webMethods EntireX

Introduction to webMethods EntireX

Version 9.7

October 2014

WEBMETHODS

This document applies to webMethods EntireX Version 9.7.

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

Document ID: EXX-CONCEPTS-97-20160805

Table of Contents

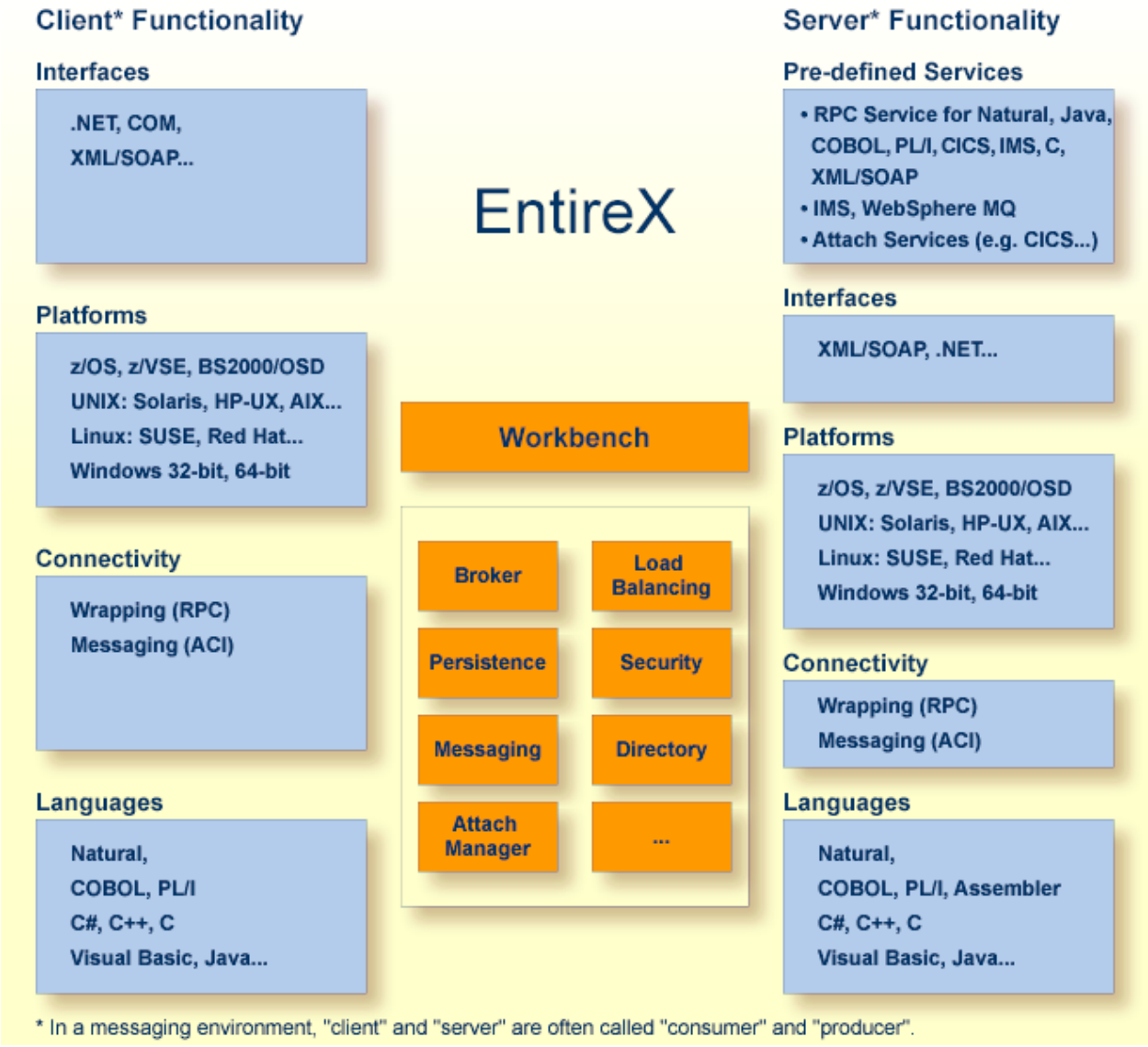
1 Introduction to webMethods EntireX

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.

See also *Common Integration Scenarios*, for example *Calling COBOL from Integration Server*.

EntireX Components

The graphic below provides an overview of platforms and functionality supported by EntireX.

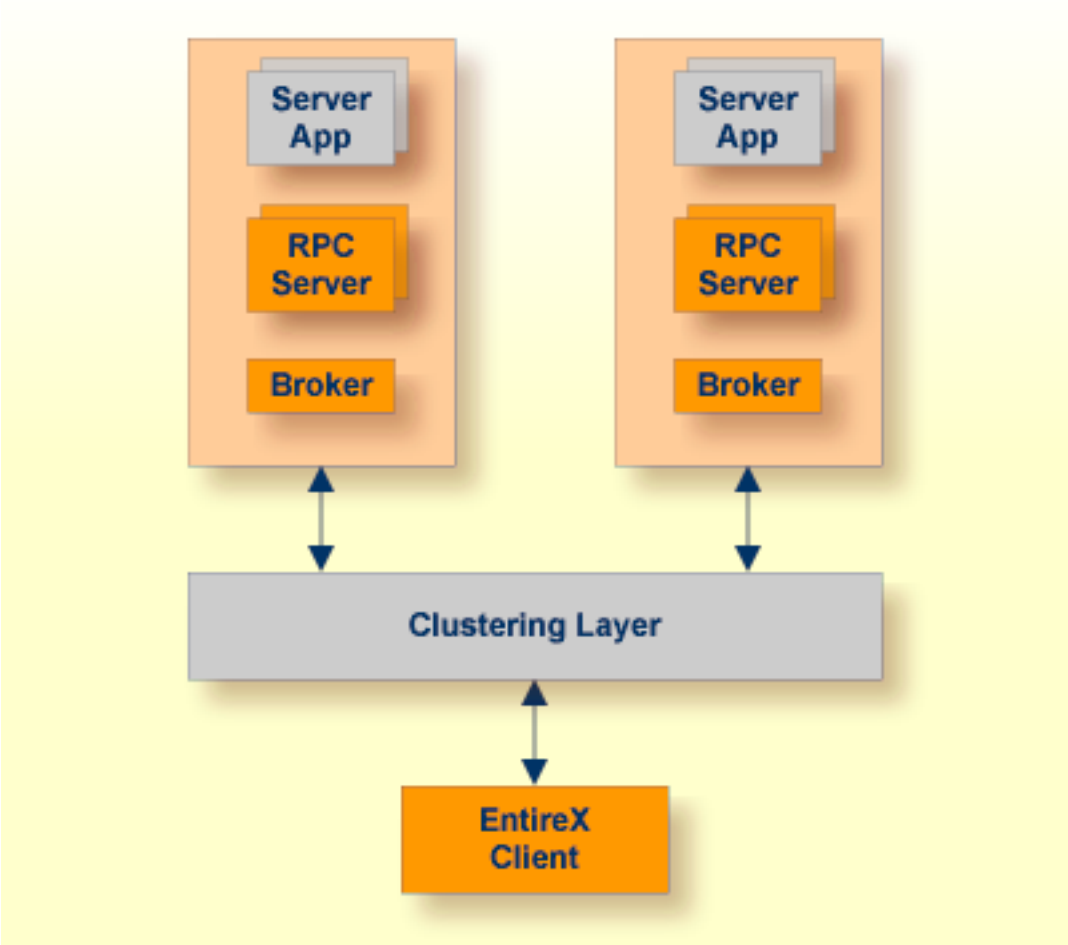


High Availability

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.

Many of the world's largest organizations including financial institutions, manufacturing, transportation, and communication companies along with large government agencies, rely on the availability and reliability of applications to deliver their most important business transactions and data. These large-scale information systems consist of several hardware and software components, each of which performs a particular function and is a critical-path to successful daily operations. If any of these components fail however, the outcome could vary drastically - from a single user experiencing a slow-down in their order response time, to thousands of retail bank customers not being able to access any of their cash assets. Creating a highly available system topology removes any single points of failure from a large system, enabling another redundant component to effectively take over the workload of the failed component. Improving availability ultimately leads to a reduction in downtime, improved performance, and a better user experience.

Another IT concern is how to apply maintenance and upgrades to these important systems without affecting users. In a highly available world, a system in need of maintenance can be taken out of the workload pool and updated while the rest of the system continues to process service requests.



See *High Availability in EntireX*.

EntireX Broker

In general, EntireX can be compared to a hardware bus, adding new functions or applications to the system by means of extender cards. Every application that is "plugged into" EntireX represents a functional expansion, with the main purpose of the bus system being to provide a means of communication between the individual components. This bus architecture provides organizations with the foundation for state-of-the-art integrated electronic business applications. The EntireX Broker, a high-performance message server, is the centerpiece of the EntireX Bus System.

The EntireX Broker offers support for multiple communication and component models and provides the backbone for integration between an organization's applications and the internet.

Inside the EntireX Broker

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

With the EntireX Broker, companies benefit from the flexibility, ease of use and high-performance characteristics that make it unique among Message Oriented Middleware (MOM)-based products. Other features of the EntireX Broker include:

- Ensured delivery of messages and publications
- Transparency of physical location of servers to clients, and of publishers to subscribers
- Attach service for dynamic replication and start of servers
- Workload balancing between replicated servers
- Optionally keep data in persistent storage
- Native TCP/IP support

Flexibility: Support for Different Communication Alternatives

In creating an electronic business IT infrastructure, flexibility is a key requirement for the middleware foundation. One area where flexibility is a must is in the way applications communicate with each other. Because different applications may require different methods of communication, EntireX provides support for a variety of different models:

- **Synchronous**
  Synchronous communication means the sender and receiver are active simultaneously. The requesting application sends its message and then stops processing while it waits for a reply (blocking).
- **Asynchronous**
  Asynchronous is non-blocking communication where applications can be either concurrently or non-concurrently executing.
- **Conversational**
  Conversational means participating programs interact with each other in the form of a dialog, with each program responding in turn to information received from the other program.
- **Request/Reply**
  Request/Reply is based on a single request and a single return, rather than a conversation. The sender is usually blocked.

- **Messaging-and-Queuing**
  Messaging-and-Queuing is based on intermediate message storage. The sending application sends the message to the messaging middleware, which places it in a queue. The sender is generally unblocked.
- **Publish and Subscribe**
  Publish and Subscribe where the publisher sends data to an unspecified number of subscribers without getting a reply.

The EntireX Broker approach is unique among message queuing systems in that applications never see the actual queues themselves, instead they only deal with the logical names of senders and receivers. This provides an additional level of transparency for applications being integrated. In addition, the Broker can reside on the network, on the sender's systems, the server side or on any other machine in the network. The roles of clients and servers are not fixed but can change dynamically within a conversation.

High Reliability

The EntireX Broker provides optional message and publication persistence by storing message queues and publication data to a non-volatile medium. Persistence ensures that messages and publications reach their destination even in the event of a system failure. EntireX can also group logically related client and server messages into units of work that are committed to the EntireX Broker for further transmission when complete. In case of failure on the server side, the receiving program can back out the whole unit of work, making it available for processing later or by another server.

High Performance and Resource Management

Server replication, automatic load balancing between replicated servers and automatic, dynamic adjustment of the system environment ensures that client requests are answered quickly and resources are put to efficient use. The EntireX Attach Service dynamically replicates and starts servers, based on Broker requests for services. In a CICS environment, this includes transactions that are offered to the network as a Broker service. Additionally, the Attach Service can start clients, based on a send operation by a server. The way the Attach Service manages replicating and starting servers also leads to an efficient balancing of the workload within an EntireX environment: the Attach Service registers as the Attach Server with the Broker and can receive attach requests to start additional servers to avoid bottleneck situations. Workload balancing is achieved using high and low water marks specified in the server definitions. The Attach Service keeps the number of server replicas registered with the Broker between these levels and starts new replicas when their number falls below the set minimum.

The Basics: Communicating with the Broker

Before discussing the types of applications which can be plugged into the EntireX bus, it is best to start with a short description of the basic communication mechanisms used by EntireX.

ACI (Advanced Communication Interface)

Communication with the Broker is handled through Advanced Communication Interface (ACI) programs. The ACI has `SEND` and `RECEIVE` functions that provide the Broker-based communication basis for point-to-point transmission between clients and servers on the EntireX bus. Similarly, ACI offers `SEND-PUBLICATION` and `RECEIVE-PUBLICATION` functions that provide the basis for publish-and-subscribe communication. On the ACI level, any communication model is supported.

See also *EntireX Broker ACI Programming*.

EntireX RPC

EntireX's Remote Procedure Call (RPC) represents a higher level of ACI abstraction. It shields the application developer from the technical details of the ACI, such as addressing the (remote) server, establishing communication, converting data formats, etc. The communication model used with RPC technology is synchronous request/reply. The developer can use familiar call syntax (C, Visual Basic, etc.) to generate the appropriate RPC stub automatically. The EntireX RPC is the underlying communication method used by Software AG's wrapper technology, described in further detail below.

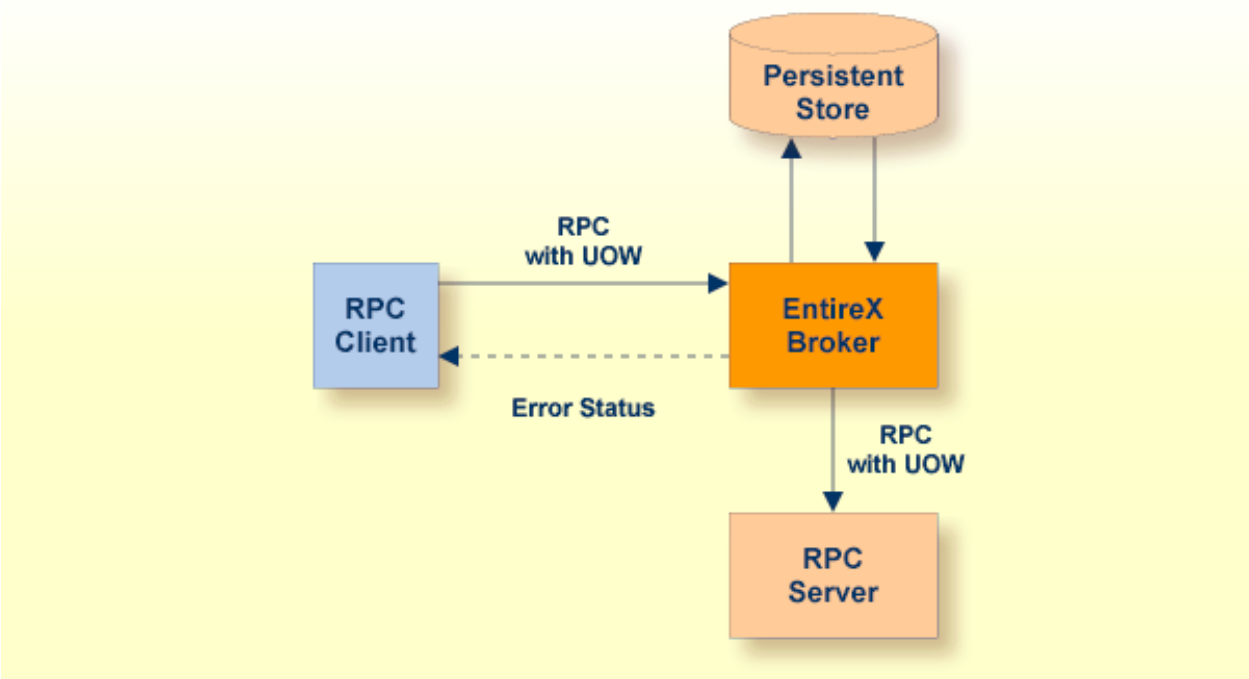
See also *RPC-based Components*.

Reliable RPC

In the architecture of modern e-business applications (such as SOA), loosely coupled systems are becoming more and more important. Reliable messaging is one important technology for this type of system.

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).

- Reliable RPC allows asynchronous calls (“fire and forget”)
- Reliable RPC is supported by most EntireX wrappers
- Reliable RPC messages are stored in the Broker's persistent store until a server is available
- Reliable RPC clients are able to request the status of the messages they have sent



See *Reliable RPC*.

JMS

Java Message Service (JMS) is a standard API. The EntireX Broker is a message service provider that supports JMS with both point-to-point and publish and subscribe messaging. This enables JMS applications to profit from the reliability and performance of the EntireX Broker.

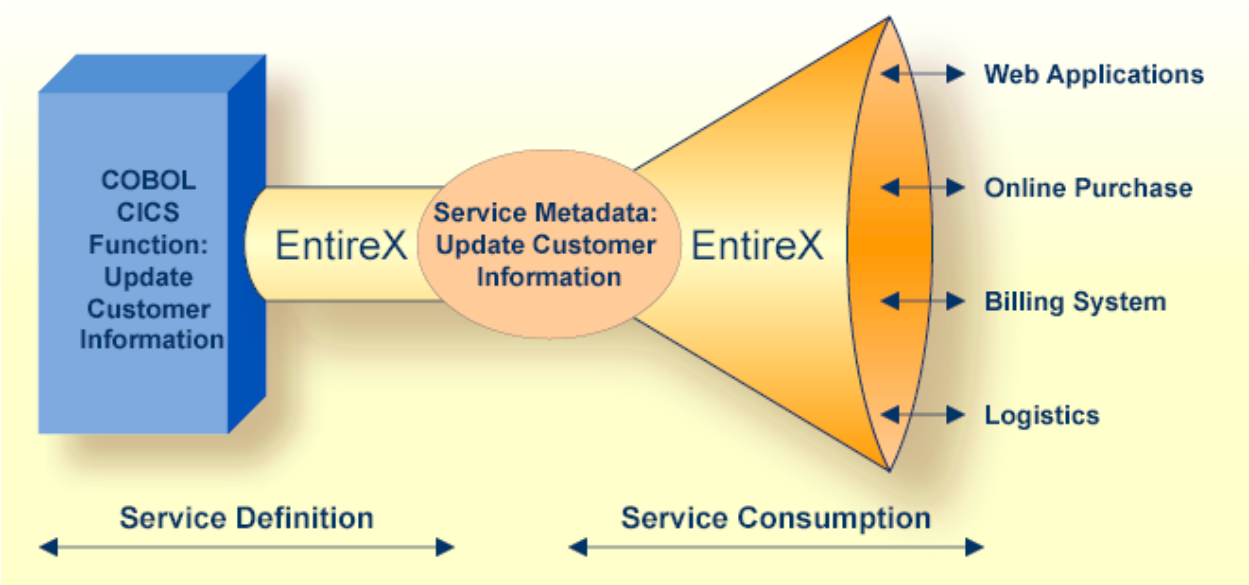
Wrapping and ACI Functionality

EntireX Enables the Creation of Business Services

EntireX “wraps” existing applications so that they can be accessed as “services”. The interfaces to these services can be XML/SOAP, DCOM, Java etc. Whereas most new development is focused on providing a Web services interface, EntireX allows all these interface standards to work seamlessly together to meet customers' technology needs - i.e. manage the mix of technology adoption that exists within the enterprise. The developer starts, for example, with an existing business function that is likely needed in various applications such as a function for updating customer information. This function could be implemented as a COBOL CICS transaction, a Natural subprogram, a UNIX C program, etc.

Using the EntireX Workbench, this function can easily be turned into a reusable service by generating the service metadata (service name and interface description) and optionally publish this metadata.

Once the service has been defined, it can be used / consumed from any application. The function to update customer information could be provided in an internal Web application, an Online Purchase application, the billing system, etc. These applications can now, thanks to EntireX, access the business function transparently as if it was developed as part of the application using it. The actual implementation details of the function are hidden by EntireX.



Java Applications

EntireX provides special Java class libraries to link applications written in Java to the EntireX bus.

- **Java Wrapper**
  The Java Wrapper provides access to RPC-based components from Java applications. Java Wrapper enables users to develop both client and server applications written in Java. In addition, Java applets can also be used as an RPC client.

  See also *EntireX Java Wrapper*.

- **EntireX Java ACI**
  EntireX Java ACI is a Java class library that provides Java programmers with access to the Broker ACI. It enables the creation of 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 Java ACI also contains the framework necessary for Java RPC requests.

  See also *EntireX Java ACI* in the Developer's Kit documentation.

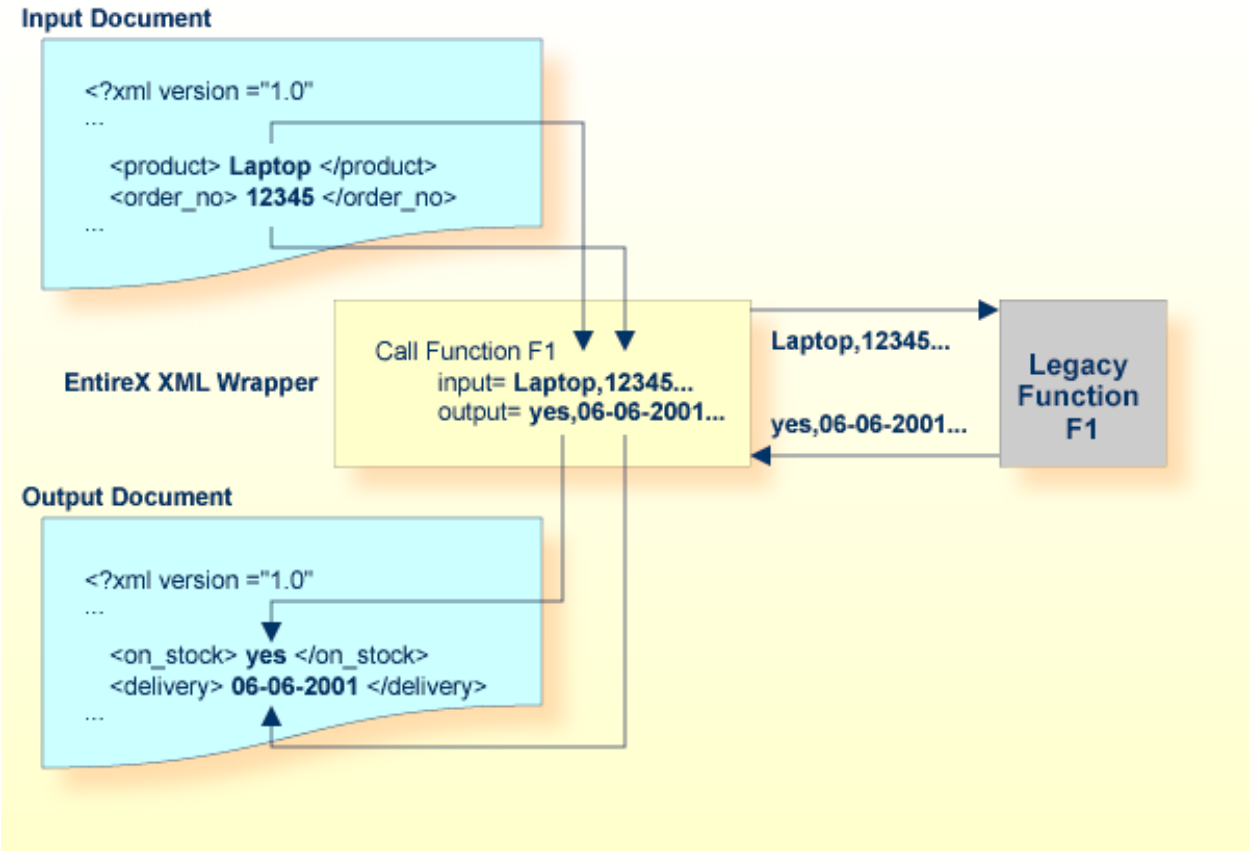
XML Applications

The following XML applications are provided:

- XML/SOAP Wrapper
- XML/SOAP RPC Server

XML/SOAP Wrapper

An extension to the Java Wrapper provides XML communication to existing non-XML servers. With no additional programming effort, data from existing applications (such as CICS transactions) can be provided in XML format to electronic business applications. The EntireX XML/SOAP Wrapper receives an XML document from a Web server or Java application, translates it into a remote procedure call (RPC) on an existing application, creates an XML document from the result of this call, and sends it back.



The XML/SOAP Wrapper enables seamless integration with any technology based on XML over HTTP, including SOAP. In terms of Software AG technology this means

- Tamino can with Tamino X-Tensions directly interact with an XML/SOAP Wrapper provided by EntireX to provide legacy integration support which enables a single XML view of XML and non-XML data within the organization.

See also *Introduction to the XML/SOAP Wrapper*.

XML/SOAP RPC Server

As Web services architectures become increasingly common, Software AG has extended its XML/SOAP Wrapper technology with the XML/SOAP RPC Server. This server enables applications to consume Web services simply and easily by encapsulating the implementation specifics. Previously EntireX XML/SOAP Wrapper technology was focused on exposing and wrapping existing applications as Web services, but now this technology is fully bidirectional. With the EntireX XML/SOAP RPC Server, any application integrated with EntireX can not only be exposed as a Web service (using the familiar XML/SOAP Wrapper), but it can also use existing Web services in the same way. This extends the service-oriented offering of EntireX by providing the ability to use any available Web service to extend the functionality of any EntireX service if desired.

EntireX can now be used as a simple mediator between any service/application and any available Web service. This technology enables even faster reaction to new business needs, because new functionality can easily be added by calling an external Web service.

Additionally, the XML/SOAP RPC Server enhances EntireX as an interface to the new EntireX XML Adapters as the adapters are Web services.

See also *Introduction to the XML/SOAP Wrapper*.

Natural Applications

The Advanced Communication Interface (ACI) for Natural allows the Natural developer to interface directly with EntireX Broker using an API suited to their needs. In addition, EntireX RPC provides developers with a connectivity and wrapping capability which allows them to "object wrap" legacy code in order to make it available as components in a wider enterprise object environment. EntireX RPC enables client programs to access Natural RPC Server subprograms and 3GL RPC server applications on mainframe, UNIX and Windows platforms. EntireX RPC is fully compatible with the Natural RPC facility.

See also *ACI Programming*.

CICS Applications

Again, the Advanced Communication Interface for 3GLs provides the 3GL developer with an interface to the EntireX Broker. Easing the integration pains of legacy applications, the EntireX z/OS CICS® RPC Server provides calling of existing CICS transactions from client programs on other platforms using EntireX RPCs.

A client can be any application for which an ACI adapter is available.

WebSphere MQ Applications

The WebSphere MQ RPC Server is EntireX's interface to IBM's message-oriented middleware WebSphere MQ. This interface allows an application developer to write RPC client applications that can send or read messages to/from a local or remote MQ queue.

See also *EntireX WebSphere MQ RPC Server*.

Applications Supporting DCOM

DCOM / ActiveX is supported by most development tools on Windows and is what application developers in the Windows world are familiar with. EntireX provides the following DCOM adapters:

- **ActiveX Control**
  The ActiveX Control addresses developers who use DCOM-enabled application development tools on the PC as ActiveX containers (for example, Visual Basic, PowerBuilder, or scripting tools on the Web server.) EntireX ActiveX Control enables integration of the desktop and the Web with mainframe or UNIX-based server applications (“back ends”).

  See also *Broker ActiveX Control* in the Developer's Kit documentation.

- **DCOM Wrapper**
  The EntireX DCOM Wrapper is a PC-based service program (a wizard) which uses the EntireX ACI and RPC technology to generate DCOM-enabled components automatically. The wrapping makes it possible to treat existing applications as ActiveX components. The DCOM Wrapper provides access to server applications on server platforms such as z/OS, UNIX or Windows and many others. The DCOM Wrapper uses an Interface Definition Language (IDL) file that describes the RPC interface and generates a Wrapper object on the basis of the interface description.

  See also *EntireX DCOM Wrapper*.

Enterprise Security

EntireX Security

EntireX Security allows seamless joining of authentication on the desktop with existing authorization rules on the server (for example, RACF, ACF2 under z/OS) without introducing any new APIs. EntireX Security provides protection for applications utilizing the Broker with functionality that includes authentication for distributed application components, authorization to start and execute as well as encryption of messages. All user and resource definitions are managed through RACF, CA Top Secret and CA ACF2. In addition, customer-unique solutions are supported through exits for customized solutions.

The goal of EntireX Security is to allow an organization to control the use of all applications, including distributed components, from a central point. The result is that an organization can implement full, flexible control with a “one user = one definition” approach, enhancing previous investments in host-based security systems and infrastructures.

See also *Overview of EntireX Security* in the EntireX Security documentation.

For non-mainframe environments, the authentication of EntireX users is performed against the native UNIX or Windows security system.

Secure Sockets Layer (SSL) and Transport Layer Security (TLS)

Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), are program layers for managing the security of message transmissions in a network. The idea is to contain the programming required to keep messages confidential in a program layer between an application (such as your Web browser or HTTP) and the internet's TCP/IP layers. The term sockets refers to the method of passing data back and forth between a client and a server program in a network or between program layers in the same computer. SSL and TLS use the public-and-private key encryption system from RSA, which also includes the use of a digital certificate.

See *SSL or TLS and Certificates with EntireX* for information on how to use SSL and TLS as transport layers for EntireX.

Authorization Rules

Authorization rules can be used to perform an access check for a particular broker instance against an (authenticated) user ID and list of services. Authorization rules can be administered with System Management Hub (see below) and stored within a repository.

See *Administering Authorization Rules using System Management Hub* in the UNIX and Windows administration documentation.

Application Management

EntireX provides a powerful management component, the System Management Hub (SMH), that solves the difficult problems of managing and monitoring applications in a distributed and heterogeneous enterprise environment. The System Management Hub is Software AG's cross-product and cross-platform product management framework. The basic concepts of this product and System Management Hub features common to all Software AG products are described in the separate System Management Hub documentation.

Web Services and SOAP

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”. EntireX supports the creation of Web services from existing EntireX RPC servers.

SOAP (originally Simple Object Access Protocol) (SOAP 1.1) is a messaging and RPC protocol designed for integrating heterogeneous Web services in the internet. It defines a message format in the Extensible Markup Language (XML) that can be transported over existing internet transport

protocols (HTTP, SMTP, FTP or others). By using standard XML, SOAP messages are self-describing, that is, they carry enough information for a receiver to decompose and process the message in a standard way. By using standard internet protocols, SOAP seamlessly fits into existing internet infrastructure (for example, routers, firewalls, Web servers).

General Web Services Architecture

In general, the architecture of the EntireX SOAP solution is Web-server-based. For the XML/SOAP Wrapper a Java interface can also be used. A Web server module (SOAP handler) receives the SOAP request, validates it and transforms the request into an RPC request (EntireX RPC or DCOM). The result of the RPC request in turn is transformed into a SOAP response message and sent back to the client. If an error occurs, a SOAP fault message is sent back to the client.

Web Service Enabling EntireX Servers and Clients

The EntireX XML/SOAP Wrapper enables XML-based communication to EntireX or Natural RPC servers. From an incoming XML document, the XML/SOAP Wrapper extracts the information for and assembles a call to an EntireX/Natural RPC Server. The result is converted into an outgoing XML document. The XML/SOAP Wrapper also supports a SOAP mapping. This includes proper handling of the SOAP envelope, parsing and validating SOAP documents in accordance with the SOAP encoding rules, serializing RPC replies into SOAP responses and assembling correct SOAP fault messages on errors. In addition, WSDL descriptions are generated automatically from the XML mapping information for the EntireX RPC Servers.

Using the XML/SOAP RPC Server, EntireX clients can be given access to any existing Web service.

EntireX CentraSite Integration

Web services created with the EntireX Web Services Wrapper can be stored in the CentraSite repository and registered as services in the CentraSite registry. Dependencies between the different artifacts that constitute an EntireX Web service (IDL, WSDL and mapping files) are also registered in CentraSite to allow for impact analysis.