



BEA WebLogic Collaborate

Glossary

BEA WebLogic Collaborate 1.0.1
Document Edition 1.0.1
March 2001

Copyright5

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

All other product names may be trademarks of the respective companies with which they are associated.

BEA WebLogic Collaborate Glossary

Document Edition	Date	Software Version
1.0.1	March 2001	1.0.1

Glossary

This glossary defines terms that are used in BEA WebLogic Collaborate™ and BEA WebLogic Process Integrator™ documentation. Cross-referenced terms are italicized. For BEA WebLogic Server™ terminology, see [SEE](http://edocs.bea.com/wls/docs60/glossary/index.html) <http://edocs.bea.com/wls/docs60/glossary/index.html>

access control lists (ACLs)

Data structures used to authorize or deny access to *BEA WebLogic Collaborate* resources by *principals*. Human users can be authorized to access Java Server Pages (*JSPs*) for administrative and configuration purposes, to access the administration consoles, and to access the JDBC connection pool. *Trading partners* are authorized to access the *transport servlet* and the WebLogic Server JDBC connection pool. The *c-hub* server is authorized to access the *transport servlet*. WebLogic Collaborate supports four custom ACLs: `hubconfig` (configure and monitor the *c-hub*); `hubmonitor` (only monitor *c-hub*); `enablerconfig` (configure and monitor the *c-enabler*); and `enablermonitor` (only monitor the *c-enabler*).

action

A basic unit of work performed by *BEA WebLogic Process Integrator*. Actions define *workflow* and *task* behavior, and can be defined as part of tasks, *decisions*, and *events*. Numerous actions are shipped with WebLogic Process Integrator.

Administration Console

See *C-Enabler Administration Console* and *C-Hub Administration Console*.

Apache XML Parser

Third-party parser provided by the Apache Software Foundation that implements the W3C *XML*, *DOM*, and *SAX* standards.

attachment

A non-XML *payload* part of the *business message*. An attachment has a binary content type. It is represented in the C-Enabler Class Library by the `com.bea.b2b.protocol.messaging.Attachment` class.

authentication

Process by which *BEA WebLogic Collaborate* establishes the identity of a *principal* (*trading partners*, human users, and the *c-hub* server). *Digital certificates* with two-way *SSL* authentication are used between the *c-enabler* (*trading partner*) and the *c-hub*. Both the *c-hub* and the *c-enabler* examine and validate the remote *SSL* certificates with local security configuration information. Human users access the *WebLogic Collaborate Administration Console* through a Web browser. The authentication between the browser and the *c-hub* and between the browser and *c-enabler* is based on username and password.

authorization

Permission granted to *WebLogic Collaborate principals* (*trading partners*, human users, and *c-hub* server) to access certain resources.

Trading partner authorization is a *role*-based security feature whereby a trading partner is allowed to send and receive certain types of messages as defined by the trading partner's *subscription* in a specific *conversation*. Trading partners are authorized to access the *transport servlet* and *repository JDBC* pool on the *c-hub*.

Human users can be authorized to access *JSPs* and indirectly the *JDBC* pool on the *c-hub*. *WebLogic Server access control lists (ACLs)* can be configured selectively to allow or deny user access to these resources.

The *c-hub* server is authorized to access the *transport servlet* on the *c-enabler* in order to send messages to the *c-enabler*.

BEA WebLogic Collaborate

An open, standards-based software platform that enables companies to implement business-to-business collaborations on the Web. Key components of *WebLogic Collaborate* include a recoverable central *c-hub* and any number of *c-enablers*. A *trading partner* uses a *c-enabler* to send messages through the *c-hub* to other trading partners. *BEA WebLogic Collaborate* runs on the *EJB* platform *WebLogic Server* and is implemented entirely in *Java*. Central to *BEA WebLogic Collaborate* is *XML (eXtensible Markup Language)*, which provides an open data interchange format between loosely coupled participants. *WebLogic Collaborate*

also leverages J2EE and workflow technologies (*BEA WebLogic Process Integrator*) and supports leading e-commerce standards such as *RosettaNet* and *XOCP*.

BEA WebLogic Process Integrator

User modeling and *workflow* environment for *BEA WebLogic Collaborate*. The WebLogic Process Integrator product combines the functions of workflow definition and workflow monitoring. It is written in 100-percent pure Java, is fully hosted on WebLogic Server, and is integrated with the WebLogic Collaborate *XOCP c-enabler API*.

Bulk Loader

BEA WebLogic Collaborate utility that exports *repository* data to an XML file, imports XML data into the repository, and deletes data from the repository. The Bulk Loader can also be used to migrate repository data from one environment to another; for example, from one supported database to another. The Bulk Loader supports several JDBC databases.

business document

An XML-based *payload* part of a *business message*. A business document is represented in the C-Enabler Class Library by the `com.bea.b2b.protocol.messaging.BusinessDocumentclass`.

business message

The basic unit of communication exchanged between *trading partners* in a *conversation*. A business message is a multi-part MIME message that consists of *business documents*, *attachments*, and *message headers*.

business operation

A method of adding user-defined operations in *BEA WebLogic Process Integrator* by invoking EJBs or calling Java classes.

business protocol plug-ins

A collection of *logic plug-ins* that together provide support for *business protocols*. This includes out-of-the-box support for BEA WebLogic Collaborate *XOCP* and support for *RosettaNet*.

business protocols

Protocols that govern the exchange of business information between enterprises. BEA WebLogic Collaborate *trading partners* use *XOCP* and *RosettaNet* business protocols to send and receive *business messages*.

c-enabler

BEA WebLogic Collaborate 100-percent Java class library that is deployed at each participant node to allow a *trading partner* to send and receive *business messages* through the *c-hub*.

C-Enabler Administration Console

HTML interface that the *c-enabler* administrator uses to monitor active c-enablers, *c-enabler sessions*, and *conversations*; start and shut down c-enablers and c-enabler sessions; leave and terminate conversations; and track messages in a conversation.

c-enabler node

A system that participates in *conversations* by way of BEA WebLogic Collaborate *c-enabler* software installed on the node.

c-enabler session

A logical session between one *c-enabler node* and a *c-hub* that connects the *c-enabler node* to one *collaboration space*. Each c-enabler session allows a *trading partner* to collaborate with other trading partners in a single *collaboration space*.

c-enabler XPath routing expression

An *XPath* expression that is part of the *XOCP router logic plug-in*, is dynamically specified by a sending *c-enabler*, and specifies a list of *trading partners* to which a message should be sent. C-enabler XPath routing expressions can support references to *extended properties* in the *message-context XML document* that is generated for the message. C-enabler XPath routing expression specifications can be modified or overridden by *trading partner XPath routing expressions*.

c-enabler XML configuration file

Contains configuration information that the *c-enabler session* factory requires to initialize a *c-enabler session*. Each *c-enabler node* has one *XML* configuration file, but the file can specify configuration information for multiple c-enabler sessions.

certificate authenticator

A standard interface defined in WebLogic Server to map *digital certificates* to WebLogic Server users. After verifying the incoming certificate, the certificate authenticator maps the incoming *trading partner* associated with the certificate to the corresponding WebLogic Server user based on the certificate.

certification authority

A well-known and trusted organization, such as VeriSign, that issues *digital certificates* for the purpose of authenticating the certificate holder's identity to another party.

chain

Multiple *logic plug-ins* that share the same *business protocol* and are processed in a predefined sequence at run time. *Routers* and *filters* can consist, respectively, of multiple router logic plug-ins and multiple filter logic plug-ins, each of which executes as a *business message* passes through that plug-in point of the *c-hub*. After one plug-in finishes executing, the next one in the sequence (the "chain") is activated. Each successive plug-in can access the shared message information as the business message passes through the c-hub and can modify or override the changes made by predecessor plug-ins in the chain. Router logic plug-ins form one chain, and filter logic plug-ins form another. The logic plug-ins in the router chain execute first, followed by the plug-ins in the filter chain. Customer-supplied logic plug-ins must be part of a router logic plug-in chain or a filter logic plug-in chain, even when they do not have routing or filtering functionality.

c-hub

The central component of *BEA WebLogic Collaborate*, through which *trading partners* communicate with each other. The c-hub is responsible for routing messages between *c-enablers* and between non-*XOCP* clients, and for managing the life cycle of *conversations* between trading partners. A single recoverable c-hub can host multiple *collaboration spaces*. The c-hub provides the following services: *conversation coordination*, *security*, administration, *reliable messaging*, persistence and recovery processes, *local logging*, and *XML* services.

C-Hub Administration Console

HTML interface through which the *WebLogic Collaborate* c-hub system administrator enables, disables, configures, and monitors *c-hubs*, *trading partners*, and *collaboration spaces*. The console is also used to specify *business protocols*, *document definitions*, *message definitions*, *subscriptions*, and *Large Messaging Support*; to import *XML* files from and export them to the *repository*;

and to administer *SSL* protocol and the *HTTP proxy* server. The C-Hub Administration Console displays a list of all c-spaces, active *conversations*, and active *trading partners*, with identification, message, and state information.

c-hub recovery

C-hub functionality that enables the c-hub to survive node failures.

C-hub XPath filtering expressions

Sequence of *XPath* expressions that are part of the *XOCP filter logic plug-in*; are specified in the *repository* by an administrator; and are defined for the *c-hub*. C-hub XPath filtering expressions apply to all *business messages* that the c-hub receives for all recipient *trading partners*. Each expression can examine different parts of the *message-context XML document* and return True or False using different selection criteria, thereby blocking messages passed by previous expressions. The c-hub XPath filtering expressions are the final set of XPath expressions that evaluate messages before they are finally received by trading partners.

C-hub XPath routing expressions

Sequence of *XPath* expressions that are part of the *XOCP router logic plug-in*; are specified in the *repository* by an administrator; and are defined for the *c-hub*. The expressions specify all messages that should be sent from all trading partners. C-hub XPath routing expressions are evaluated after the *c-enabler XPath routing expression* and *trading partner XPath routing expressions*, and thus can modify or override the list of trading partners selected by those expressions.

collaboration space (c-space)

An abstract entity in which *trading partners* conduct and coordinate *conversations* with each other. A collaboration space has a specific business purpose and logically represents a digital marketplace or common trading environment. In conjunction with a central *c-hub*, a collaboration space provides the administration capability, conversation coordination, and underlying messaging services that are used to create a dynamic B2B integration environment. A single collaboration space can support multiple *business protocols*. Each collaboration space/business protocol combination has a unique URL, through which a trading partner using the business protocol will access the collaboration space.

CollaboratorMBean

MBean interface that represents *trading partners* in the run-time environment. C-hub management applications can use CollaboratorMBean objects to monitor trading partners.

configuration XML file

File that contains the database login information and the processing instructions for the *Bulk Loader* program. The corresponding parameter *DTD* is `BulkLoader.dtd`.

Confirmed_Delivery_to_Destination (QoS)

A messaging system *Quality of Service* that confirms message delivery to all destinations (recipient *trading partners*). Delivery can be made more reliable by use of options such as *Retry Attempts* and *Durability (Persistent option)*. `Confirmed_Delivery_to_Destination` can also provide additional information to the sender and administrators, including delivery status from recipients, complete *message tracking* for administrators, and confirmation from the *c-hub* upon completion of delivery. Availability depends on the *business protocol* that is used.

Confirmed_Delivery_to_Hub (QoS)

A messaging system default *Quality of Service* that confirms message delivery when the message reaches the *c-hub*. This QoS is often used when system performance is an issue. It does not provide delivery status from recipients, nor does it provide *message tracking*. Because the selection process for *trading partners* is performed after the confirmation reaches the sender, no information about potential recipients is available. Availability depends on the *business protocol* that is used.

Confirmed_Routing (QoS)

An XOCP messaging system *Quality of Service* that confirms delivery when the message reaches the *c-hub router* component. This option provides the list of *trading partners* selected by the router to receive the message. The sending *c-enabler* administrator receives *message tracking* information until the message reaches the router. Availability depends on the *business protocol* that is used.

conversation

A series of message exchanges between *trading partners* that takes place in a *collaboration space* and that is predefined according to a *conversation definition*. Each message in the conversation can cause any number of back-end transactions.

conversation coordinator

C-hub based service that manages *conversation life cycles* according to the rules of the corresponding *business protocols* and supports long-living, durable conversations that span multiple organizational boundaries. The conversation coordinator generates conversation IDs; registers *trading partners* in a conversation; delivers business and system messages to clients; maintains status information about conversations; and provides the conversational context for execution of the *business protocol*.

conversation definition

Consists of a unique *conversation* name, conversation version, *document definitions*, *trading partner* IDs, and trading partner *roles* for one conversation.

ConversationHandler

C-enabler interface that allows a *trading partner* to send and receive *business messages* in *conversations*. A conversation handler must be registered for one or more *conversation types* in order for the associated trading partner to participate in conversations. A *c-enabler session* supports one conversation handler per conversation type.

conversation initiator workflow

In *BEA WebLogic Process Integrator*, a workflow that is defined to have conversation properties and a Manual start property. This type of workflow initiates and terminates the *conversation*.

conversation life cycle

Refers to the fact that *conversations* are explicitly demarcated by a beginning and an end. Conversations can also end prematurely due to various network, computer, and software failures.

ConversationMBean

MBean interface that represents a *conversation* in the *c-enabler* run-time package. *C-enabler* management applications use *ConversationMBean* objects to monitor conversations in the run-time environment.

conversation participant workflow

In *BEA WebLogic Process Integrator*, a workflow that is defined to have conversation properties and a Business Message start property. This type of workflow does not initiate or terminate the *conversation*.

conversation termination

The ability of *BEA WebLogic Collaborate* and *BEA WebLogic Process Integrator* to terminate a *conversation* intelligently and in a controlled manner. For WebLogic Process Integrator clients, conversation is terminated when the *conversation initiator workflow* reaches a *done* state, as defined by the conversation termination property (Success or Failure) for the done node. *Conversation participant workflows* can end their own participation in a conversation, but only conversation initiator workflows can terminate a conversation. For BEA WebLogic Collaborate clients, conversation termination is initiated by the `Conversation.terminate()` operation, which is performed by the conversation initiator and which results in the receipt of a `ConversationHandler.onTerminate()` message by each conversation participant.

conversation type

A subset of *conversation definition*. The conversation type defines a conversation for one *trading partner*. It consists of the conversation name, version, and the trading partner's *role*.

Correlation ID (QoS)

Quality of Service that contains the message identification string. It can be used to correlate different *business messages* within an application. Availability depends on the business protocol that is used.

c-space

See *collaboration space*.

CSPACEMBean

MBean interface that represents a c-space. *C-hub* management applications use `CSPACEMBean` objects to monitor *collaboration spaces* in the run-time environment.

c-space URLs

Support the use of *XOCP* and non-*XOCP business protocols* in *collaboration spaces*. All c-spaces require one unique URL per business protocol that is supported by that c-space, thus allowing the *c-hub* to identify the proper c-space and business protocol of incoming messages.

data privacy

WebLogic Collaborate *SSL*-based security feature that provides link-level encryption of messages in communications between the *c-hub* and *c-enablers*. Human users do not have data privacy (*SSL*) when usernames and passwords are used for identification. *Digital certificates* are required to provide data privacy.

decision

A *BEA WebLogic Process Integrator* notation for a *workflow* component that contains a condition to be evaluated when a transition occurs. The result is either true or false, with subsequent flow control passed to the dependent node.

deferred synchronous message delivery

Message delivery method whereby control is returned to the application once a message is sent. A *message token* is returned to the application, which the application can access later to check the status of message delivery. Once the token is accessed, the application waits for a specified time or until any of the following events occur: acknowledgments received from all potential destinations; message times out; *conversation* in which message was sent terminates.

digital certificate

Digital equivalent of an ID card that *BEA WebLogic Collaborate* uses in conjunction with a public key encryption system to authenticate *trading partners* and the *c-hub* server. The digital certificate contains the owner's public key that has been digitally signed by a *certification authority*.

document definition

A schema such as a document type definition (*DTD*) that defines a valid document. In *BEA WebLogic Collaborate*, *XML* *DTDs* describe document definitions. The attributes are System ID (a *DTD* system identifier) and URL, which specifies the location of the document definition.

done

A *BEA WebLogic Process Integrator* notation for a *workflow* component that marks the entire workflow as done. At this point no further processing for any *task* occurs.

DTD (Document Type Definition)

Defines the format (grammar and syntax) that an *XML* message must follow.

DUNS (Data Universal Numbering System)

An identification system consisting of a distinctive nine-digit sequence that is an internationally recognized type of business (company) identifier in global electronic commerce transactions. Business identifiers are assigned to *trading partners* for external use; the business identifier is different from the internal *BEA WebLogic Collaborate* trading partner name. The *RosettaNet* business protocol uses DUNS numbers.

Durability (QoS)

Messaging system *Quality of Service* that specifies whether a durable message store is used to guarantee message delivery in case of node failures. The options are *Nonpersistent* (default) and *Persistent*. Availability depends on the *business protocol* that is used.

e-market

Generic term for the environment or container in which businesses conduct e-commerce with each other. An e-market supports a single business model and offers a set of services for conducting and managing *conversations* between *trading partners*. An e-market contains a message vocabulary, a business model, business processes, *roles*, and other metadata. The WebLogic Collaborate e-market environment is the *collaboration space (c-space)*. An e-market can include more than one collaboration space.

E-Market Integration (EMI)

Architecture that uses the Internet, a common vocabulary, and common standards such as *XML* and Java to enable multiple businesses to create fully integrated e-markets that encompass enterprise systems, trading communities, and existing B2B solutions.

EnablerMBean

MBean interface that represents a *c-enabler*. C-enabler management applications use *EnablerMBean* objects to monitor c-enablers in the run-time environment.

EnablerSessionMBean

MBean interface that represents a *c-enabler session*. C-enabler management applications use *EnablerSessionMBean* objects to monitor c-enabler sessions in the run-time environment.

event

An *WebLogic Process Integrator* notation for a *workflow* component and notification node. When an event is defined, the workflow waits for an *XML* message to trigger the event. Upon that trigger, various sub-actions defined within the event can be executed and/or workflow variables can be set.

extended properties

User-defined elements, attributes, and text that can be associated with entities in the *repository*. For this release, extended properties can be associated with *trading partners* only, through the *C-Hub Administration Console* or the *Bulk Loader* utility. These properties provide application extensions to the standard predefined attributes of trading partners. *XPath* routing and filtering expressions can reference these extended properties.

filter

Code that determines which messages a trading partner will receive. A filter consists of one or more logic plug-ins. *WebLogic Collaborate* provides an *XOCP filter* and a *RosettaNet filter*.

filtering expression

See *c-hub XPath filtering expression* and *trading partner XPath filtering expression*.

filter logic plug-ins

See *RosettaNet filter logic plug-in* and *XOCP filter logic plug-in*.

GlobalConversationMBean

MBean interface that represents a *conversation definition*. *C-hub* management applications use *GlobalConversationMBean* objects to monitor *conversations* in the run-time environment.

HTTP proxy

Server that monitors Internet usage and that hides internal network IP addresses to prevent intrusion. An HTTP proxy is configured to allow the pass-through of HTTP traffic.

HubMBean

MBean interface that represents a *c-hub*. *C-hub* management applications use *HubMBean* objects to monitor run-time *c-hub* activity.

Hop Timestamp

Facility through which each message in the WebLogic Collaborate messaging system is time-stamped upon its arrival at the *c-hub* and *c-enabler*. No input is required from the application. This information can aid in debugging and performance measurement. Availability depends on the *business protocol* that is used.

JMS (Java Message Service)

An API developed by JavaSoft that is used to access enterprise messaging systems from Java programs for the purpose of inter-application messaging. *BEA WebLogic Collaborate* uses the WebLogic Server implementation of JMS in the *c-hub*.

JMX (Java Management Extensions)

The basis of the administration API. JMX is the standard API for management, as published by Sun Microsystems, Inc.

join

A notation for a *workflow* component that links *BEA WebLogic Process Integrator events, tasks, and decisions*, thus controlling the transition between them. Joins can be AND or OR. If the join is an AND, all components linked by it must be satisfied before the successor to the join can be activated. If the join is an OR, only one of the components linked by that join has to be satisfied in order to activate the successor to the join and continue the flow.

Large Message Support

Feature that allows the *c-hub* administrator to specify that all messages above a certain size be stored on disk rather than in memory, which is the default location. The administrator configures Large Message Support on the Hub tab of the *C-Hub Administration Console*. Unless otherwise specified, a message must be at least 3 kilobytes to qualify for Large Message Support.

local logging

Performed by each *c-hub* and *c-enabler* in *BEA WebLogic Collaborate*. The local log records *log messages* reflecting *c-hub* activity, *c-enabler* activity, and user-defined output.

local view

Distributed *workflow* defined for one *role* in a *conversation*.

logging

See *local logging*, *log message*, and *system log message*.

logic plug-ins

Value-added logic that is installed on the *c-hub*; provides additional processing of the information that passes through the *c-hub*; and is subject to guidelines and interfaces imposed by BEA. *BEA WebLogic Collaborate* provides *XOCP* and *RosettaNet* filter and router logic plug-ins. *C-hub* owners can provide additional functionality by customizing the *WebLogic Collaborate* logic plug-ins or by creating custom logic plug-ins that conform to *XOCP* and *RosettaNet* standards. Customer-provided logic plug-ins can provide functionality other than routing and filtering, such as billing. See *RosettaNet filter logic plug-ins*, *RosettaNet router logic plug-ins*, *XOCP filter logic plug-ins*, and *XOCP router logic plug-ins*.

log message

Data recorded to the local log. *Local logging* messages have four levels of severity: fatal, error, warning, and info. The log contains the complete message and timestamp.

MBean (Managed Bean)

A Java object (JavaBean) and part of the Instrumentation level in *JMX* architecture that contains attributes for management operations. In *BEA WebLogic Collaborate*, MBean classes are used to monitor *c-enabler* and *c-hub* run-time information. MBeans are registered with the MBean server that runs inside the *c-hub* and *c-enabler* components. When MBeans are created, their attributes are populated from the *repository* and the *c-enabler XML configuration file*. At run time *WebLogic Collaborate* updates MBeans attributes to reflect the state of the running system. MBeans are implemented as Standard MBeans; that is, each class implements its own MBean interface. See *CollaboratorMBean*, *ConversationMBean*, *CSpaceMBean*, *EnablerMBean*, *EnablerSessionMBean*, *GlobalConversationMBean*, and *HubMBean*.

message

See *business message*, *log message*, *system log message*, *system transport message*, and *transport message*.

message context

Information that identifies and distinguishes a message in such a way that recipients can determine the structure and content of the message, as well as the *conversation* in which the message was sent. The message context identifies the conversation, *message definition*, message parts, content-type for each message part, and *document definition* for each message part.

message-context XML document

A document generated by the c-hub from each *XOCP* message and associated information in the *repository*. The associated *XPath* routing and filtering expressions examine different parts of the message-context XML document to determine a set of recipient *trading partners* for the message. The message-context XML document describes header and payload information about the message, such as the *c-hub*, large message information, *c-space*, *business protocol*, *conversation*, sending and receiving trading partners, *message definition*, and *document definition*.

message definition

Defines the business content, or *payload* (*business documents* and *attachments*), of a *business message*. A message definition consists of ordered message parts, which can have a content type of binary or *XML*. An XML type message part defines a business document and requires a *document definition*. A binary type message part defines an attachment and requires no other information.

message delivery methods

See *deferred synchronous delivery* and *synchronous delivery*.

message envelope

Container for the business message as it travels through the c-hub. As the business message enters the c-hub, the c-hub creates a message envelope for each intended recipient of the message. In addition to the business message, the message envelope contains high-level routing and processing information, such as the sender URL and the URL of the recipient. A message envelope is represented in the C-Enabler Class Library by the `com.bea.b2b.protocol.messaging.MessageEnvelope` class.

message filtering

Process by which *routers* and *filters* can restrict what kind of messages a specific *trading partner* receives; restrict messages to a specified set of trading partners; and implement matching algorithms so that information flows to the correct trading partners. See also *logic plug-ins*.

message header

Part of a *business message* that contains attributes such as sender and recipient information, *conversation* information, and *Quality of Service*.

message manipulators

Java classes that implement the message manipulator interface `com.bea.b2b.wlpi.MessageManipulator` and that are part of the integration between *BEA WebLogic Collaborate* and *BEA WebLogic Process Integrator*. Message manipulators use BEA WebLogic Process Integrator variables to create and process WebLogic Collaborate *business messages*.

message token

Token that is returned to an application after the application sends a message. The token provides the following information: message ID; ID of the *conversation* in which the message is sent; whether the message was sent successfully; and -- if allowed by the *Quality of Service* delivery option -- time elapsed for acknowledgments from all recipients; number of recipient destinations after initial selection (*router* specifications applied) at the *c-hub*; and number of recipient destinations after final selection (*filter* specifications applied) at the *c-hub*.

message tracking

Feature that provides information about the status of a message as it progresses through various predefined message tracking locations in the *BEA WebLogic Collaborate* messaging system. *C-hub* and *c-enabler* administrators can use message tracking information for debugging and to identify bottlenecks in the system. The availability of message tracking locations depends on the configuration of the WebLogic Collaborate system, the *Quality of Service* for the message, and the *business protocol*.

metadata

Data used to indicate the purpose, meaning, or location of other data. Database schemas and *XML* attributes are examples of metadata.

Nonpersistent (QoS option)

Default *Durability Quality of Service* option whereby the message is not stored anywhere in the *BEA WebLogic Collaborate* system. The message is not recoverable in the event of a whole or partial system failure that occurs while the message is enroute to its destination. Messages delivered with this option use fewer system resources and thus help provide better throughput.

OASIS

International consortium focused on the rapid adoption of product- and platform-independent formats that use public standards such as *XML*. OASIS operates www.xml.org, a site that offers resources and functions as a repository for XML specifications such as vocabularies, *DTDs*, schemas, and namespaces.

OBI (Open Buying on the Internet)

A consortium of companies, including BEA Systems, that develops and deploys open, platform-neutral standards for Internet-based procurement.

parser

See *Apache XML Parser*.

payload

Business content of a *business message*, consisting of one or more *business documents* and *attachments*. A payload part is represented in the C-Enabler Class Library by the `com.bea.b2b.protocol.messaging.PayloadPart` interface.

Persistent (QoS option)

Durability Quality of Service option whereby a message is persisted in multiple locations to a durable data store during the course of its delivery to a destination, in order to protect against node failures. The message is stored and forwarded at each location. An application can direct the system to persist messages on a per message or per *conversation* basis. However, the *c-hub* must be recoverable in order for the message to persist. Persistence is achieved at the expense of system throughput. A persistent message travels slower and consumes more resources than a message that is *Nonpersistent*.

PIPs (Partner Interface Processes)

Standards supported by a master dictionary, coupled with an established implementation framework (exchange protocols) defined by *RosettaNet*. PIPs enable companies to integrate inbound and outbound supply chain processes;

rapidly deploy interoperable applications for supply chain functions; and optimize networks of suppliers, manufacturers, and distributors. As part of the exchange framework, PIPs provide defined fields and labels for different types of transactions. The purpose of each PIP is to provide common business and data models and documents.

plug-ins

See *logic plug-ins*.

plug-in chain

See *chain*.

principal

An entity that requires access to the *BEA WebLogic Collaborate* system. Principals include *trading partners*, human users, and the *c-hub* server.

Process Engine

BEA WebLogic Process Integrator run-time controller and *workflow* engine. The Business Process Engine tracks workflow instances and executes and manages workflows.

Quality of Service (QoS)

BEA WebLogic Collaborate class of options that the messaging system and *c-hub* use to define the levels of reliability in message delivery. The messaging system provides options for reliable delivery of messages in the event of network-link and node failures. The *c-hub* uses options that improve reliability by using a persistent data store that will enable it to recover in case of node failures.

realm

Domain for a set of security features that provide access to *access control lists (ACLs)*, names of *principals*, and related security services. The realm organizes security information and defines its range of operations. It determines how users are authenticated. *BEA WebLogic Collaborate* leverages security functionality from the WebLogic Server realm.

registration

A request by a *c-enabler* to receive *business messages* for a given *role* in a *conversation definition*. Registration requests are sent to the *c-hub* and require the *c-enabler's trading partner* to have a *subscription*.

reliable messaging

In BEA WebLogic Collaborate, the capability to offer the reliable delivery of messages through various *Quality of Service* options that can confirm receipt of messages and provide message delivery in case of node failures; message logging and tracking; correlation of messages; retry attempts; and a choice of message delivery methods.

repository

Stores configuration information for the *c-hub*, *collaboration spaces*, *trading partners*, and *conversations*; *document definitions*; *XML schemas*; and *XSLT mappings*. The primary store is a relational database accessed through JDBC. The repository can be configured and accessed through an HTML interface, the *C-Hub Administration Console*. The *Bulk Loader* utility provides repository data updates and processing and permits data deletion.

repository export

Bulk Loader functionality that extracts data from the *repository* into an *XML* repository data file.

repository import

Bulk Loader functionality that loads into the *repository* *XML* data that was either extracted from the repository or created by editing *XML*.

Retry Attempts (QoS option)

Messaging system *Quality of Service* whereby the messaging system attempts to re-send the message a specified number of times at any delivery point in order to mask intermittent network failures. The messaging system waits for a predefined interval before attempting to re-send a message. Availability depends on the specific protocol.

role

In *BEA WebLogic Collaborate*, the *subscription* unit in a *conversation*. A role is defined in terms of the documents that can be sent or received by a *trading partner* in the conversation. The role defines what a trading partner can do, such as buy or sell. Each conversation has two or more roles. Role attributes include a unique role name, in-document definitions, and out-document definitions.

In *BEA WebLogic Process Integrator*, an area of responsibility, ability, or authorization level that is shared by a group of individuals. An individual user can belong to one or more roles.

RosettaNet

(1) A non-profit consortium of high technology companies whose purpose is to define and standardize business interfaces for e-commerce.

(2) The business and transport protocol specified by the RosettaNet consortium.

RosettaNet filter

Filter provided by *BEA WebLogic Collaborate* that can consist solely of the *RosettaNet filter logic plug-in*, provided by *WebLogic Collaborate*, or can consist of a *chain of logic plug-ins* that includes one or more customer-supplied filter logic plug-ins at the beginning of the chain, followed by the *RosettaNet filter logic plug-in*.

RosettaNet filter logic plug-in

Logic plug-in that determines which *RosettaNet business messages* a *trading partner* will receive.

RosettaNet router

Router provided by *BEA WebLogic Collaborate* that consists of a *chain of logic plug-ins*: the *RosettaNet router logic plug-in* at the beginning of the chain, the *RosettaNet router enqueue logic plug-in* at the end of the chain, and optional customer-supplied *RosettaNet logic plug-ins* in the middle.

RosettaNet router enqueue logic plug-in

Logic plug-in that queues a *RosettaNet* message for potential delivery to a recipient.

RosettaNet router logic plug-in

Logic plug-in that sets the recipient for a *RosettaNet* message.

router

Code that determines the trading partners to which a message will be sent. A router consists of a *chain of logic plug-ins*. There is one router for each *business protocol*. *WebLogic Collaborate* currently provides a *RosettaNet* router and an *XOCP* router.

router logic plug-in

See *RosettaNet router enqueue logic plug-in*, *RosettaNet router logic plug-in*, *XOCP router enqueue logic plug-in*, and *XOCP router logic plug-in*.

routing expression

See *c-enabler XPath routing expression*, *c-hub XPath routing expression*, and *trading partner XPath routing expression*.

routing service

A BEA-specific internal *c-hub* component that provides a list of recipients for a message, performs final validation of message recipients, and stores the message for delivery to *trading partners*.

scheduling service

A BEA-specific internal *c-hub* component that helps ensure proper routing, queuing, and sequencing of messages.

Secure Sockets Layer

See *SSL*.

security

See *authentication*, *authorization*, *data privacy*, and *SSL*.

shape

The graphical representation, in the *BEA WebLogic Process Integrator* drawing area, of a *workflow node*.

SSL (Secure Sockets Layer)

Internet transport security protocol that provides *data privacy* between applications. In *WebLogic Collaborate*, two-way SSL requires *digital certificates* from both communicating applications.

start

In *BEA WebLogic Process Integrator*, the first component of a *workflow*. The triggering properties of the workflow are specified in the start node, and the first *shape* following the start node will be the first active component in the workflow.

subscription

The right (permission) of a *trading partner* to participate in a specific *role* in a specific *conversation definition*. To send *business messages* for a role in a *conversation*, a *trading partner* must have a subscription to that role. Subscription information includes trading partner name, conversation name, conversation version, and the role to which the trading partner is subscribed.

synchronous message delivery

Message delivery method whereby the messaging system returns control to the application once the outcome of the message transmission is known. Control is returned to the application after any of the following events occur: acknowledgements received from all potential destinations; message times out; or *conversation* in which message was sent terminates.

systematic collaboration

A type of collaboration whereby from the beginning of the business relationship, each *trading partner* is cognizant of all other participants in the collaboration. The business relationship is ongoing. Process changes occur infrequently.

System Error Handler

A *BEA WebLogic Process Integrator* error handler available to every *workflow template definition*. Unless overridden by the workflow template definition, WebLogic Process Integrator invokes the System Error Handler whenever an exception occurs. The System Error Handler responds by rolling back the transaction and throwing an exception to the client application.

system log message

Log message placed in the local log to record *c-hub* or *c-enabler* activity; for example, a system log message indicating the inability to deliver a *transport message* to a *c-hub* because the *c-hub* has died.

system transport message

A type of *XOCP transport message* exchanged between *c-hub* and *c-enabler* to effect system activity; for example, notification of *conversation termination*.

task

A discrete activity performed automatically by the *BEA WebLogic Process Integrator* server or by a *workflow* user. A workflow consists of a series of tasks. Task behavior is determined by various task properties and by the task *actions* that define the processing to be performed as certain *task events* occur.

task events

Events in the life cycle of a *task* during which a *workflow* analyst can program *actions* to occur. When a task event occurs, the actions associated with the event are executed. The four task events are Created, Activated, Executed, and Marked Done.

Timeout (QoS)

Messaging system *Quality of Service* that specifies how long a sender waits for acknowledgment of a message before the message times out. After the timeout, the message, acknowledgment processing, and retries are abandoned. Availability depends on the *business protocol* that is used.

token

See *message token*.

trading partner

A representation of an *e-market* participant, such as an automobile parts manufacturer, that is authorized to participate in one or more *collaboration spaces* on a *c-hub*. A trading partner is registered at the *c-hub* level.

trading partner protocol

Defines additional *trading partner* information needed for non-*XOCP business protocol* support. The trading partner protocol consists of the URL (external URL to which *business messages* are routed); business ID (business identifier to be used by an external trading partner, which is distinct from the internal WebLogic Collaborate trading partner name); business ID type (type of business identifier, such as *DUNS* for *RosettaNet*, which must be synchronized with the related *business protocol plug-in*); and business protocol definition name (name of the business protocol to associate with the trading partner).

trading partner XPath filtering expressions

A sequence of *XPath* expressions that are part of the *XOCP filter plug-in*; are specified in the *repository* by an administrator; and are associated with a particular target (recipient) *trading partner*. The expression reduces the set of messages that the trading partner receives by selectively passing certain messages to the targeted trading partner and blocking other messages before they are sent from the *c-hub*. Each trading partner *XPath* filtering expression can examine different parts of the *message context XML document*, return True or False using different selection criteria, and thereby potentially block a message that was passed by an earlier expression.

trading partner XPath routing expressions

XPath expressions that are part of the *XOCP router plug-in*; are specified in the *repository* by an administrator; and are defined for a particular sending *trading partner*. The expressions specify the list of recipients for each message sent from that trading partner. Each trading partner *XPath* routing expression can examine

different parts of the *message-context XML document* and select a different set of recipient trading partners. The trading partners produced by each expression can replace or add to the previously generated set of recipients.

transport message

A data entity that is exchanged between *c-enabler nodes* and *c-hubs*. There are two types of transport message: *system transport message* and *business message*.

transport service

A BEA-specific internal *c-hub* component that reads incoming messages to be passed to a decoder and writes outgoing messages from an encoder for delivery to a recipient. In this release of *BEA WebLogic Collaborate*, the decoder/encoder parts must use the same *business protocol*, such as *XOCP* or *RosettaNet*, and the transport component is based on HTTP/HTTPS transport protocol.

transport servlet

The entry point into the *BEA WebLogic Collaborate* system for HTTP/HTTPS communication between the *c-hub* and *c-enablers* that use *XOCP* and between the *c-hub* and non-*XOCP* clients. Access control at this entry point is set to allow relevant *WebLogic Collaborate principals* only. (Human users do not access this entry point.)

W3C (World Wide Web Consortium)

Major standards body that establishes international standards for client and server protocols in order to facilitate Internet-based communications and commerce.

WebLogic Collaborate

See *BEA WebLogic Collaborate*.

WebLogic Process Integrator

See *BEA WebLogic Process Integrator*.

WebLogic Process Integrator Studio

BEA WebLogic Process Integrator client application that is used to define and monitor *workflows*.

workflow

A business process, such as order processing, that is automated in whole or in part, during which information is passed to the right participant at the right time according to a set of intelligent business rules that allow computers to perform most of the work, leaving humans to deal only with exceptions.

workflow expression

An expression that defines a calculation or an evaluation that *BEA WebLogic Process Integrator* performs at run time. Workflow expressions use syntax comparable to that of SQL.

workflow nodes

BEA WebLogic Process Integrator workflow components, consisting of *starts*, *tasks*, *events*, *joins*, *decisions*, and *done*s.

workflow template

A folder or container in *BEA WebLogic Process Integrator Studio*. A workflow template represents a *workflow* and is given a meaningful name such as Order Processing or Billing. The workflow template aggregates various *workflow template definitions* (versions) of its implementation. A workflow template controls which organizations can use the contained workflow template definitions.

workflow template definition

A definition or version of a *workflow*, distinguished by its effective and expiration dates. At run time, *BEA WebLogic Process Integrator* starts an instance (or session) of a workflow template definition, selecting the most effective (or current and active) definition.

X.509

A widely used specification for *digital certificates*.

XML (eXtensible Markup Language)

A metadata language and subset of SGML that is rapidly becoming a universal standard for defining, validating, and sharing data formats and documents. Because XML is a text-based, non-binary format and uses syntax rather than binary markers to organize data, it can be deployed across heterogeneous and potentially incompatible systems and platforms. Its extensibility derives from markup symbols that are unlimited and self-defining, unlike those of HTML. Like

HTML, XML can describe how a file is to be displayed. Unlike HTML, XML defines how data is described. XML is a crucial component of the *BEA WebLogic Collaborate* solution.

XML services

Apache *XML* services provided for *BEA WebLogic Collaborate*, which at this time include *Apache XML Parser*, DOM implementation, and *XSLT* style sheet processor.

XML vocabulary

A set of *XML* tags that define the elements that may occur in a *DTD*. An XML vocabulary can be used for a particular industry or business function.

XOCP (eXtensible Open Collaboration Protocol)

Default *business protocol* used by *BEA WebLogic Collaborate*. Trading partners use *XOCP* to send and receive messages by way of a *c-hub* and *XOCP c-enabler API*. *XOCP* provides support for *conversation life cycle* tracking, dynamic routing of messages, and the management of business interactions by a mediating *c-hub*.

XOCP c-enabler API

Key *c-enabler* component that is the interface between the application and the *c-enabler* (class library). The API is used in starting *c-enabler sessions*, creating *conversations*, and sending and receiving *business messages*. The *c-enabler API* consists of the following packages: *c-enabler* (`com.bea.b2b.enabler` and `com.bea.b2b.enabler.xocp`); *conversation* (`com.bea.b2b.enabler.xocp.conversation.local`); and *messaging* (`com.bea.b2b.protocol.messaging` and `com.bea.b2b.protocol.xocp.messaging`).

XOCP filter

Filter provided by *WebLogic Collaborate* that can consist solely of the *XOCP filter logic plug-in*, provided by *BEA WebLogic Collaborate*, or can consist of a *chain* of logic plug-ins that includes the *XOCP filter logic plug-in* and one or more customer-supplied filter logic plug-ins.

XOCP filter logic plug-in

Logic plug-in that determines which *XOCP* messages a *trading partner* will receive. The *XOCP* filter logic plug-in uses sequences of *XPath* expressions that are defined in the *repository*. The *XOCP* filter logic plug-in is the only filter logic plug-in that uses *XPath* expressions. *Trading partner XPath filtering expressions* are evaluated first, followed by *c-hub XPath filtering expressions*.

XOCP router

Router provided by WebLogic Collaborate that consists of a *chain* of logic plug-ins placed in the following sequence: the *XOCP router logic plug-in*, optional customer-supplied logic plug-ins, and the *XOCP router enqueue logic plug-in*.

XOCP router enqueue logic plug-in

Logic plug-in that queues an *XOCP* message for potential delivery to each recipient.

XOCP router logic plug-in

Logic plug-in that chooses the recipients for a message that uses the business protocol *XOCP*. The *XOCP router* logic plug-in uses sequences of *XPath* expressions that are defined in the *repository* and by the sending *c-enabler*. The *XOCP* router logic plug-in is the only router logic plug-in that uses *XPath* expressions. *XPath* routing expressions are evaluated in the following sequence: *c-enabler XPath routing expression*, *trading partner XPath routing expressions*, and *c-hub XPath routing expressions*.

XPath

XML path language. *XPath* models a *message-context XML document* as a tree of nodes. It is used to address nodes of the *XML* document. *XPath* also has a natural subset that can be used to test whether a node matches a pattern. In *BEA WebLogic Collaborate*, *routers* and *filters* use this feature to identify *trading partner* elements of *c-hub* configuration information. *XPath* routing expressions specify business criteria for message distribution. *XPath* filtering expressions return a true or false result and act as gatekeepers to filter out unwanted messages sent to a receiving trading partner. *XPath* expressions are associated exclusively with messages that use the default *XOCP* protocol. See also *c-enabler XPath routing expression*, *c-hub XPath filtering expressions*, *c-hub XPath routing expressions*, *trading partner XPath filtering expressions*, and *trading partner XPath routing expressions*.

XPath filtering expression

See *XPath*, *c-hub XPath filtering expressions*, and *trading partner XPath filtering expressions*.

XPath routing expression

See *c-enabler XPath routing expression*, *c-hub XPath routing expressions*, and *trading partner XPath routing expressions*.

XSL (eXtensible Style Sheet Language)

W3C standard for describing presentation rules that apply to *XML* documents.

XSLT (XSL Style Sheet Language Transformations)

W3C specification that describes how to transform *XML* documents from one *document definition* format to another.