropriately in either case.

### Implicit Logon

When the EntireX Broker is configured with `AUTOLOGON` it is not required to use the `LOGON` function as the first function. Clients can start a broker sessions with a `SEND` function; servers with `REGISTER`. Before executing the `SEND/REGISTER` the EntireX Broker executes an implicit logon. See also *Explicit Logon*.

### IMS Connect RPC Server

See [RPC Server for IMS Connect](#).

### IMS RPC Server

See [RPC Server for IMS](#).

### Integration Server

The webMethods Integration Server is a Java-based, multiplatform enterprise integration engine supporting the execution of services to perform integration logic such as data mapping and communication with other systems.

### Integration Server Wrapper

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

### Interface

A specification of a set of methods, which implement functionality of the same subject. An interface is a strongly typed, semantic contract between client and server.

In EntireX Broker, the prominent interface is the ACI in its various implementations (for example, `broker.dll`).

In the context of RPC programming, a client/server interface contains one or more remote procedure call operations and associated data, each of which may have input and output parameters associated with it.

### Interface Object

An interface object is an application-dependent generated piece of code to issue/receive remote procedure calls (RPCs).

In EntireX RPC programming, a *client-side interface object* is a piece of code generated by a wrapper contained in the Designer that allows you to call a remote RPC service. The signature of each RPC service is described by a Software AG IDL file (Interface Definition Language) which must be provided as input for a wrapper. The main task of the *client-side interface object* is to marshal/unmarshal the RPC parameters and to call the underlying transport layer using an available Broker stub.

A *server-side interface object* is a piece of code also generated by a wrapper contained in the Designer to be used on the server side. The main task of the server-side interface object is to marshal/unmarshal the parameters received/returned by an RPC server.



**Note:** In earlier versions of EntireX, interface objects were referred to as “stubs”.

## Internationalization

Software internationalization is the process of designing products and services so that they can be adapted easily to a variety of different local languages and cultures. Internationalization within EntireX means internationalization of messages: the incoming and outgoing messages are converted to the desired codepage of the platform in use.

## IUBL

The Internal User Buffer Length (IUBL) is the EntireX Broker parameter that sets the maximum length (in bytes) of the buffer that can be passed from the caller to the EntireX Broker. It must be large enough to hold the maximum send length plus receive length required for any caller program plus an administration overhead of about 600 bytes. Default is 8000. The maximum size of IUBL is the same as the maximum value of the Adabas parameter LU.

## J

### Java ACI

EntireX Java ACI is a Java class library that provides access to the EntireX Broker ACI for Java programmers. It covers the whole EntireX Broker ACI which enables you to write both client and server applications in Java. Any of these can then interact with each other and with other applications written in other languages on the same network using EntireX Broker. The EntireX Java ACI also contains the framework necessary for Java RPC requests.

### Java Runtime

EntireX Java Runtime is a protocol that enables RPC-based communication between clients and servers in a Java environment.

### Java RPC Server

See [RPC Server for Java](#).

### Java Wrapper

The EntireX Java Wrapper provides access to EntireX RPC-based components from Java applications. With EntireX Java RPC you can develop both client and server applications written in Java. Java applets can also be used as EntireX RPC clients.

### Java Wrapper for Natural

The EntireX Java Wrapper for Natural allows you to generate EntireX Java client interface objects from Natural subprograms in a Natur-

alONE project in Eclipse. The generated Java client interface objects can be used by Java application developers to access Natural server components, using EntireX/Natural RPC.

## L

---

|                         |   |
|-------------------------|---|
| Lightweight Diagnostics | Lightweight Diagnostics is a subset of the Trace facility of Entirex Broker. With Lightweight Diagnostics you can run the Broker kernel without needing to run a heavyweight trace all the time. Lightweight Diagnostics report on problems that impact the performance and the functionality of the Broker kernel.                                   |
| Listener                | A listener is an EntireX component that listens on incoming requests and sends RPC requests in the background (for example XML or MQ).  |
| Listener for IBM MQ     | The EntireX Listener for IBM® MQ runs as a listener on an IBM MQ queue and passes messages to an RPC server. It is used to send messages received from an IBM MQ queue to an RPC server application. This means that existing RPC servers can be used for communication with IBM MQ.  |
| Listener for XML/SOAP   | The EntireX Listener for XML/SOAP runs in a servlet-enabled Web server with an installation of the Software AG Web Services Stack. It plugs the web service's archives (Designer file with extension .aar) into Web servers. Typically the web service clients access the Listener for XML/SOAP and send/receive XML/SOAP documents using HTTP/HTTPS. |
| Load Balancing          | A network-based technique to distribute common workload across multiple back-end services that can be evenly distributed or routed based on weighted inference of a key performance attribute (commonly called filtering).  |
| Locale                  | A locale provides a means of identifying a specific region for the purposes of internationalization and localization. EntireX sends properties of the operating system locale to the Broker, using the locale string.   |
| LPAR                    | A logical partition (LPAR) describes the virtualization of the System z hardware platform; if one LPAR goes down it has no effect on any others running on the same hardware.   |

## M

---

---

|             |  |
|-------------|--|
| Mainframe   | Some sites that deploy one or more Linux servers refer to these as 'mainframes'. The EntireX documentation uses the term 'mainframe' specifically to refer to z/OS, z/VSE (IBM) and BS2000 (Fujitsu) systems as opposed to Linux machines.   |
| Marshalling | Marshalling is the process of packaging interface data into RPC packets for delivery across process or network boundaries. This includes converting data from its true data type (for example int, long etc.) to an internal representation. Converting data from an internal representation to its true data type (for example int, long etc.) is called unmarshalling. RPC programming thus has an advantage over native ACI coding in the area of transparent data translation.   |
| Message     | In an environment that uses EntireX as communication technique, clients and servers communicate by exchanging messages. For example, a message can be both a request for data and the returned data itself.  |
| Message ID  | Message ID identifies a message and can be used in several ways: <ul style="list-style-type: none"><li>■ It allows a REPLY to be associated with a specific REQUEST.</li><li>■ It can be used on a RECEIVE to limit the RECEIVE to a specific message. If message ID is not specified, the next available message is returned.</li><li>■ It can be used by a client program when sending a REQUEST or a DATAGRAM. If message ID is not specified, the system generates one and returns it upon completion of the SEND.</li><li>■ When sending a REPLY, the server program should specify the message ID of the associated REQUEST.</li></ul>   |
| Messaging   | <p>Messaging is defined as the creation, exchange, storage and management of files in a communications network. Those files can be text, images, voice, telex, fax, e-mail, paging, and Electronic Data Interchange (EDI).</p> <p>In programming, messaging is defined as the exchange of specially formatted data describing events, requests, and replies (messages) to a messaging server, which acts as a message exchange program for client programs. With messaging, programs can share a common message-handling code, isolate resources and interdependencies, and easily handle an increase in message volume. Messaging also makes it possible for programs to communicate across different programming languages and operating systems. All that each envir-</p> |

onment needs to understand is the common messaging format and protocol.

**Mini Runtime**

The EntireX Mini Runtime is a set of components that can be used for the deployment of applications using the Broker ACI, the Broker RPC, and - under Windows only - DCOM Wrapper objects and the Broker ActiveX Control. Unlike the full EntireX Runtime, the Mini Runtime does not include Java capabilities. The EntireX Mini Runtime is provided as a separate package in the Software AG Installer.

---

## N

**NABS**

The Number of Attached Buffers (NABS) is the Broker parameter that specifies the number of attached buffers to be used. The default value is 10. An attached buffer is an internal buffer used for interregional communication. An attached buffer pool equal to the NABS value multiplied by 4096 will be allocated. This buffer pool must be large enough to hold all data (IUBL) of all parallel calls to EntireX Broker.

**Natural Mapping**

See [Server Mapping File for Natural](#).

**Natural Wrapper**

The Natural Wrapper allows you to develop Natural client applications that access RPC-based server components, and to create Natural RPC server skeletons you can use as a basis to write a Natural RPC server that can be accessed by RPC clients.

**NCQE**

The Number of Command Queue Elements (NCQE) is the Broker parameter that specifies the number of command queue elements to be established. Its value determines the maximum number of EntireX Broker commands that can be queued and/or be in progress at any time. Each command is assigned a command queue element. This element is released when the user has received the results of the command or when the user has been timed out.

**.NET RPC Server**

See [.NET Wrapper](#)

**.NET Wrapper**

The EntireX .NET Wrapper provides access to RPC servers for .NET client applications and access to .NET servers for any RPC client. The .NET Wrapper generation tools of the Designer take as input a Software AG IDL file, which describes the interface of the RPC, and generates C# classes that implement the methods and data types of the interface.

---

|                  |   |
|------------------|---|
| NLB              | Network Load Balancing (NLB) is a clustering technology for Windows servers that enhances the scalability and availability of mission-critical, TCP/IP-based services, such as Web, Terminal Services, virtual private networking, and streaming media servers. |
| Node Convergence | The process of converging static machine addresses into a clustered IP address on Windows.  |

## P

---

|                              |   |
|------------------------------|---|
| Participant                  | Within EntireX Broker nomenclature, a participant is an application implicitly or explicitly logged on to the Broker as a specific user. See <i>Implicit Logon</i> and <i>Explicit Logon</i> . A participant could act as client or server.   |
| PCB                          | Participant control block. In the context of persistence in EntireX, part of the UOW (unit of work).  |
| Persistence                  | Persistence is the ability of a message or message queue to live longer than the connection it is using, allowing the system to recover to a known point without loss of data. Persistence is implemented in EntireX as an optional attribute of a UOW (unit of work). If a UOW is specified as persistent, it is stored on some non-volatile medium after being committed by the sender. Its delivery to the intended receiver is assured, even if a system failure occurs. Persistent UOWs are automatically restored to their last status prior to Broker or system failure. All of the necessary internal structures are reconstructed. |
| Persistent Status            | <p>Persistent status is an attribute of a UOW (unit of work). If a UOW has persistent status, the status of the UOW is maintained in the persistent store, and is updated whenever the status changes. The persistent status remains in the persistent store after the UOW is completed, until its status lifetime has expired.</p> <p>A persistent status value represents a multiple of the UOW lifetime value. Thus if a UOW has a lifetime of 5M (whereby M stands for minutes) and a persistent status value of 4, the status of the UOW would be deleted 20M (5M*4) after the UOW was completed.</p>                                  |
| Persistent Storage Interface | The Persistent Storage Interface (PSI) is the EntireX Broker component that maintains non-volatile images of persistent UOWs (units of work). The PSI provides status information and is responsible for recovery if a system failure occurs.   |
| Persistent Store             | The persistent store is used for storing unit-of-work messages to disk. This means message and status information can be recovered  |

after a hardware or software failure to the previous commit point issued by each application component.

**Persistent Unit of Work** Persistence is an attribute of a UOW (unit of work). If a UOW is persistent, its messages are saved in the persistent store when the sender `COMMITs` the UOW and they are retained until the receiver `COMMITs` or `CANCELS` the UOW, or until its lifetime expires. If the Broker or system should fail after the UOW is committed by the sender, the UOW (and its conversation) will be restored to their last, stable status when the Broker restarts.

**PL/I Wrapper** The EntireX PL/I Wrapper provides access to RPC-based components from PL/I applications. It enables you to develop both client and server applications.

**Privacy** In the context of EntireX Security, privacy means that data packages cannot be examined by unauthorized persons.

**Proxy** In COM/DCOM terms, a proxy is a small binary loaded into the client's process space that acts as a "front-end" to the remote server's interface. The client calls the proxy just as it would the server, and the data is marshalled to the stub which in turn calls the server object.

In EntireX Broker terms, the ACI acts both as a proxy and a stub, but is commonly referred to as the "broker stub" or simply "stub" regardless of whether it is installed on the client side or the server side.

## Q

---

**Queued Direct I/O (QDIO)** Queued Direct I/O (QDIO) is a System z hardware feature offering an enhanced data transfer architecture for improving data transfer speed and efficiency for TCP/IP traffic.

## R

---

**Reliable RPC** Reliable RPC is the EntireX implementation of a reliable messaging system. It combines EntireX RPC technology and persistence, which is implemented with units of work (UOWs).

**Remote Program** The application program with which the local program converses.

**RPC** A Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details. (A



procedure call is also sometimes known as a function call or a sub-routine call.)

RPC uses the client/server model. The requesting program is a client and the service-providing program is the server. Like a regular or local procedure call, an RPC is a synchronous operation requiring the requesting program to be suspended until the results of the remote procedure are returned. However, the use of lightweight processes or threads that share the same address space allows multiple RPCs to be performed concurrently.

When program statements that use RPC are compiled into an executable program, an interface object is included in the compiled code that acts as the representative of the remote procedure code. When the program is run and the procedure call is issued, the interface object receives the request and forwards it to a client runtime program in the local computer. The client runtime program has the knowledge of how to address the remote computer and server application and sends the message across the network that requests the remote procedure. Similarly, the server includes a runtime program and interface object that interface with the remote procedure itself. Results are returned the same way.

Some examples of RPC technology are Software AG's EntireX RPC, Microsoft RPC, and DCE RPC.

Software AG's Natural RPC uses the same RPC protocol as EntireX RPC, which means they are fully compatible.

|                               |   |
|-------------------------------|---|
| RPC Runtime                   | Software AG RPC runtime is a protocol that enables RPC-based communication between clients and servers.   |
| RPC Server for AS/400         | The EntireX RPC Server for IBM® AS/400® enables you to remotely call server programs written in all ILE-based languages, for example C, COBOL, RPG.   |
| RPC Server for Batch (BS2000) | The EntireX RPC Server for BS2000 allows standard RPC clients to communicate with RPC servers on the operating system BS2000. It supports the programming languages COBOL and C.  |
| RPC Server for Batch (z/OS)   | The EntireX RPC Server for z/OS Batch allows standard RPC clients to communicate with RPC servers on the operating system z/OS running in batch mode. It supports the programming languages COBOL, PL/I and C and works together with the <i>COBOL Wrapper</i> and <i>IDL Extractor for COBOL</i> . |

|                                     |  |
|-------------------------------------|--|
| RPC Server for C                    | The EntireX RPC Server for C allows standard RPC clients to communicate with servers written in C. It works together with the C Wrapper and calls standard libraries (Windows DLLs or Linux shared objects/libraries).   |
| RPC Server for CICS (z/OS)          | The EntireX RPC Server for z/OS CICS® allows standard RPC clients to communicate with RPC servers on the operating system z/OS under CICS. It supports the programming languages COBOL and PL/I.   |
| RPC Server for CICS ECI             | The EntireX RPC Server for CICS® ECI allows standard RPC clients to communicate with CICS programs running on IBM CICS®. It supports the programming languages COBOL and PL/I and transforms the RPC requests from clients into messages to CICS ECI.  |
| RPC Server for CICS Socket Listener | The EntireX RPC Server for CICS® Socket Listener allows standard RPC clients to communicate with CICS programs running on IBM CICS®. All CICS interface types are supported: (DFHCOMMAREA, Channel Container and Large Buffer).  |
| RPC Server for .NET                 | The EntireX RPC Server for .NET allows standard RPC clients to communicate with .NET server assemblies. It works together with the .NET Wrapper.   |
| RPC Server for IBM MQ               | The EntireX RPC Server for IBM® MQ runs as an RPC server and processes RPC client calls. It is used to send messages to and receive messages from an IBM MQ queue. This means that existing EntireX wrappers can be used for communication with IBM MQ.  |
| RPC Server for IMS                  | The EntireX RPC Server for IMS allows standard RPC clients to communicate with RPC servers on the operating system z/OS running with IMS in BMP mode. It supports the programming languages COBOL, PL/I and C and can provide IMS-specific PCB pointers for access to IMS databases if needed. |
| RPC Server for IMS Connect          | The EntireX RPC Server for IMS Connect allows standard RPC clients to communicate with IMS MPP programs. It works together with the IDL Extractor for COBOL and transforms RPC requests from clients into message to IMS, using IMS Connect.   |
| RPC Server for Java                 | The EntireX RPC Server for Java allows standard RPC clients to communicate with servers written in Java. It works together with the Java Wrapper and calls Java server interface objects.  |
| RPC Server for XML/SOAP             | The EntireX RPC Server for XML/SOAP allows RPC clients to communicate with target servers via HTTP(S). The RPC Server for  |

---


XML/SOAP transforms RPC client calls into XML/SOAP calls. It works together with the XML/SOAP Wrapper.

|                |   |
|----------------|---|
| RPC Servers    | The EntireX RPC Servers (available under CICS, z/OS Batch, Linux, Windows and Java) execute servers on RPC client requests. Using EntireX RPC Wrapper technology of the Software AG Designer, you can generate these servers together with integrated wrappers (server interface objects). The Designer tool you choose to generate these servers depends on your environment, for example Java Wrapper, COBOL Wrapper, C Wrapper, XML/SOAP Wrapper, etc. |
| RPC Service    | An RPC service is a customer-written program to be provided as a service by means of a configured RPC server.   |
| RPC-ACI Bridge | The EntireX RPC-ACI Bridge allows standard RPC clients to communicate with an ACI server. The RPC-ACI Bridge transforms RPC requests from clients into ACI messages.  |

## S

---

|                               |  |
|-------------------------------|--|
| SCB                           | Sender control block. In the context of persistence in EntireX, part of the UOW (unit of work).  |
| SDL                           | The Server Definition Language (SDL) is the language in which the statements are written that define the EntireX Broker systems installed, the servers that are to be controlled by the Attach Server and the path definitions required to establish the execution environments of the servers.  |
| Server                        | Within EntireX Broker nomenclature, a server is an application issuing Broker requests to process client requests assigned to specific services.   |
| Server Mapping File for COBOL | Server mapping enables the RPC server to correctly support special COBOL syntax such as <code>REDEFINES</code> , <code>SIGN LEADING</code> and <code>OCCURS DEPENDING ON</code> clauses, <code>LEVEL-88</code> fields, etc. If one of these elements is used, the IDL Extractor for COBOL automatically extracts a server mapping file (Designer file with extension <code>.cvm</code> ) in addition to the IDL file (interface definition language). Also, the COBOL Wrapper may generate a server mapping file for RPC server generation. The server mapping is used at runtime to marshal and unmarshal the RPC data stream. A server mapping file is wrapped into the client interface objects when an RPC client is generated. Therefore, it is important the server mapping ( <code>.cvm</code> ) is available before creating |

any RPC client component, that is, the server program must be extracted or generated first.  More info

#### Server Mapping File for Natural

Server mapping enables the RPC server to correctly support special Natural syntax such as `REDEFINES`, special Natural Mapping Editor operations, etc. If one of these elements is used the IDL Extractor for Natural automatically extracts a server mapping file (Designer file with extension `.cvm`) in addition to the IDL file (interface definition language). Also the Natural Wrapper may generate a server mapping file for RPC server generation. The server mapping file is used at runtime to marshal and unmarshal the RPC data stream. A server mapping file is wrapped into the client interface objects when an RPC client is generated. Therefore, it is important the server mapping (`.cvm`) is available before creating any RPC client component, that is, the server program must be extracted or generated first.

#### Server Program

A piece of application software that can provide one or more services to client programs on request.

#### Server-side Mapping File

Server mapping files with extension `.svm` are no longer supported at design time by the Designer. You can still use them at runtime in a server-side mapping container. All special COBOL syntax and features supported by server mapping files with extension `.svm` are also covered by server mapping files with extension `.cvm`. See *When is a Server Mapping File Required?*

We recommend migrating `.svm` files to `.cvm` files. See *Migrating Server Mapping Files* under *Server Mapping Files for COBOL* in the Designer documentation. See also glossary entry for [Server Mapping File for COBOL](#).

See also glossary entry for [Server Mapping File for COBOL](#).

#### Service

A particular type of functionality provided by one or more servers. Within EntireX Broker nomenclature, a service is uniquely identified by a combination of service name, server name and server class.

#### SMA

System Maintenance Aid. Software AG product for installing and managing the configuration of a Software AG product environment.

#### Software AG Common Platform

The Software AG Common Platform is a Java Runtime environment based on the OSGi framework. It provides a standard platform on which to run Software AG products and the enterprise applications you develop around those products. The Software AG Common Platform provides common infrastructure for user authentication, event handling, and the execution of Web applications. Infrastructure components that the Software AG Common Platform provide include

---

|                                     |  |
|-------------------------------------|--|
|                                     | Software AG Security Infrastructure, Software AG Web Server based on Apache Tomcat, and Web Services Stack.  |
| Software AG IDL                     | <p>Software AG IDL (Interface Definition Language) is a language that lets a program or object (the client) written in one language communicate with another program written in another language (the server). An interface definition language works by requiring a program's interfaces to be described in an interface object or slight extension of the program that is compiled into it.</p> <p>Software AG IDL is used to define an interface between the client and the server. These definitions are contained in a Software AG IDL file. IDL compilers read IDL files and generate interface objects or descriptions from the definitions they contain.</p> |
| Software AG Runtime                 | The Software AG Runtime is an installable instance of the Software AG Common Platform that functions as a stand-alone Tomcat server and a container for Web applications. EntireX uses the Software AG Runtime to host the EntireX Listener for XML/SOAP and RPC Server for XML/SOAP.  |
| Software AG Security Infrastructure | Software AG's webMethods line of products has a common authentication infrastructure called SIN. It provides the products with security components for authentication of users, management of roles, and query of user, role, and group information. It works both on client-side applications and on server-side applications.  |
| Software AG Web Server              | The Software AG Web Server based on Apache Tomcat is one of the basic infrastructure components provided by the Software AG Common Platform. It provides HTTP/HTTPS services, a JSP engine, and a servlet container. Unlike a typical Tomcat implementation, the Software AG Web Server is OSGi-based and supports both .WAR-based and .WAB-based web applications.  |
| SPFX                                | Stub prefix. The SPFX acts as a table of contents for the other sections of the ACI request or reply.  |
| SSL                                 | See <a href="#">TLS</a> .  |
| SSX                                 | Software AG Security Extension. SSX handles authentication and provides interfaces to the user database to perform list, add, modify and delete operations.  |
| Stub                                | A Broker stub is a module delivered with EntireX to communicate with a Broker. It is platform-dependent, but application-independent. A Broker stub implements an API (application programming interface) that allows programs written in various programming lan-   |

guages (Assembler | C | COBOL | PL/I) to access the EntireX Broker. *EntireX RPC Programming* also makes use of stubs to communicate with the EntireX Broker. The preferred transport method for communication is TCP/IP. Depending on the platform, SSL/TLS or Entire Net-Work can be used instead.

For Java programmers, the Java class library EntireX Java ACI provides similar functionality and access to the EntireX Broker.

In earlier versions of EntireX, the term "stub" was also used to refer to application-dependent, Designer-generated pieces of code for issuing and receiving remote procedure calls. These objects are now referred to as *interface objects*.

## Sysplex

A Parallel Sysplex is a cluster of IBM mainframes acting together as a single system image with z/OS. Parallel Sysplex combines data sharing and parallel computing to allow a cluster of up to 32 systems to share a workload for high performance and high availability.

## T

---

### Tamino X-Tensions

Tamino X-Tensions can be described as the technology you need to develop, administer and execute XML-based application logic within the Tamino context. Tamino X-Tensions provide you with the ability to extend the functionality of the Tamino Server in cases in which the standard functions are not sufficient to satisfy all of the needs of an internet application. Tamino X-Tensions can be generated by the Software AG Designer to call an EntireX/Natural RPC Server.

### TCP/IP Communicator

The EntireX Broker TCP/IP communicator enables transparent direct TCP/IP communications between a broker stub and the EntireX Broker running under z/OS.

### Template

In EntireX RPC programming, a template file is required by the Software AG IDL Compiler and describes the source language in which the client interface object or server interface object is generated. Several templates are provided by EntireX for various programming languages. Experienced users can modify these templates or create new templates for other program languages.

### TOR

In Broker ActiveX Control, the transaction object repository (TOR) is the compound file within the file system in which transaction objects (TOs) are stored.

### TLS

Transport Layer Security (TLS), and its predecessor, Secure Sockets Layer (SSL) are cryptographic protocols. They provide communica-

tions security in computer networks. TLS and SSL use the public-and-private key encryption system from RSA, which also includes the use of digital certificates.

|                      |   |
|----------------------|---|
| Transaction Object   | Transaction Object (TOs) in Broker ActiveX Control are selections of logical methods that are stored in a transaction object repository (TOR). These logical methods contain all the connection and interface details necessary to communicate with EntireX Broker. |
| Transaction Recovery | A feature for recovering resources in the event of an unplanned outage.   |
| TRAP                 | Translation Routine / Area for Parameters (TRAP) control block fields supplied by the Broker.   |
| Tunnel Servlet       | See <a href="#">Broker HTTP(S) Agent</a> .  |

## U

|                   |  |
|-------------------|--|
| UDDI Registration | EntireX UDDI Registration is a tool with which you can register a web service with any UDDI registry for which you have an account.  |
| UOW               | <p>A unit of work (UOW) is a set of one or more messages that are processed as a single unit. The sender of a UOW adds messages to the UOW and then indicates that the UOW is complete (COMMIT). The UOW and its messages are not visible to the receiver until the sender has committed the UOW. Once the UOW is committed, the receiver can receive the messages, and can indicate when the UOW is complete (COMMIT).</p>  |
| UOW Lifetime      | <p>Each UOW has a lifetime value associated with it. This is the period of time that the UOW is allowed to exist without being completed. This time starts when the UOW is initially created and runs until the UOW is completed. A UOW is completed when it is successfully:</p> <ul style="list-style-type: none"> <li>■ cancelled or backed out by its sender, or</li> <li>■ cancelled or committed by its receiver.</li> </ul> <p>If the UOW is in <code>ACCEPTED</code> status when this lifetime expires, the UOW is placed into a <code>TIMEOUT</code> status. Lifetime timeouts will not occur when the UOW is in either <code>RECEIVED</code> or <code>DELIVERED</code> status.</p> <p>A special “pseudo-clock” is maintained for UOW lifetimes. This clock is implemented in such a way that it only runs when the Broker is active. This prevents a UOW lifetime from expiring while the Broker is down or otherwise unavailable.</p> |

**UOW Status Query** Each unit of work (UOW) has a unique identifier that allows you to check the status of the UOW. This is particularly useful during recovery from a system failure. The status of persistent and non-persistent UOWs is maintained by the Broker for a period of time which is determined by user-defined attributes.

**USRSEC** USRSEC is the name of the security exit which is called if `SECURITY=YES` is specified in the Broker attribute file. A security component called USRSEC is provided by EntireX Security in binary form. EntireX Security provides full protection of applications using EntireX Broker. Functionality includes authentication for distributed application components as well as authorization to start and execute services. All user and resource definitions are managed through RACF and CA ACF2. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See *SSL/TLS, HTTP(S), and Certificates with EntireX* in the platform-independent Administration documentation.

**UWCB** UOW (unit of work) control block. In the context of persistence in EntireX, the UWCB represents the UOW.

## V

---

**VIPA** Virtual IP addressing is an IP address assigned to multiple static IP addresses to make them accessible as one address for the purpose of improving availability or load balancing.

**Visual Studio Wizard for .NET Wrapper** The Visual Studio Wizard for .NET Wrapper is an extension for Microsoft Visual Studio, which makes the functionality of the EntireX .NET Wrapper available to Microsoft Visual Studio.

## W

---

**webMethods EntireX** webMethods EntireX is a secure, robust, high-performance communication infrastructure that uniquely combines message queuing capabilities with built-in support for synchronous request/reply and conversational communication. It manages interactions for tightly coupled XML-enabled and non-XML enabled systems in time-critical user applications. EntireX's powerful and easy-to-use wrapping technology and flexible programming interfaces turn existing application functions into business services or web services. These features allow an organization to define an integration architecture for enhancing existing applications, linking up legacy applications with new standard software or adding new custom applications to the mix.



---

|   |   |
|---|---|
| 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 services. EntireX supports bidirectional integration with Natural, COBOL and PL/I on z/OS, BS2000, z/VSE and other platforms. This includes zero footprint integration with CICS, IMS and IBM i (AS/400).  |
| webMethods EntireX Package                        | A webMethods EntireX package is an Integration Server (IS) package written to the Adapter runtime specification that allows EntireX assets to be tightly integrated with IS Flow Services. The package features IS nodes and automatic generation of communication services and application services that are associated with making program calls to mainframe transactions via EntireX RPC.                             |
| webMethods Integration Server                     | The webMethods Integration Server is a Java-based, multiplatform enterprise integration engine supporting the execution of services to perform integration logic such as data mapping and communication with other systems.   |
| webMethods Integration Server Wrapper             | The webMethods Integration Server Wrapper generates Integration Server adapter services and listeners from a Software AG IDL file within an Integration Server connection definition.   |
| Web Services                                      | Web services are programmable, distributed application components accessible on the Web using solely standard internet protocols. In contrast to the current “document Web”, which specializes in human interaction, web services are designed to be accessed by programs to form a new application architecture, the “application Web”.  |
| Web Services Wrapper                              | The EntireX Web Services Wrapper is a wizard that generates and optionally deploys web services (Designer file with extension .aar) to offer an RPC server - for example a COBOL or Natural RPC server - as a web service. The generated XML/SOAP mapping file (Designer file with extension .xmm) can also be used to enable RPC clients - for example a COBOL or Natural client - consuming (or calling) a web service. |
| Web Services Wrapper for Natural                  | The EntireX Web Services Wrapper for Natural allows you to develop Web Services that access Natural server components, using EntireX/Natural RPC.   |
| WLM   | IBM Workload Manager is a Sysplex based load balancer offering workload filtering.  |
| Workbench   | See <a href="#">Designer</a> .  |

**Wrapper** In EntireX terms, a wrapper is a tool contained in the Designer to generate interface objects based on a Software AG IDL file and additional wrapping properties.

## X

---

**XCOM Communicator** The XCOM Communicator consists of the Adabas modules ADAMPM, ADALNK, and ADASVC. It can be used for intra-mainframe communications, that is, communications between client and server applications via the EntireX Broker when all components are located on the z/OS mainframe.

**XMLAdapter** A Java class generated by the XML Mapping Editor and used by the XML/SOAP Runtime to map between client XML documents and EntireX/Natural RPC Server parameters (deprecated). Not to be confused with EntireX XML Adapters.

**XML Mapping Editor** The EntireX XML Mapping Editor allows you to map XML document structures to IDL libraries, programs and parameters. The mappings can be defined for the request and response to the server application, or from the server to the client. The input for the XML Mapping Editor can be a Software AG IDL file and/or an IDL-XML mapping file (perhaps produced by a previous XML Mapping Editor session or by importing a WSDL file, XML Document or XML Schema). The output is an IDL-XML mapping file, other XML structure definitions (such as sample XML files), and perhaps a created or changed IDL file.

**XML Mapping File** The XML Mapping File is an XML file (file extension .xmm) generated by the EntireX XML Mapping Editor and used by the EntireX XML/SOAP Runtime, to map between XML documents and EntireX/Natural RPC parameters.

**XML RPC** XML RPC is a simple, language and platform neutral, RPC mechanism using XML over HTTP. It is a specification and a set of implementations that allow software running on disparate operating systems, running in different environments to make procedure calls over the internet. HTTP provides the transport and XML provides the encoding. XML RPC is designed for simplicity but allows complex data structures to be transmitted, processed and returned.

**XML Schema** XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents in more detail.

---

|                       |  |
|-----------------------|--|
| XML Tester            | Using the EntireX XML Tester you can send an XML document to the EntireX Listener for XML/SOAP.  |
| XML to IDL Generators | XML to IDL Generators generate Software AG IDL files and related mapping files from DTD files, XML Schema files, XML files and WSDL files.   |
| XML/SOAP Listener     | See <a href="#">Listener for XML/SOAP</a> .  |
| XML/SOAP RPC Server   | See <a href="#">RPC Server for XML/SOAP</a> .  |
| XML/SOAP Runtime      | The EntireX XML/SOAP Runtime enables XML-based clients to communicate with an EntireX/Natural RPC Server; and it enables EntireX RPC clients to communicate with applications that have an XML or SOAP interface (via XML/SOAP Runtime). |
| XML/SOAP Wrapper      | The EntireX XML/SOAP Wrapper enables XML-based communication to EntireX/Natural RPC servers and communication from EntireX/Natural RPC clients to XML-based servers.   |
| XMM                   | XML Mapping File, file extension .xmm, replaces the deprecated XML Adapter.  |

## Z

---

|                       |  |
|-----------------------|--|
| Zero Client Footprint | In the context of EntireX Security, the fact that the application requires no space on the client machine. |
|-----------------------|--|

## Numbers

---

|        |  |
|--------|--|
| 24x7   | A business definition for highly available applications representing 24 hours, 7 days a week availability.<br><br>See also <i>High Availability in EntireX</i> .         |
| 99.999 | A business definition for highly available applications representing a percentage very close to 100% availability.<br><br>See also <i>High Availability in EntireX</i> . |

