

Functionality of EntireX Broker

This chapter gives an overview of the major value-added services provided by EntireX Broker. These services relieve the administrator or application builder of the task of providing the desired functionality.

This chapter covers the following topics:

- Application Bindings (Stubs)
 - Attach Services
 - Codepage Conversion
 - Command and Information Services
 - Accounting
 - Data Compression
 - Persistent Store
 - Units of Work
 - Security
-

Application Bindings (Stubs)

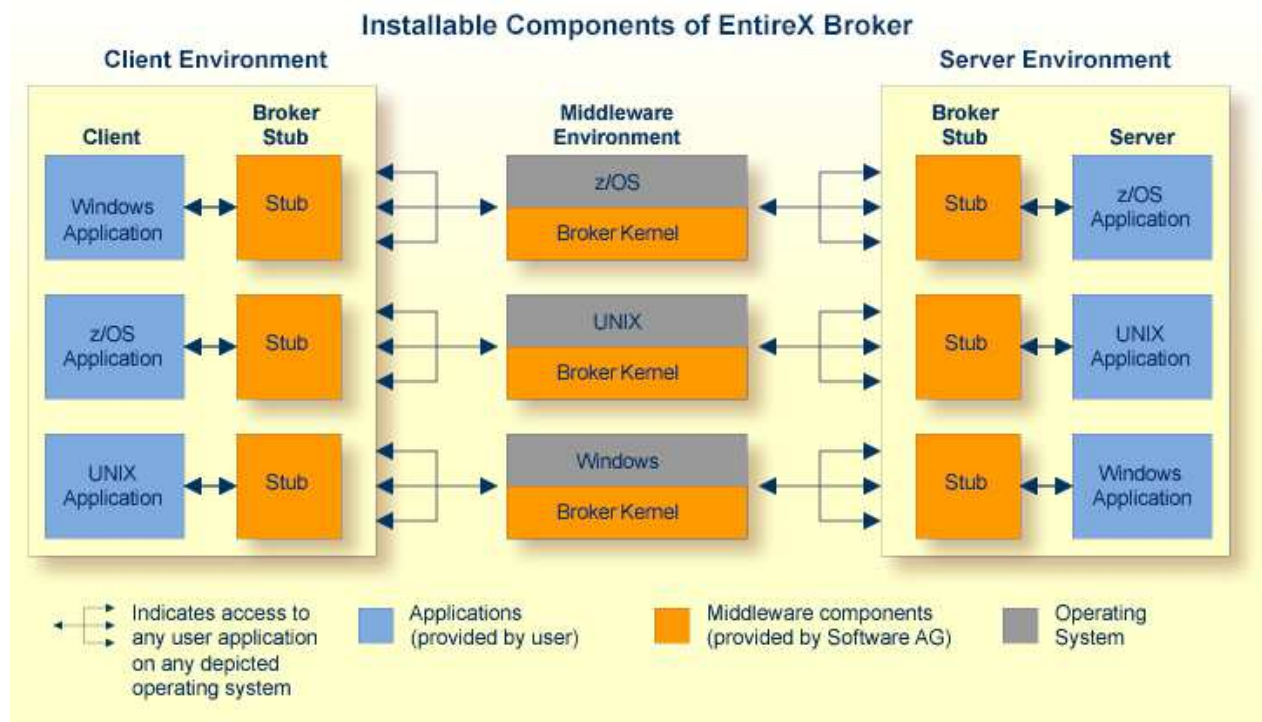
Application bindings allow applications developed in different programming languages and executing on various different platforms to be enabled by using EntireX Broker, see *Architecture of Broker Stub*. Specifically, almost all 3GL, Java and Natural programs are easily enabled using EntireX Broker. These bindings are available on all major mainframe, UNIX and Windows platforms. In addition, the SDK provided by EntireX allows different programming interfaces to be utilized, including COM, JMS, RPC and .NET, in addition to EntireX Broker's native programming interface, the Broker ACI.

The application binding - and SDK component, where appropriate - is the glue between the application and the EntireX Broker kernel (see *Architecture of Broker Kernel*, allowing your application to leverage all the functionality of EntireX regardless of

- programming language
- operating system
- hardware platform
- transport mechanism and
- choice of programming interfaces.

This binding capability enables various different application components to be integrated in a loosely coupled manner.

These are the locations where EntireX Broker stubs can be installed:



Attach Services

This topic does not apply to the publish-and-subscribe communication model.

EntireX Broker provides a choice of mechanisms which enable application components to be started automatically when required.

Example: A client application requires some processing from a server application component. The range of attach services includes starting IMS TM and CICS transactions on the mainframe, and batch programs/processes on mainframe, UNIX and Windows.

Codepage Conversion

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. Codepage conversion within the EntireX Broker facilitates the internationalization of messages: the incoming and outgoing data is converted to the desired codepage of the platform in use.

Command and Information Services

EntireX Broker includes a set of monitoring and control functions that enable you to monitor system resource utilization and view the current activities of the clients, servers, publishers and subscribers on the system. These services are available through a Web-based interface, in addition to a command-line tool. An interface exists to allow program access to these facilities.

Accounting

This topic does not apply to the publish-and-subscribe communication model.

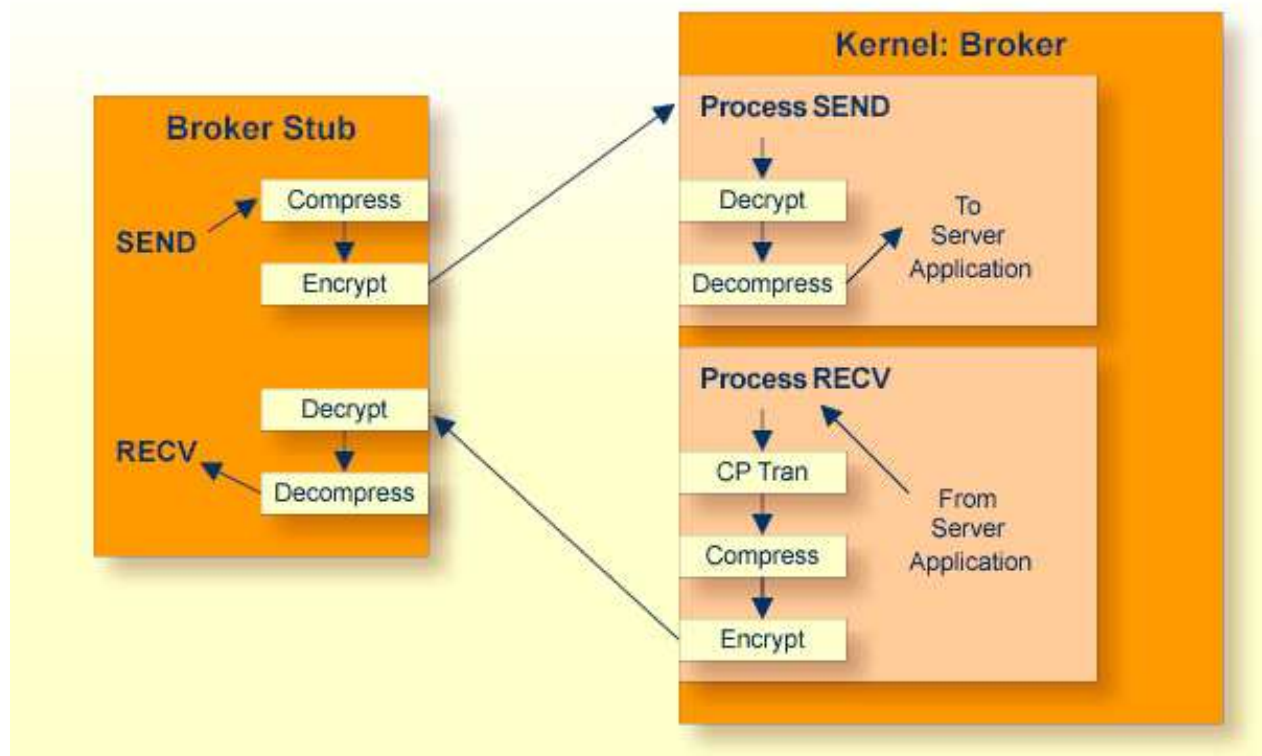
EntireX Broker provides accounting information based upon the flow of message sequences (or conversations). On z/OS, this information is written to standard accounting (SMF) records; on other platforms it is written to a file. The information can be used for:

- application chargeback: apportioning EntireX resource consumption on the conversation and/or the application level;
- performance measurement: analyzing application throughput (bytes, messages, etc.) to determine overall performance;
- trend analysis: using data to determine periods of heavy and/or light resource and/or application usage.

Data Compression

EntireX allows compression of messages passed between application components so as to consume less network bandwidth. This is done independently of transport mechanism by compressing the message in the application binding before it is transmitted to the EntireX Broker kernel. The Broker kernel decompresses the message to enable security and data conversion to be applied.

The following graphic illustrates the sequencing of data compression within the stub and Broker kernel:



Persistent Store

The persistent store stores units of work for client and server applications and also stores publication/subscription data for publish-and-subscribe applications.

- **Client and Server**

Persistent message delivery ensures that messages sent between client and server (or server and client) application components can reach their target even in the event of application or system failures. The user application programs units of work to achieve persistent messaging. EntireX Broker provides persistent message delivery by grouping messages into units of work (UOWs) that are committed in one atomic operation by the sender. See also *Units of Work*.

- **Publish and Subscribe**

Two classes of information (subscription records and the publication itself) are provided to ensure that durable subscription status is preserved and that message content remains persistent during system failure. The publish-and-subscribe specific verbs `SEND_PUBLICATION` and `RECEIVE_PUBLICATION` provide persistent messaging of publications, which relieves the user of programming units of work.

Persistence is implemented centrally within the EntireX Broker kernel. Therefore, the consistency of all the stored messages is guaranteed independently of the different application components and platforms from which the messages are derived.

Persistent Store Types

A persistent store driver is an executable, or a load module, which implements access to the physical persistent store. EntireX Broker allows the choice of three persistent store repositories: Adabas (DBMS), Data In Virtual (DIV) for z/OS, and native file system. The following table gives an overview of the persistent store options:

Persistent Store Type	Description	Operating System	Notes
Adabas	Uses Adabas database.	UNIX, Windows, z/OS, z/VSE	Adabas, Software AG's ADAptable dataBASE, is a high-performance, multithreaded, database management system.
DIV	Uses IBM Data In Virtual facility on z/OS.	z/OS	This persistent store option is implemented as a VSAM linear data set.
CTREE	c-tree© is an embedded local database that can be used as your persistent store.	UNIX and Windows	c-tree© is the fast and reliable embedded database of FairCom Corporation®.

Units of Work

This topic does not apply to the publish-and-subscribe communication model.

Units of work inform the sender of messages about their past and current status. Specifically, UOWs are used to:

- commit the sending of messages;
- acknowledge the receipt of messages;
- track the progress of sent messages at any point in time.

Units of work are also the vehicle for achieving persistent messaging, although UOWs can be used without persistence.

See also *Using Units of Work*.

Security

EntireX Security enables distributed application components running with Broker to be executed securely. EntireX Security is located centrally in the kernel of EntireX Broker giving it an overview of all messages sent between application components and therefore providing complete control over the authentication and authorization of each component.

Security checks are performed using a choice of security repositories, including:

- RACF
- CA ACF2
- CA Top Secret
- UNIX and Windows security systems

The security repository chosen depends on the location of the Broker kernel. Encryption of message data - by means of a generic RC4-compatible algorithm or SSL - is also available to protect sensitive information flowing between different application components. Since EntireX was designed to operate together with a security system, there is no additional application programming necessary.

This diagram depicts the location of the security components of the kernel and stubs of EntireX Broker:

